Analysis and Decoding of Linear Lee-Metric Codes with Application to Code-Based Cryptography

Dissertation
zur
Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)
vorgelegt der
Mathematisch-naturwissenschaftlichen Fakultät
der
Universität Zürich
von

Jessica Bariffi

von
Lugano, TI

Promotionskommission

Prof. Dr. Joachim Rosenthal (Vorsitz)
Prof. Dr. Jean Bertoin
Dr. Hannes Bartz

Zürich, 2024

To my friends and family

Abstract

Lee-metric codes are defined over integer residue rings endowed with the Lee metric. Even though the metric is one of the oldest metric considered in coding-theroy and has interesting applications in, for instance, DNA storage and code-based cryptography, it received relatively few attentions compared to other distances like the Hamming metric or the rank metric. Hence, codes in the Lee metric are still less studied than codes in other metrics. Recently, the interest in the Lee metric increased due to its similarities with the Euclidean norm used in lattice-based cryptosystem. Additionally, it is a promising metric to reduce the key sizes or signature sizes in code-based cryptosystem. However, basic coding-theoretic concepts, such as a tight Singleton-like bound or the construction of optimal codes, are still open problems.

Thus, in this thesis we focus on some open problems in the Lee metric and Lee-metric codes. Firstly, we introduce generalized weights for the Lee metric in different settings by adapting the existing theory for the Hamming metric over finite rings. We discuss their utility and derive new Singleton-like bounds in the Lee metric. Eventually, we abandon the classical idea of generalized weights and introduce generalized distances based on the algebraic structure of integer residue rings. This allows us to provide a novel and improved Singleton-like bound in the Lee metric over integer residue rings. For all the bounds we discuss the density of their optimal codes.

Originally, the Lee metric has been introduced over a q-ary alphabet to cope with phase shift modulation. We consider two channel models in the Lee metric. The first is a memoryless channel matching to the Lee metric under the decoding rule "decode to the nearest codeword". The second model is a block-wise channel introducing an error of fixed Lee weight, motivated by code-based cryptography where errors of fixed weight are added intentionally. We show that both channels coincide in the limit of large block length, meaning that their marginal distributions match. This distribution enables to provide bounds on the asymptotic growth rate of the surface and volume spectrum of spheres and balls in the Lee metric, and to derive bounds on the block error probability of the two channel models in terms of random coding union bounds. As vectors of fixed Lee weight are also of interest to cryptographic applications, we discuss the problem of scalar multiplication in the Lee metric in the asymptotic regime and in a finite-length setting. The Lee weight of a vector may be increased or decreased by the product with a nontrivial scalar. From a cryptographic view point this problem is interesting, since an attacker may be able to reduce the weight of the error and hence reduce the complexity of the underlying problem. The construction of a vector with constant Lee weight using integer partitions is analyzed and an efficient method for drawing vectors of constant Lee weight uniformly at random from the set of all such vectors is given.

We then focus on regular low-density parity-check (LDPC) code families defined over integer residue rings and analyze their performance with respect to the Lee metric. We determine the expected Lee weight enumerator for a random code in fixed regular LDPC code ensemble and analyze its asymptotic growth rate. This allows us to estimate the expected decoding error probability. Eventually, we estimate the error-correction performance of selected LDPC code families under belief propagation decoding and symbol message passing decoding and compare the performances.

The thesis is concluded with an application of the results derived to code-based cryptography. Namely, we apply the marginal distribution to improve the yet known fastest Lee-information set decoding algorithm.

Acknowledgements

First and foremost I would like to thank my advisor Joachim Rosenthal who pushed me to move out of my comfort zone and pursue a Ph.D. in his group in combination with the German Aerospace Center (DLR) in Munich, Germany. I am grateful for all his trust and belief in me, for the continuous support not only regarding the thesis but also my future and goals, and for being available any time even though I stayed in Munich and visited Zurich only every now and then.

Secondly, I thank Hannes Bartz, my supervisor at DLR, for guiding me through this adventure, for all his advices and for the freedom I had in choosing the research topic and projects. I am grateful for everything I have learned within his group. Moreover, I thank Gianluigi Liva for giving me the opportunity to come to DLR, and for all the inputs, discussions and positivity during all these years. His knowledge and his honest interest in research is truly inspiring. I am thankful for every second and comment he offered and for pushing and believing in me whenever I had doubts.

At this point I want to thank Violetta Weger! Not only has she taken the role of my mentor, helped me find exciting projects, and offered her help whenever she could, but she has also become a true friend. I thank her for all her positive and pushing words, for her patience and experience, and for all the support and trust.

I am grateful to all my colleagues at DLR and at the University of Zurich for the continuous exchange. Especially, I want to thank Felicitas Hörmann for being the best and most supportive office mate, and for always offering her help and answering even the stupidest of questions. I thank Stefano Tinelli and Riccardo Schiavone for their friendship, the coffee breaks and afternoon walks talking about life and research, for all the dinners and adventures we planed, and for making the working experience at DLR complete. Vi voglio tanto bene! I also want to thank my remaining co-authors Hugo Sauerbier Couvée, Karan Khathuria and Thomas Jerkovits for the great collaboration.

I would like to thank my family and my group of friends back in Zurich and spread all over the world for always being there even on a longer distance, and for the support that they offerd me so naturally. Coming back home has always felt like a little vacation.

Doing a Ph.D. abroad is not always easy. I consider myself very lucky to be surrounded by an amazing group of friends contributing to a great personal experience in Munich. Naming each and every one separately would end in a seemingly infinite list of names, but I am thankful to each and every one in this seemingly infinite list for the support, for all the great time we had and for making Disco Maistrasse one of Munichs most historical nightclubs. Thank you all for sharing this experience with me and for being a part of my Munich-family. More explicitly, I want to thank Ludovica Cammarone and Lorenzo Zaniboni for their incredible positivity and availability every day. To say it in Ludo's words: "Grazie di esistere! Siete mitici." Especially, I want to thank Edoardo Giordano. Even though my research does not lie in his field of expertise, he was always curious to hear about the problems I tried to solve. Talking to him helped me break down the problem and see the essential parts of it. I thank him for his patience, for finally showing his emotions while playing a game he is loosing, and for being a truly GREAT friend.

Last but not least, I would like to express my sincerest gratitude to Pietro Mambelli. His unconditional support and love, his positive energy, and his calmness helped me overcome every insecurity and obstacle I faced in this period. I thank him for all his good advice, for his time listening to my struggles, for being proud of the even tiniest of my achievements, for being my save space, and for believing in me whenever I did not. Especially, I want to thank him for accepting me as I am. This thesis would not have been possible without him. Grazie di tutto Pietro!

Jessica Bariffi
Zurich, April 2024

Contents

Chapter 1

Introduction

This thesis focuses on the Lee metric introduced in 1958 by Lee [83] which provides an interesting alternative to the Hamming metric [70] and rank metric [48, 56] which are considered for orthogonal modulation and network coding, respectively. This metric has later been considered and studied further by Prange [103], Massey [91], Golomb and Welch [63], Berlekamp [23] and many more. In 1967, Massey was the first one to introduce a channel "matching" to the Lee metric. In 1971, Chiang and Wolf [42] have derived all the discrete, memoryless, symmetric channels matched to the Lee metric. The Lee metric is most known for the celebrated result in [71], where the authors showed that some optimal non-linear binary codes can be represented as linear codes over $\mathbb{Z}/4\mathbb{Z}$ endowed with the Lee metric. Although being a rather old metric considered in coding theory, it has gained more attention only recently with its interesting applications to, for instance, code-based cryptography ([16, 18, 40, 108, 129, 130]) and DNA-storage [57]. However, compared to other metrics used in coding theory there are still many open questions in the Lee metric. This thesis provides the study of the Lee metric in terms of its algebraic structure as well as its application to communication channels and code-based cryptography reducing the gap of open problems for Lee-metric codes.

In 1948, Shannon laid the foundation of information theory and classical coding theory in his seminal work "A mathematical theory of communication" [117]. A classical communication system, as shown in Figure 1.1, consists of a source that wants to communicate a message to a receiver. Due to limitations on the construction of a channel model, the transmission may contain errors. The main problem, thus, is to ensure reliable communication over a noisy channel. To tackle this problem, we consider two main concepts describing the communication system: source coding and channel coding. The first discipline deals with removing wasteful redundancy from the information of the source to make it more compact and practical to use. Usually, the source encoder *compresses* the received data to cut down the length of the message. This data, however, is not robust against corruption caused by a noisy channel. Hence, the data at the source encoder's output is encoded by means of an error-correction code also referred to as "channel code". That is, we add controlled redundancy to the message in order to ensure reliable communication.

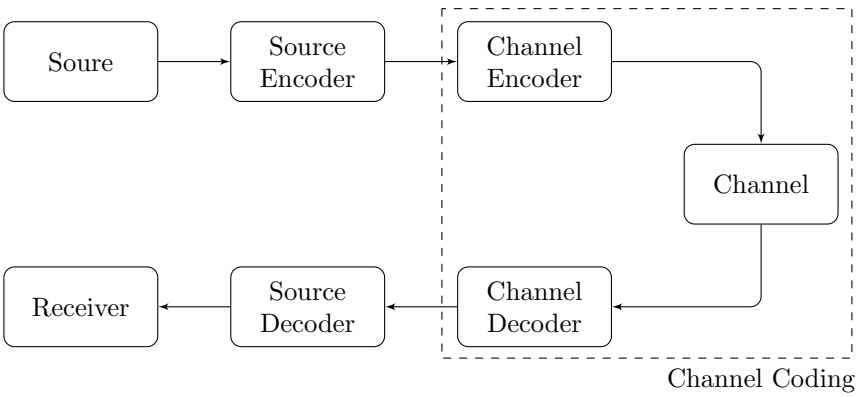

FIGURE 1.1: Communication System Diagram

Classical coding theory deals with the error correction in a communication system, i.e., the channel coding part highlighted in Figure 1.1. It concentrates on the design of the channel encoder and decoder. In a nutshell, we have the following communication set-up. Assume a source wants to send a message m. In classical coding theory, this message is a vector of fixed length k defined over a finite field $\mathbb{F}_q$ (the *alphabet*) of q elements. At the channel encoder, we use a subspace of the ambient space $\mathbb{F}_q^n$ which we call a *code*. The elements of a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ are called *codewords* and the dimension of $\mathcal{C}$ as a linear subspace is given by the length k of the original message m. We then transform the message m into a codeword $c \in \mathcal{C}$ of length $n \geq k$ using an injective map. Since $k \leq n$ this mapping adds redundancy to the message m. This step is crucial to correct potential errors. The codeword c is then sent through a (noisy) channel. We consider a probabilistic channel with a given transition probability. That is an error vector $e \in \mathbb{F}_q^n$ is added following a given probability distribution. Hence, at the receiver side we obtain a vector $r = c + e$. In order to understand the impact of the error vector, the ambient space is endowed with a metric measuring the total amount of errors added at each position. Classically, the Hamming metric is used measuring the amount of erroneous positions in r (i.e., the number of positions in which r differs from c). Every code has the capability to correct a given amount of errors in the respective distance measure. This error-correction capability of a code is given by its algebraic structure and the underlying distance measure. The "bigger" the error, the more difficult the task of recovering the codeword c. If the distance between the received word r and the original codeword c is below the error-correction capability of $\mathcal{C}$, a receiver is able to recover c using a suitable decoding algorithm More precisely, it depends on the *minimum distance* of the code capturing the smallest distance (in the respective underlying metric) between any two distinct codewords. Namely, the larger the minimum distance, the better the error-correction capability of the code.

Codes with Optimal Distance Properties

Since coding theory aims to correct as many errors in a communication model as possible, an important task is the study of codes with "good" distance properties in the sense of their error-correction capability. The most famous bound capturing the maximal error-correction capability of a code in terms of its minimum distance is the Singleton bound. In the Hamming metric this bound was introduced by Singleton [121] and was already studied by Komamiya [80]. The idea of the Singleton bound for a code of minimum distance d is to puncture the code in $d-1$ positions. This results in a punctured code of the same cardinality, since it has still a minimum distance of at least 1. However, since the punctured code is reduced in its block length, the maximal amount of possible codewords also reduced. Comparing the cardinalities of the punctured code yields then the Singleton bound. Codes attaining the Singleton bound in the Hamming metric are called *maximum distance separable* (MDS) codes and are constituting some of the most used and studied codes in coding theory. It is knwon that codes over $\mathbb{F}_q$ attain the Hamming-metric Singleton bound with high probability when letting the size of the finite field q tend to infinity (see, for instance, [90, Chapter 11]). To see this for given length n and dimension k, we show that the fraction of non-MDS codes withing all codes of the same dimension and length is negligible as the field size grows. In fact, a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ of dimension k is MDS if and only if every full size minors of any generator matrix is invertible. Additionally, the set of all $[n,k]$ linear block codes defined over a field $\mathbb{F}_q$ can be parameterized by the Grassmann variety $\mathrm{Grass}(k, \mathbb{F}_q^n)$ consisting of all k-dimensional linear subspaces inside $\mathbb{F}_q^n$. It follows from a result proven in the general situation of convolutional codes [110], that the set of $[n,k]$ MDS linear codes is Zariski open inside the Grassmannian $\mathrm{Grass}(k, \mathbb{F}_q^n)$. The Reed-Solomon construction shows that this Zariski open subset is nonempty as soon as the field size is larger than the block length n. Over the algebraic closure, the seto f MDS linear codes is also Zariski-dense as the complement is contained in a Zariski closed set of dimension strictly smaller than the dimension of $\mathrm{Grass}(k, \mathbb{F}_q^n)$. Hence, for given block length n and dimension k and for a growing field size, the fraction of MDS codes versus the non-MDS codes tends to one. Instead, if one lets the block length n tend to infinity, the probability for a code to attain the Singleton bound becomes negligible.

Changing the underlying metric yields completely different results. For instance, we can consider the rank metric which was introduced in 1978 by Delsarte in [48]. A Singleton-like

bound in the rank metric was then derived in [56] by Gabidulin laying the foundation of the study of optimal codes in the rank metric, called *maximum rank distance* (MRD) codes. In fact, we know that for $\mathbb{F}_{q^m}$-linear codes, MRD codes are dense when letting q or m tend to infinity [95]. For $\mathbb{F}_q$ linear codes, however, MRD codes are sparse when letting q tend to infinity [68], except for some special cases where m or n are 2 [6, 33, 62]. In the Lee metric, the situation is again different. The first Singleton-like bound in the Lee metric has been introduced by Shiromoto in 2000 [119] and has been further studied in [5, 34]. The existing bounds on the minimum distance are rather loose, making the analysis of *maximum Lee distance* codes nearly impossible. In fact, it has been shown in [34] that there is only one nontrivial linear code in the Lee metric attaining Shiromoto's bound.

In this thesis we abandon the classical idea of puncturing a code to derive a Singleton-like bound. Instead, we introduce the concept of generalized weights to the Lee metric to derive bounds on the minimum Lee distance.

Iterative Decoding and LDPC Codes

Another important task of channel coding is the design of codes that, for a given communication channel, allow for efficient encoding and decoding. Focusing on the code design, we aim to transmit as much information with as little redundancy as possible. On the other hand, redundancy adds robustness against corruption and minimizes errors. To face these contradicting tasks, we might switch the point of view and define the *rate* of a channel. That is the relative fraction of information bits per total number of bits sent through the channel. Shannon introduced the notion of the *channel capacity*, that is the maximum possible rate for which a reliable communication is guaranteed. He showed that nearly error-free communication over an unreliable channel is possible as long as the rate is strictly smaller than the channel capacity. This proof, however, is non-constructive and does not take into account the computational complexity of the encoding and decoding algorithm. Therefore, one big task in coding theory consists in designing code families together with an efficient (low-complexity) decoding algorithm such that a performance near the channel's capacity can be achieved. In terms of complexity, iterative decoders showed a great potential. In 1988, Judea Pearl introduced the *belief propagation* algorithm based on probabilistic reasoning [99]. Nowadays, many iterative decoders can be traced back to Pearl's belief propagation algorithm. The main idea of such algorithms is to use a graphical representation of a parity-check matrix of a code into check nodes (representing the rows of the matrix) and variable nodes (corresponding to the columns). Then the messages are communicated in terms of probability distributions (and therefore the name *belief*) between the connecting check nodes and variables nodes. The connections between the nodes are determined by the entries of the parity-check matrix.

In his seminal work [58] in 1963, Gallager introduced a new family of codes based on a sparse parity-check matrix, that is a matrix with relatively few nonzero entries. This family of codes is generally known as *low-density parity-check* (LDPC) codes and have almost been forgotten for thirty years. Besides the family of LDPC codes, Gallager introduced a message-passing algorithm making use of the graphical representation of the codes. LDPC codes have shown to perform close to the Shannon limit under low-complexity iterative decoding algorithms and are hence widely used in communication systems. In [114] the first definition of Lee-LDPC codes appeared together with a bit-flipping decoding variant that they introduced as Lee-symbol flipping decoder. This code family defines the first Lee-metric code which is efficiently decodable.

Code-based Cryptography

Apart from information theory, coding theory is also applied to cryptography and more explicitly to post-quantum cryptography. Nowadays, all public key cryptosystems based on the factorization of large integers into its prime factors (like RSA [109]) or the discrete logarithm problem (in elliptic curve cryptography) can be broken by a capable quantum computer using Shor's algorithm [120]. Therefore, the National Institute of Standards and Technology (NIST) launched a competition in 2016 with the aim of finding new cryptosystems that are resistant against quantum-attacks. Ever since, many schemes have been proposed in five main directions: code-based, lattice-based, isogeny-based, hash-based and multivariate cryptography. Most of these schemes are based on mathematically hard problems, for which

it is believed that they are resistant against quantum-attacks [41]. We refer to [25] for more details on post-quantum cryptography. In the currently fourth round of the competition, there are three code-based candidates: Bit Flipping Key Encapsulation (BIKE) [7], Classic McEliece [4], and Hamming Quasi-Cyclic (HQC) [3]. Compared to, for instance, lattice-based cryptography, code-based cryptosystems suffer from large and impractical key sizes used to ensure security. Recently, with the scope of reducing the key size other metrics, like the rank metric and the Lee metric, have been introduced to code-based cryptography [1, 2, 108]

Code-based cryptosystems work in the following fashion. The goal is to communicate a message m in a way that only authorized parties can read it. For this we use an error-correcting code $\mathcal{C}$ able to correct t errors as a private key. To encode the message m we use the presentation of $\mathcal{C}$ in terms of a matrix called a *generator matrix* denoted by G. Then the encoded message is given by $c = mG + e$ where e is an error vector of weight t. To decode c into the original message m an efficient decoding algorithm suitable for the code $\mathcal{C}$ with generator matrix G is needed. The necessary information for an efficient decoder is the code's generator matrix. Since using G as a public key to the communication system would authorize anyone to read the message, we have to find a way to hide the generator matrix G and hence the code's structure but still provide a public key to ensable the encoding of the message m. To obtain the public key, we disguise the code by transforming either its generator matrix or its parity-check matrix (depending on the perspective). This transformed version is again a linear code $\mathcal{C}'$ and serves as the public key. Additionally, $\mathcal{C}'$ should behave like a random code, i.e., should not show any characteristic of the secret code $\mathcal{C}$. This prevents an attacker from gaining insights about the structure of the private key $\mathcal{C}$. Eventually, everyone can encrypt a message by means of the public key and add some intentional errors (but no more than t). However, only authorized parties having access to the private key can efficiently decrypt the encrypted message back into a codeword of $\mathcal{C}$, which is efficiently decodable.

The first code-based cryptosystem dates back to 1978 and was introduced by McEliece [93]. Note that the scheme is as old as RSA and is still unbroken. Its robustness against attacks is decisive for the trust in code-based cryptosystems and the continuous research in this area. In fact, Classic McEliece, based on McEliece's original system, reached the fourth round of the standardization process introduced by NIST. The system's security relies on the hardness of the underlying coding problem (i.e., the hardness of decoding a random linear code). In classical coding theory we usually consider a finite field $\mathbb{F}_q$ of q elements which, from a cryptographic viewpoint, represents a q-ary input alphabet.

McEliece's cryptosystem suggests to disguising the generator matrix G by computing $G' = SGP$, where S is a random invertible matrix and P is a random permutation matrix. In this way we obtain an equivalent code whereas the weight of the error vectors introduce remain invariant. The public key is given by the disguised code $\mathcal{C}'$ with generator matrix G' which looks seemingly random and does hence not reveal the structure of the secret code $\mathcal{C}$ whereas the secret key consists of the matrices G, S and P. This assures that authorized parties (with the knowledge of the secret key) are able to decode the original message. An equivalent scheme using a parity-check matrix representation of the code has been proposed by Niederreiter [97]. Originally, McEliece used a special family of codes called Goppa Codes. Many variants of the scheme based on other code families (such as LDPC, quasi-cyclic codes and many more) have been proposed [7, 3]. Depending on the codes' structure, the security level of the resulting cryptosystems varies. More structure usually means better algorithmic efficiency (i.e., decoding and encoding is highly efficient) but on the other hand also potentially longer key sizes and more information to hide when disguising the generator matrix or parity-check matrix.

Code-based cryptosystems, usually, are based on the hardness of the generic decoding problem when using a generator matrix, or equivalently on the syndrome decoding problem when using a parity-check matrix to represent the code (representing the code's kernel). For the scope of this thesis we focus on the syndrome decoding problem. Given a received word $r = mG+e$ composed of the original message m and an error e, the goal is to recover either m or e. Using a parity-check matrix H to represent the code defining the code's kernel, we can consider equivalently $rH^\top = eH^\top$. We refer to $rH^\top =: s$ as the *syndrome*. The syndrome decoding problem then is stated as follows.

Problem 1.0.1 (Syndrome Decoding Problem)**.** Given a linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ of dimension k with a parity-check matrix H, a positive integer $t \in \mathbb{N}$ and a vector $s \in \mathbb{F}_q^{n-k}$. Find a vector $e \in \mathbb{F}_q^n$ of weight t, such that $s = eH^\top$.

The hardness of solving the sydrome decoding problem is highly dependent on the weight of the error vector and the underlying metric. It has been shown that the syndrome decoding problem is NP-hard in the Hamming metric [14, 22] and the Lee metric [130]. If we wanted to solve this problem via a brute-force algorithm (that is going through all possible vectors $e \in \mathbb{F}_q^n$), the cost would result in

$$\binom{n}{t}(q-1)^t(n-k)t.$$

However, this cost can be reduced using more efficient algorithms. The fastest algorithm to tackle the syndrome decoding problem is Information Set Decoding (ISD) introduced by Prange in 1962 [104]. The high-level idea of Prange's algorithm is to transform the syndrome decoding problem using linear algebra by permuting and scrambling the parity-check matrix in such a way that the last k columns of the transformed parity-check matrix form an information set, i.e., index set representing the whole code. We then apply the same transformation to the error vector and hope for the nonzero entries to lie in the first few positions. Assuming, additionally, that the information set is error-free, the original message can be recovered. Ever since, a large list of improvements has been published (see [21, 26, 36, 37, 39, 54, 84, 85, 92, 125]). However, the cost has only been reduced little and is considered stable. In [129] Lee-metric variants of the existing ISD algorithms have been proposed and compared.

1.1 Organization

We start by introducing the most basic concepts and results needed throughout the thesis in Chapter 2. One of the main focus is channel coding and, especially, codes over integer residue rings. We introduce the entropy and its properties to be able to understand and estimate the channel's error correction capability. We furthermore discuss the method of types in this chapter indicating the most probable sequence over a given alphabet and under given circumstances. We will use this method to elaborate the marginal distribution of a given channel model. The chapter is concluded by introducing classical coding theory over finite fields endowed with the Hamming metric.

Given this background, we then introduce the Lee metric and Lee-metric codes in Chapter 3. As the Lee metric is defined over integer residue rings, we will give a formal definition of linear codes over integer residue rings and will discuss their properties and bounds (such as Singleton-like bounds and the Gilbert-Varshamov bound). As we will see, the existing Singleton-like bounds in the Lee metric are far from being tight.

In Chapter 4 we provide new bounds on the minimum Lee distance of a code defined over an integer residue ring $\mathbb{Z}/p^s\mathbb{Z}$ introducing several novel notions of generalized Lee weights. The first generalized weights introduced in Section 4.2 are a straightforward adaption of the generalized Hamming weights over rings, for which we will discuss their suitability in the Lee metric. In Section 4.3 we define the generalized weights over the columns of a generator matrix instead of defining it over every codeword. Lastly, we abandon the classical definition of generalized weights and make use of the natural properties of the integer residue ring $\mathbb{Z}/p^s\mathbb{Z}$ in Section 4.4.

In Chapter 5 we introduce two channel models in the Lee metric; a discrete memoryless channel and a block-channel introducing an error of fixed Lee weight to the message sent. The channel models are introduced in Section 5.1 where we discuss their motivation and properties. In Section 5.2 we estimate the error probability of both the channel models using a union bound argument based additionally on the size of given spheres and balls in the Lee metric. We bound the sphere and ball sizes using the entropy of the marginal distribution of the channel models introduced and show that the bound is asymptotically tight. To conclude the chapter, we give an algorithm that constructs a vector of fixed length and Lee weight in a uniform way among all such vectors. For these vectors we discuss the problem of reducing or increasing its Lee weight by multiplying every entry by a nonzero scalar.

Chapter 6 focuses on random regular LDPC code ensembles defined over $\mathbb{Z}/q\mathbb{Z}$, for any positive integer q. We derive an expression for the average weight enumerator in Section 6.2 and discuss its asymptotic growth rate. On the one hand this serves to understand the possible minimum Lee distance of a randomly chosen LDPC code in this ensemble. On the other hand, the average weight enumerator itself is crucial to understand the error-correction in the error floor region (i.e., the region of relatively small error weight). The error-correction performance in terms of the weight enumerator is discussed in Section 6.3 together with numerical results and supported by density evolution analysis.

The theoretical part of the thesis is finalized with an application to code-based cryptography. Namely, we apply the knowledge of the marginal distribution of a vector of fixed weight to information set decoding presented in Chapter 7.

Lastly, we give some concluding remarks and related open problems in Chapter 8.

1.2 Overview of Results

This thesis consists of novel results for linear Lee-metric codes in their algebraic structure and a in more information theoretic sense.

We start by the basic structure of linear codes in the Lee metric. With the aim of building a solid basis for optimal codes in the Lee metric, we propose improved bounds on the minimum Lee distance in Chapter 4 and more explicitly in Sections 4.2 - 4.4. The chapter is based on the paper [19]:

Better Bounds on the Minimum Lee Distance
by Jessica Bariffi and Violetta Weger
submitted to SIAM Journal of Discrete Mathematics
available as arXiv preprint arXiv:2307.06079, 2023.

In Section 5.1 we introduce a discrete memoryless channel and a block-wise constant-weight channel in the Lee metric. The motivation for the latter channel model comes from a cryptographic point of view (for the constant-weight channel) where errors of fixed weight are intentionally introduced. We then also introduced a memoryless counterpart to this channel model. The main result of this chapter lies in the derivation of the marginal distribution of the constant-weight channel, Lemma 5.1.4. The result allows us to bound sizes of spheres which we use to analyze the block error probabilities of both channel models. With the motivation of introducing errors of fixed Lee weight, we provide an algorithm to construct such a vector randomly in Section 5.3.1. Additionally, we discuss the impact of a scalar multiplication to the Lee weight of a vector and show (Theorem 5.3.9) that reducing the Lee weight of a vector in the limit of large block length is impossible when working over $\mathbb{Z}/p\mathbb{Z}$ where p is a prime number. The results in Chapter 5 is based on the papers [15, 16, 17]:

Analysis of Low-Density Parity-Check Codes over Finite Integer Rings for the Lee Channel
by Jessica Bariffi, Hannes Bartz, Gianluigi Liva and Joachim Rosenthal
in IEEE Global Communications Conference, 2022.

On the Properties of Error Patterns in the Constant Lee Weight Channel
by Jessica Bariffi, Hannes Bartz, Gianluigi Liva and Joachim Rosenthal
in International Zurich Seminar on Information and Communication, 2022.

Error-Correction Performance of Regular Ring-Linear LDPC Codes over Lee Channels
by Jessica Bariffi, Hannes Bartz, Gianluigi Liva and Joachim Rosenthal
submitted to IEEE Transactions on Information Theory
available as arXiv preprint arXiv:2312.14674, 2023.

We then study regular LDPC code ensembles with distance properties in the Lee metric. In Section 6.2 we derive a formula for the expected weight enumerator of a code in a regular LDPC code ensemble and give its asymptotic growth rate. This allows to estimate the error probability in the error floor regime (i.e., the regime of small error weight) and to understand the smallest possible minimum Lee distance a code of the ensemble can admit.

For the decoding performance we focus on belief propagation and the symbol message passing decoder. In Section 6.3 we analyze the performance of selected LDPC code ensembles over the channels introduced in Chapter 5. Chapter 6 is based on the results found in [15, 17]:

Analysis of Low-Density Parity-Check Codes over Finite Integer Rings for the Lee Channel
by Jessica Bariffi, Hannes Bartz, Gianluigi Liva and Joachim Rosenthal
in IEEE Global Communications Conference, 2022.

Error-Correction Performance of Regular Ring-Linear LDPC Codes over Lee Channels
by Jessica Bariffi, Hannes Bartz, Gianluigi Liva and Joachim Rosenthal
submitted to IEEE Transactions on Information Theory
available as arXiv preprint arXiv:2312.14674, 2023.

Lastly, in Sections 7.2 and 7.3 we improve the yet fastest Lee-information set decoding variant based on the algorithm presented by Becker Joux, May and Meurer [21]. We use the marginal distribution (see Chapter 5, Lemma 5.1.4) to improve the complexity of the algorithm. The results in both sections are based on the paper [18]:

Information Set Decoding for Lee-Metric Codes using Restricted Balls
by Jessica Bariffi, Karan Khathuria and Violetta Weger
Code-Based Cryptography 10th International Workshop, CBCrypto 2022, Revised Selected Papers, 2022.

Chapter 2

Preliminaries

In this chapter we provide the necessary background needed for the remainder of this dissertation. The chapter consists of two main areas forming the essential background, namely, information theory and coding theory. Hence, in a first part we start with an introduction to information theory and introduce the entropy and the method of types in Sections 2.1 and 2.2, respectively. In Section 2.3 we give an introduction to classical coding theory over finite fields and endow it with the Hamming metric. We give the fundamental results needed, such as the Singleton-bound and the error-correction capability of a code. The interested reader is referred to [45, 59] and [90, 112, 127] for more details on information theory and coding theory, respectively.

Both areas cover a large spectrum of mathematical disciplines, such as algebra, complex analysis, probability theory and combinatorics. Especially the basics of probability theory and combinatorics are common tools we make use of. In the course of the thesis we often use (discrete) random variables and their probability distributions. More formally, we denote by X a random variable over a discrete alphabet $\mathcal{X}$ and let $x \in \mathcal{X}$ be its realization. For every $x \in \mathcal{X}$, we will denote the probability distribution of X by

$$P_X(x) := \mathbb{P}(X = x).$$

We omit the subscript X and simply write $P(x)$, if it is clear from the context.

We often use counting arguments to describe the cardinality of a sphere or a ball of given dimension and radius. Counting elements is an elementary task of combinatorics. For positive integers $n, k, k_1, \ldots, k_s \in \mathbb{N}$ with $\sum_{i=1}^{s} k_i = n$ the *binomial coefficient* and *multinomial coefficient*, respectively, are defined as

$$\binom{n}{k} := \frac{n!}{(n-k)!k!}, \quad \text{and} \quad \binom{n}{k_1, \ldots, k_s} := \frac{n!}{k_1! \cdots k_s!}.$$

Another theory used to lay the connection between combinatorics, analysis and other mathematical fields is the theory of (ordinary) generating functions. Generating functions enable us to write a sequence of integers in terms of a series (or polynomial). A single element of the sequence then represents a coefficient of the sequence (or polynomial).

Definition 2.0.1 (Generating Function)**.** Let $(a_n)_{n\in\mathbb{N}}$ be a sequence of numbers depending on an integer n. A *generating function* A of the sequence is $(a_n)_{n\in\mathbb{N}}$ defined to be

$$A(x) = \sum_{n\geq 0} a_n \cdot x^n.$$

Theorem 2.0.2 shows how to retrieve a coefficient a_i of the sequence $(a_n)_{n\in\mathbb{N}}$.

Theorem 2.0.2 (Taylor's Theorem)**.** *If $A(x) = \sum_{n\geq 0} a_n \cdot x^n$ is a generating function for a sequence $(a_n)_{n\in\mathbb{N}}$ then for every $i \in \mathbb{N}$*

$$A_i = \frac{A^{(i)}(0)}{i!},$$

where $A^{(i)}(0)$ denotes the i-th derivative of the generator function $A(x)$ evaluated at $x = 0$.

For instance, it is well-known that the sequence $((1+x)^n)_{n\in\mathbb{N}}$ can be expressed by the ordinary generating function involving the binomial coefficient. That is,

$$(1+x)^n = \sum_{k=0}^{\infty} \binom{n}{k} x^k.$$

This result is generally known as the *Binomial Theorem* [139].

2.1 Entropy

One of the key measures in information theory is the entropy. It indicates the degree of uncertainty contained in a random variable and is used to characterize the achievable transmission rates of a communication channel. Intuitively, the more likely an event is to happen, the smaller is the surprise of the outcome, i.e., the entropy of this event is small. In a more probabilistic setting, consider an event E that happens with a probability $p(E)$. If this probability is close to 1 and hence the event E is very likely to happen, the entropy of E will be close to zero. On the contrary, if $p(E)$ is close to zero, then the entropy of E is high.

In the course of this chapter let X and Y be two random variables over the two discrete alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively.

Definition 2.1.1 (Entropy)**.** The binary *entropy* of a discrete random variable X with probability mass function $P_X(x)$ is defined as

$$H(X) = -\sum_{\substack{x\in\mathcal{X}\\ P_X(x)\neq 0}} P_X(x)\log_2(P_X(x)).$$

By convention, we set the entropy $H(X) = 0$ whenever $P_X(x) = 0$. Whenever it is clear that the probability mass function P corresponds to a random variable X, we will often write $H(P)$ for the entropy of X. If we use the logarithm to some base b we will write $H_b(X)$ for the entropy. Due to the base change property of the logarithm, we can always deduce $H_b(X)$ from $H_a(X)$ for some bases a and b by observing that

$$H_b(X) = \log_b(a)H_a(X).$$

Example 2.1.2. A famous example is the entropy of a (biased) flipping of a coin. Assume we consider a binary random variable $X \in \{0,1\}$ with probability $P_X = (\mathbb{P}(X=0), \mathbb{P}(X=1)) = (1-p, p)$ for some $p \in [0,1]$. The entropy of X is then given By

$$H(X) = -p\log(p) - (1-p)\log(p) =: H(p). \tag{2.1}$$

Figure 2.1 shows the binary entropy function with respect to the probability p. If $p = 1/2$ both outcomes are equiprobable. Hence, the outcome is the most uncertain and therefore the entropy is maximal and hence equal to 1. Similarly, it shows that for $p = 0$ or $p = 1$ the random variable X is deterministic and there is no uncertainty about its outcome. Thus, its entropy is equal to zero.

The definition of the entropy of a random variable can be extended to a pair of random variables X and Y. Note that this can be considered as a random variable defined on the Cartesian product $\mathcal{X}\times\mathcal{Y}$.

Definition 2.1.3 (Joint Entropy)**.** Let X and Y be two discrete random variables defined over $\mathcal{X}$ and $\mathcal{Y}$, respectively. Denote by (X,Y) the joint probability with probability mass function $P(x,y)$. Then we define the *joint entropy* of X and Y as

$$H(X,Y) := -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log_2(P(x,y)).$$

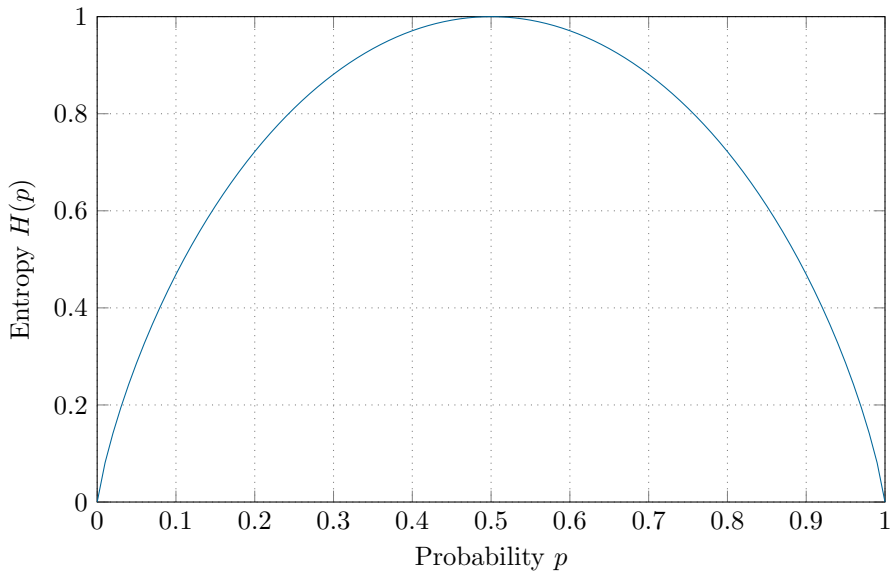

FIGURE 2.1: Entropy function $H(p)$ defined in (2.1) with respect to the probability $p \in [0, 1]$.

The *conditional entropy* of Y given X is defined by

$$H(Y|X) = -\sum_{x \in \mathcal{X}} H(Y|X = x).$$

It directly follows from the definition that the joint entropy of two random variables is symmetric with respect to the arguments, that is

$$H(X, Y) = H(Y, X).$$

We observe that the following relation between the conditional entropy and the joint entropy of two random variables X and Y holds.

$$H(X, Y) = H(Y|X) - H(X) = H(X|Y) - H(Y)$$

Theorem 2.1.4 gives a formula for the conditional entropy for n random variables.

Theorem 2.1.4 (Chain Rule Property [45, Theorem 2.5.1])**.** *Let $X_1, \ldots, X_n$ be a sequence of random variables drawn according to a probability distribution $P(x_1, \ldots, x_n)$. Then the entropy of the sequence satisfies*

$$H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i \mid X_{i-1}, \ldots, X_1) \leq \sum_{i=1}^{n} H(X_i)$$

where equality holds if and only if the X_i are independent.

Let us introduce a measure of the distance between two probability distributions. In particular, it measures the inefficiency of assuming a distribution Q when the true distribution is P.

Definition 2.1.5 (Relative Entropy)**.** Let $P(x)$ and $Q(x)$ be two probability mass functions over the alphabet $\mathcal{X}$. The *relative entropy* between P and Q is defined to be

$$\mathsf{D}(P \,||\, Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right).$$

In literature, the relative entropy is sometimes also referred to the *Kullback–Leibler distance* between two probability mass functions. If $P \neq 0$, we make use of the convention

$\lim_{Q \longrightarrow 0} \mathsf{D}(P \,||\, Q) = \infty$. If, additionally $P(x) = 0$, we set $\mathsf{D}(P \,||\, Q) = 0$. One disadvantage of the Kullback-Leibler divergence is that it is not symmetric. If a point is outside the support of Q, the Kullback-Leibler divergence grows large. To cope with this fenomena we add some random noise to Q. However, this introduces a degree of error and a lot of noise is often needed for convergence. The relative entropy serves inter alias as a separability measures and is often used in statistics and information theory. It emerges naturally from the study of maximum likelihood estimation and is well suited for use with product measures.

An alternative measure of the similarity of two probability distributions is the total variation distance. Here, we define the distance only for discrete probability distributions. We follow the description of [135, Proposition 5.2] and define the total variation distance between two distributions P and Q over $\mathcal{X}$ as

$$\mathsf{TV}(P, Q) := \frac{1}{2} \sum_{x \in \mathcal{X}} |\, P(x) - Q(x) \,|\,.$$

In contrast to the relative entropy, the total variation distance is symmetric. While the relative entropy is only bounded from below by zero, the total variation distance lies in the interval $[0, 1]$ and can be bounded from above using the relative entropy in terms of the Pinkser inequality [101] as

$$\mathsf{TV}(p, Q) \leq \sqrt{\frac{1}{2} \mathsf{D}(P \,||\, Q)}.$$

2.2 Typicality

Throughout this subsection we consider a sequence of n random variables $X_1, \ldots, X_n$ in an alphabet $\mathcal{X}$. By x we denote the sequence of realizations $x_1, \ldots, x_n$ of the random variables $X_1, \ldots, X_n$, respectively. First, we explain the general concept of typicality in terms of the method of types which analyzes the empirical distribution of a given sequence of symbols. In a next step, given a set E of sequences satisfying a common property, we study the most probable type among these sequences captured in the set of typical sequences. We will discuss the type of a typical sequence and show that the empirical distribution of a sequence in E converges exponentially fast to the type of the typical sequence identified.

2.2.1 Method of Types

The method of types was developed in [46]. It is a powerful tool to compute the set of sequences with the same empirical distribution which we will refer to as *type* in this context. For each element in the alphabet we can define its type as follows.

Definition 2.2.1 (Type)**.** The *type*, θ_x, of a sequence $x \in \mathcal{X}^n$ is the relative proportion of occurrences of each symbol, i.e., for each $a \in \mathcal{X}$,

$$\theta_x(a) = \frac{|\,\{i \in \{1, \ldots, n\} \mid x_i = a\}\,|}{n}.$$

It is obvious that θ_x defines a probability mass function on the alphabet $\mathcal{X}$. We will sometimes also refer to the type as the *empirical distribution* of a sequence.

Example 2.2.2. Let $\mathcal{X} = \{0, 1\}$ and consider the sequence $x = (1, 0, 1) \in \mathcal{X}^3$. Then the type θ_x of x is given by

$$\theta_x(0) = \frac{1}{3} \quad \text{and} \quad \theta_x(1) = \frac{2}{3}. \tag{2.2}$$

Let $\mathcal{T}(\mathcal{X}^n)$ denote the set of all possible types of sequences of length n over $\mathcal{X}$. Note that, in Example 2.2.2, $x = (1, 0, 1)$ is not the only sequence in $\{0, 1\}^3$ of type θ_x described in (2.2). Thus, over an alphabet $\mathcal{X}$, we define the set of sequences of the same type $\theta \in \mathcal{T}(\mathcal{X}^n)$ by

$$T_\theta^{(n)} := \{x \in \mathcal{X} \mid \theta_x = \theta\}\,,$$

which we refer to as *type class*. The number of sequences of length n over $\mathcal{X}$ of a given type θ is determined by the number of permutations which in turn is defined by the multinomial coefficient based on the number of occurrences of each symbol $a \in \mathcal{X}$, i.e.,

$$\left| T_\theta^{(n)} \right| = \binom{n}{n\theta(a_1), \ldots, n\theta(a_{|\mathcal{X}|})} =: \binom{n}{n\theta}. \tag{2.3}$$

Given a type (or empirical distribution) $\theta := (\theta_1, \theta_2, \ldots, \theta_{|\mathcal{X}|})$ over $\mathcal{X}$, for any positive integer n, we have [45, Theorem 11.1.3]

$$\frac{1}{(n+1)^{|\mathcal{X}|}} 2^{nH(\theta)} \leq \binom{n}{n\theta} \leq 2^{nH(\theta)}. \tag{2.4}$$

For the probability of a type class $T_\theta^{(n)}$ a similar result holds.

Theorem 2.2.3 (Probability of Type Class)**.** *For any type $\theta \in \mathcal{T}(\mathcal{X}^n)$ and any distribution P the following bounds hold for the probability of the type class $T_\theta^{(n)}$.*

$$\frac{1}{(n+1)^{|\mathcal{X}|}} 2^{-nD(\theta \,||\, P)} \leq \mathbb{P}\left(T_\theta^{(n)}\right) \leq 2^{-nD(\theta \,||\, P)}.$$

Assume now that the random variables $X_1, \ldots, X_n$ are identically distributed with distribution P_X over $\mathcal{X}$ and let $P_X^{(n)}$ be the distribution of a length-n sequence whose entries are drawn independently according to X_i. Theorem 2.2.4 gives an expression for the distribution $P_X^{(n)}$.

Theorem 2.2.4. *If $X_1, \ldots, X_n$ are independent and identically distributed $\sim P_X$ random variables and x a sequence of realizations, then the probability of x depends on its type θ_x only and is given by*

$$P_X^{(n)} = 2^{-n[H(\theta_x) + D(\theta_x \,||\, P_X)]}.$$

2.2.2 Typical Sequences

Consider the sequences $x \in \mathcal{X}^n$ for a given alphabet $\mathcal{X}$ and a positive integer n. We have seen in (2.3) that the cardinality of the set of sequences of a type $\theta \in \mathcal{T}(\mathcal{X}^n)$ highly depends on θ. Thus, if we were to sample a sequence $x \in \mathcal{X}^n$ uniformly at random, some types are more probable for x to occur than others. The asymptotic equipartition property [45, Theorem 3.1.1] is the information-theoretic analogue to the law of large numbers and allows us to define two categories of sequences in $\mathcal{X}^n$: The set of *typical sequence* and the set of non-typical sequences. Formally, the set of typical sequences are sequences with a sample entropy that converges to the true entropy.

Definition 2.2.5 (Typical Set)**.** Given $\varepsilon > 0$ and independent and identically distributed random variables $X_1, \ldots, X_n$ having distribution Q. We define the *typical set* A_θ^ε of sequences with type θ as

$$A_\theta^\varepsilon := \left\{ x \in T_\theta^{(n)} \mid D(\theta \,||\, Q) \leq \varepsilon \right\}.$$

Remark 2.2.6. Note that the probability that a sequence x is not typical is $1 - \mathbb{P}\left(x \in A_\theta^\varepsilon\right)$, and we can upper bound it, using Theorem 2.2.3, by

$$1 - \mathbb{P}\left(x \in A_\theta^\varepsilon\right) = \sum_{\substack{\theta \in \mathcal{T}(\mathcal{X}^n) \\ D(\theta \,||\, Q) > \varepsilon}} \mathbb{P}\left(T_\theta^{(n)}\right) \leq \sum_{\substack{\theta \in \mathcal{T}(\mathcal{X}^n) \\ D(\theta \,||\, Q) > \varepsilon}} 2^{-nD(\theta \,||\, Q)} = 2^{-n\left[\varepsilon - |\mathcal{X}| \frac{\log(n+1)}{n}\right]}.$$

As $n \longrightarrow \infty$, we have $2^{-n\left[\varepsilon - |\mathcal{X}| \frac{\log(n+1)}{n}\right]} \longrightarrow 0$, and thus,

$$\mathbb{P}\left(x \in A_\theta^\varepsilon\right) \overset{n \longrightarrow \infty}{\longrightarrow} 1.$$

We can then deduce that the empirical distribution of a sequence of realizations converges to the distribution of the random variables.

Theorem 2.2.7 (Convergence of Empirical Distribution)**.** *Let $X_1, \dots, X_n$ be a sequence of independent and identically distributed random variables with distribution Q. Let $x = (x_1, \dots, x_n)$ be the realization of the sequence of the random variables and let θ_x be its type. Then it holds that*

$$\mathbb{P}(D(\theta_x \,||\, Q) > \varepsilon) \leq 2^{-n\left[\varepsilon - |\,\mathcal{X}\,| \frac{\log(n+1)}{n}\right]}$$

and hence, $D(\theta_x \,||\, Q) \longrightarrow 0$ almost surely as $n \longrightarrow \infty$.

Let us now assume that we sample sequences from a given probability distribution. To understand the probability of an observed sequence belonging to the set of atypical (or typical) set we use Sanov's Theorem [113]. Typicality is always referred to a common property defining the sequences observed. Therefore, let E denote a subset of probability distributions (or set of types) in $\mathcal{X}^n$ defining such a common property. For instance, E could be the set of probability distributions with a given expected value m. Then the probability that the empirical distribution of a sampled sequence falls into the set E is captured in Sanov's Theorem.

Theorem 2.2.8 (Sanov's Theorem [113])**.** *Given a set of probability distributions E over an alphabet $\mathcal{X}$. Let $X_1, \dots, X_n$ be n independent and identically distributed random variables drawn according to a distribution Q over $\mathcal{X}$, not necessarily included in E. Let $x = (x_1, \dots, x_n)$ denote the sequence of realizations of the random variables with type θ_x. Furthermore, we denote by $P^\star$ the distribution in E closest (in relative entropy) to the distribution Q. Then the probability that the type of x belongs to the set E is upper bounded by*

$$\mathbb{P}(\theta_x \in E) \leq (n+1)^{|\,\mathcal{X}\,|} 2^{-\mathsf{D}(P^\star \,||\, Q)}.$$

In particular, if E is a closed set it holds

$$\lim_{n \longrightarrow \infty} \frac{1}{n} \log\left(\mathbb{P}(\theta_x \in E)\right) = -\mathsf{D}(P^\star \,||\, Q).$$

Sanov's Theorem hence tells us that the probability of observing an empirical distribution that belongs to the set E is exponentially equivalent to $2^{-\mathsf{D}(P^\star \,||\, Q)}$, meaning that

$$\lim_{n \to \infty} \frac{1}{n} \log\left(\frac{\mathbb{P}(\theta_x \in E)}{2^{-\mathsf{D}(P^\star \,||\, Q)}}\right) = 0\,.$$

A simple way to find the distribution $P^\star = \operatorname{argmin}_{P \in E} \mathsf{D}(P \,||\, Q)$ is using Lagrange multipliers. Hence, we compute the distribution minimizing the relative entropy $\mathsf{D}(P \,||\, Q)$ constrained to the shared condition the distributions in E must fulfill.

In a nutshell, the probability of types under a given distribution Q over an alphabet $\mathcal{X}$ is determined by the probability of the distribution closest (in relative entropy) to Q which, by applying Theorem 2.2.3, is given by $2^{-\mathsf{D}(P^\star \,||\, Q)}$. Thus, the probability of observing a type in set E is the same as the probability of observing the type closest to $P^\star$.

The statement of Sanov's theorem can further be strengthened. In fact, considering the same scenario as for Sanov's Theorem, it holds that the probability of observing any other type is negligible and moreover, types that are far away from Q are exponentially less likely to be observed. This result is stated in Theorem 2.2.9 introduced as the *Conditional Limit Theorem* [45, Theorem 11.6.2].

Theorem 2.2.9 (Conditional Limit Theorem [45])**.** *Let E be a closed convex subset of probability distributions over a given alphabet $\mathcal{X}$ and let Q be a distribution not in E over the same alphabet $\mathcal{X}$. Consider $X_1, \dots, X_n$ to be discrete random variables drawn independent and identically distributed $\sim Q$ and let $P^\star = \arg\min_{P \in E} D(P \,||\, Q)$. Denote by x the sequence of realizations given by the random variables and θ_x its empirical distribution. Then, for any $a \in \mathcal{X}$ and for any $i = 1, \dots, n$,*

$$\mathbb{P}\left(X_i = a \,|\, \theta_x \in E\right) \longrightarrow P^\star(a)$$

in probability as n grows large.

Hence, the empirical distribution of the random variables X_i of a random sequence $X_1, \ldots, X_n$ with type in E converges exponentially fast to the distribution $P^\star$ closest to Q.

2.3 Coding Theory

We are now introducing the second main topic of this thesis: coding theory. Coding theory and information theory are strongly related, and we can view coding theory as a direct application of information theory. In fact, the theory of error-correcting codes started with Shannon's seminal work [117] where he showed that error-correcting codes of relatively low rate (i.e., a rate that is smaller than the channel's capacity) allows for the transmission of discrete data with nearly no error. Classical coding theory studies the properties and error-correction capability of (linear) block codes defined over finite fields endowed with the Hamming metric [70]. However, motivated by applications, for instance, in modern code-based cryptography, alternative metrics such as the rank metric [48] or the Lee metric [83] have gained more attention in the last decades.

Classical coding theory is about error correction in a communication model. It plays a role in the channel encoding and channel decoding part of the communication. With the rationale given in Chapter 1, we consider a finite field $\mathbb{F}_q$ and a communication model as shown in Figure 2.2.

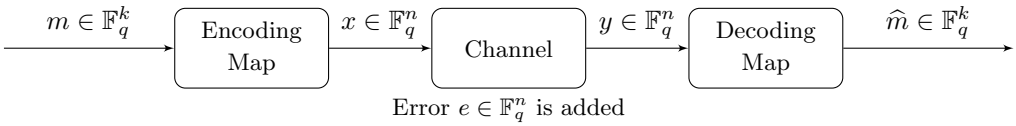

FIGURE 2.2: Message transmission model.

The errors during the transmission through the channel are introduced according to a given transition probability which we sometimes refer to as the *channel law*. The channel law is the probability observing the channel output $y \in \mathbb{F}_q^n$, given that the channel input was $x \in \mathbb{F}_q^n$, and we denote it as

$$P_{Y \mid X}(y \mid x) := P_{Y \mid X}(Y = y \mid X = x). \tag{2.5}$$

For a channel model there are two main distinctions: the memoryless channel, transmitting symbol by symbol independently of the previous symbol and with the same probability, and a conditioned transition where the symbols are transmitted with a probability depending on the output of the preceeding symbols. The first channel model, due to its simple application, is widely used and studied. Let us hence define this channel more explicitly.

Definition 2.3.1 (Discrete Memoryless Channel)**.** Consider a channel with input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$. Assume that $x \in \mathcal{X}^n$ is transmitted and $y \in \mathcal{Y}^n$ is received. A channel is called *discrete memoryless*, if the input alphabet $\mathcal{X}$ and the output alphabet $\mathcal{Y}$ are discrete, finite sets and the output $Y_i = y_i$ at time i only depends on the input $X_i = x_i$ at that time i, i.e.,

$$P_{Y \mid X}(y_1, \ldots, y_n \mid x_1, \ldots, x_n) = \prod_{i=1}^{n} P_{Y_i \mid X_i}(y_i \mid x_i).$$

After defining the channel with its transition probability, we focus on the decoding part of the model in Figure 2.2. There are different decoding rules that can be considered. In this thesis we merely focus on *maximum likelihood* decoding. Considering now blocks of length n, maximum likelihood describes the decoding rule "decode to the most likely codeword". Let us denote by E the random variable according to the error introduced in the channel. Let

$\widehat{x}$ denote the decoded word. Similarly to the channel law defined in (2.5), we can define the following two probabilities.

1. The *conditional error probability* of the decoder is

$$\mathbb{P}(E \mid y) := \mathbb{P}(\widehat{x} \neq x \mid y),$$

   where $\hat{x}$ is an estimate of the codeword x that was transmitted.

2. The *error probability* of the decoder is defined as

$$\mathbb{P}(E) := \sum_{y \in \mathcal{Y}^n} \mathbb{P}(E \mid y)\mathbb{P}(y).$$

Maximum likelihood decoding then translates into minimizing the conditional error probability $\mathbb{P}(E \mid y)$ given the channel output y. In our set-up, we consider a symmetric channel where every channel input is equally likely. Hence, for a given $x \in \mathcal{X}$, $\mathbb{P}(x)$ is constant and thus for every channel output $y \in \mathcal{Y}$ is too. Then minimizing the conditional error probability is equivalent to maximizing $\mathbb{P}_{Y \mid X}(y \mid x)$.

2.3.1 Linear Block Codes over Finite Fields

Let us now focus on the encoding part by properly defining codes and their properties.

Definition 2.3.2 (Linear Code)**.** Given $1 \leq k \leq n$, an $[n, k]$ *linear code* over $\mathbb{F}_q$ is a k-dimensional subspace $\mathcal{C} \subseteq \mathbb{F}_q^n$. The parameter n is referred to as the *block length* of the code $\mathcal{C}$ and k defines its *dimension*. The elements of $\mathcal{C}$ are called *codewords*.

Given a codeword $c \in \mathcal{C} \subseteq \mathbb{F}_q^n$, we say that c has k *information bits*. The remaining $n-k$ bits are redundant. The fraction of information bits per block length is called the *rate* of the code, defined by

$$R := \frac{\log_q(|\,\mathcal{C}\,|)}{n} = \frac{k}{n}.$$

Since k is the dimension of an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$, $\mathcal{C}$ has k linearly independent codewords spanning the whole code. Therefore, every codeword of $\mathcal{C}$ is a linear combination of the k basis vectors. Over $\mathbb{F}_q$ this means that $\mathcal{C}$ contains $|\,\mathcal{C}\,| = q^k$ distinct codewords. Thus, for practical reasons, we use a representation in terms of a matrix whose rows are formed by a basis of $\mathcal{C}$.

Definition 2.3.3 (Generator Matrix)**.** A matrix $G \in \mathbb{F}_q^{k \times n}$ is called a *generator matrix* of an $[n, k]$ linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ if its rows form a basis of $\mathcal{C}$.

From a communication model point of view, the generator matrix is needed to transform a message $m \in \mathbb{F}_q^k$ into a codeword $c \in \mathcal{C}$. We call this step the *encoding* of the message. Note that a generator matrix is not unique as there may exist more than one basis spanning the same subspace. Furthermore, given an $[n, k]$ linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with generator matrix G, we have that the code $\mathcal{C}$ is characterized by the image of G, i.e.,

$$\mathcal{C} = \left\{xG \mid x \in \mathbb{F}_q^k\right\}.$$

We say that a generator matrix of $\mathcal{C}$ is in *systematic form*, if there exist a matrix $A \in \mathbb{F}_q^{k \times (n-k)}$ such that

$$G = \begin{pmatrix} \mathbb{I}_k & A \end{pmatrix} =: G_{\mathsf{sys}}, \tag{2.6}$$

where $\mathbb{I}_k$ denotes the $k \times k$ identity matrix. A generator matrix G admits a systematic generator matrix G_{sys}, if and only if the first columns of G are linearly independent. In order to check whether a vector $x \in \mathbb{F}_q^n$ is a codeword of $\mathcal{C}$, we define a matrix H whose kernel defines $\mathcal{C}$.

Definition 2.3.4 (Parity-Check Matrix)**.** Consider an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$. A matrix $H \in \mathbb{F}_q^{(n-k)\times n}$ is called a *parity-check matrix* of $\mathcal{C}$ if its kernel corresponds to $\mathcal{C}$, i.e.,

$$\mathcal{C} = \left\{x \in \mathbb{F}_q^n \mid xH^\top = 0\right\} = \ker(H)$$

In particular, given an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ with parity-check matrix H and a vector $x \in \mathbb{F}_q$, we can easily verify whether $x \in \mathcal{C}$ by checking if

$$xH^\top = 0.$$

Since a code $\mathcal{C}$ can be represented either by the image of a generator matrix G or by the kernel of a parity-check matrix H, it holds that

$$GH^\top = 0. \tag{2.7}$$

This implies $HG^\top = 0$ and moreover, if G is in systematic form $\begin{pmatrix}\mathbb{I}_k & A\end{pmatrix}$ then H can easily be computed by

$$H = \begin{pmatrix}-A^\top & \mathbb{I}_{n-k}\end{pmatrix} =: H_{\mathsf{sys}}.$$

In alignment with the systematic form of a generator matrix, we call H_{sys} the *systematic form* of a parity-check matrix. By Equation (2.7) we observe that H and G can play the inverted role for a specific linear code with parameters $[n, n-k]$. This code is commonly known as the dual code and is defined in the subsequent way.

Definition 2.3.5 (Dual Code)**.** Let $\mathcal{C} \subseteq \mathbb{F}_q$ be a k-dimensional linear code. The *dual code* of $\mathcal{C}$ is defined as

$$\mathcal{C}^\perp := \left\{x \in \mathbb{F}_q^n \mid xc^\top = 0 \text{ for all } c \in \mathcal{C}\right\}.$$

Thus, a generator of a code $\mathcal{C}$ is a parity-check matrix of its dual $\mathcal{C}^\perp$ and vice versa.

A linear code is capable of correcting a certain amount of errors. This amount of errors is measured using a distance. Classically, this distance is the Hamming metric, defined as follows.

Definition 2.3.6 (Hamming Weight)**.** The *Hamming weight* of a vector $x \in \mathbb{F}_q^n$ is given by the number of nonzero entries of x, i.e.,

$$\mathrm{wt}_{\mathsf{H}}(x) := |\{i = 1, \ldots, n \mid x_i \neq 0\}|.$$

Similarly, for two vectors $x, y \in \mathbb{F}_q^n$ their *Hamming distance* is given by the number of positions in which they differ. That is

$$\mathrm{d}_{\mathsf{H}}(x, y) := |\{i = 1, \ldots, n \mid x_i \neq y_i\}|.$$

We here note that the Hamming distance between two vectors x and y can be interpreted as the Hamming weight of the difference of the two vectors, meaning that $\mathrm{d}_{\mathsf{H}}(x, y) = \mathrm{wt}_{\mathsf{H}}(x - y)$. Therefore, the Hamming weight naturally induces the Hamming distance. By the definition of the Hamming weight it follows that the Hamming distance is a metric. To understand a code's performance in terms of error correction, we need to introduce another important parameter of a linear code: its minimum distance.

Definition 2.3.7 (Minimum Hamming Distance)**.** Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be an $[n, k]$ linear code. The *minimum Hamming distance* of $\mathcal{C}$ is defined to be the minimum distance between any two distinct codewords, i.e.,

$$\mathrm{d}_{\mathsf{H}}(\mathcal{C}) := \min\left\{\mathrm{d}_{\mathsf{H}}(c_1, c_2) \mid c_1, c_2 \in \mathcal{C} \text{ with } c_1 \neq c_2\right\}.$$

Again, as the Hamming distance is induced by the Hamming weight, the minimum distance of a code $\mathcal{C}$ can analogously be written as the smallest Hamming weight among all nonzero

codewords as

$$\mathrm{d_H}(\mathcal{C}) = \min\left\{\mathrm{wt_H}(c) \mid c \in \mathcal{C} \setminus \{0\}\right\}.$$

The minimum Hamming distance of a code $\mathcal{C}$ is in one-to-one correspondence with its error-correction capability.

Proposition 2.3.8 (Error-Correction Capability)**.** *Let $k \leq n$ be two positive integers and consider an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$. Given the minimum Hamming distance $d = \mathrm{d_H}(\mathcal{C})$, we can correct up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors.*

2.3.2 Bounds on Linear Block Codes

Proposition 2.3.8 thus shows that the error-correction capability of a q-ary $[n, k]$ linear code $\mathcal{C}$ grows as its minimum Hamming distance grows. Considering a communication channel, we obviously would like to use "good" codes in terms of their error-correction capability. Thus, we want to use codes with maximum minimum Hamming distance. The most famous bound capturing the trade-off between the minimum distance and the code's parameter n and k is the Singleton bound [121] introduced by Singleton in 1964.

Theorem 2.3.9 (Singleton bound)**.** *Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be an $[n, k]$ linear code with minimum Hamming distance $\mathrm{d_H}(\mathcal{C}) = d$. Then the following upper bound applies to d.*

$$d \leq n - k + 1.$$

The proof follows a puncturing argument and relies on the fact that the Hamming weight of a codeword is given by the number of nonzero positions of the codeword. In the Lee metric, this puncturing argument leads to a rather loose bound (as we will discuss in Section 3.4). In order to get a clear idea of the arguement, we now give the proof of the Singleton bound.

Proof. Since the minimum Hamming distance of the code $\mathcal{C}$ is d, any two distinct codewords c_i and c_j differ in at least d positions. Let us puncture the code $\mathcal{C}$ in $d-1$ randomly chosen positions and let us denote by $\mathcal{C}'$ the punctured code. It follows that any two punctured codewords c_i' and c_j' have a Hamming distance of at least 1. Hence, all resulting codewords are still distinct and thus

$$|\,\mathcal{C}'\,| = |\,\mathcal{C}\,| = q^k.$$

On the other hand, every codeword $c_i' \in \mathcal{C}'$ has length $n - (d-1)$ and its entries lie in $\mathbb{F}_q$. This means that there can be at most $q^{n-(d-1)}$ distinct codewords and thus

$$|\,\mathcal{C}'\,| = q^k \leq q^{n-(d-1)} \tag{2.8}$$

Solving (2.8) for the minimum Hamming distance d yields the desired result. □

Note that the Singleton bound in Theorem 2.3.9 holds for nonlinear codes too and can be shown with a similar puncturing argument.

Codes attaining the Singleton bound in the Hamming metric are called *maximum distance separable* (MDS) codes, and are well studied in the coding theory community. It is common folklore that MDS codes are dense in the limit of large field size q. That is, letting q tend to infinity, any randomly chosen linear code over $\mathbb{F}_q$ attains the Singleton bound with high probability. If we instead let the block length n tend to infinity, the situation is different. In 1955, Segre introduced in [116] the following conjecture, known as *the MDS conjecture*.

Conjecture 2.3.10 (MDS Conjecture)**.** Consider an MDS code $\mathcal{C} \subseteq \mathbb{F}_q^n$. If $q \geq 3$ is odd, then

$$n \leq q + 1.$$

On the other hand, if q is a power of 2 and $k \in \{3, q-1\}$, then

$$n \leq q + 2.$$

The conjecture implies that MDS codes are sparse in the limit of large block length n.

The Singleton bound presents necessary conditions for a code with the stated parameters to exist. Another bound implying necessary conditions on the parameters of the code such that such a code exists is the sphere-packing (or Hamming) bound. The bound gives a limit in the parameters of the code and can be interpreted as packing spheres into a space of all codewords. Hence, the bound involves the volume of an n-dimensional ball of Hamming-radius t which is given by

$$\sum_{i=0}^{t} \binom{n}{i} (q-1)^i.$$

Theorem 2.3.11 (Sphere-Packing Bound)**.** *Given an $[n,k]$ code $\mathcal{C} \subseteq \mathbb{F}_q^n$ and $t = \lfloor (d-1)/2 \rfloor$, then*

$$M \leq \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i} (q-1)^i}.$$

A sufficient condition on the existence of a code of given parameters, is captured in the Gilbert-Varshamov bound [61].

Theorem 2.3.12 (Gilbert-Varshamov Bound [61])**.** *There exists a linear $[n,k]$ code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with minimum Hamming distance $\mathrm{d}_{\mathsf{H}}(\mathcal{C}) \leq d$ if it holds That*

$$\sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i \leq q^{n-k}.$$

A second version of the Gilbert-Varshamov bound gives a lower bound on the maximal cardinality a code of given parameter can have. It states that for an $[n,k]$ code $\mathcal{C}$ over $\mathbb{F}_q$ with Hamming distance $\mathrm{d}_{\mathsf{H}}(\mathcal{C}) = d$ the cardinality of the code is bounded by

$$|\,\mathcal{C}\,| \geq \frac{q^n}{\sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i}.$$

In an asymptotic setting these bounds state the relation between the rate of the code and its relative minimum distance $\delta_{\mathsf{H},\mathsf{min}} = \mathrm{d}_{\mathsf{H}}(\mathcal{C})/n$. For lower bounds like the Gilbert-Varshamov bound, the asymptotic bound tell us the possible rates a code can have in order to meet a given error-correction capability.

Chapter 3

Introduction to the Lee Metric

The Lee metric was introduced by C. Y. Lee in 1958 in [83] to cope with phase modulation in communication. It provides an interesting alternative to the Hamming and rank metric which, for instance, are considered for orthogonal modulation and network coding, respectively. The Lee metric has later been considered and studied further by Prange [103], Massey [91], Golomb and Welch [63], Berlekamp [23] and many more. In 1967, Massey was the first one to introduce a channel "matching" to the Lee metric. In 1971, Chiang and Wolf [42] have derived all the discrete, memoryless, symmetric channels matched to the Lee metric. The Lee metric is mainly known for the celebrated result in [71], where the authors showed that some optimal non-linear binary codes can be represented as linear codes over $\mathbb{Z}/4\mathbb{Z}$ endowed with the Lee metric. It has gained more attention with its promising application to code-based cryptography. Only recently it was discovered that Lee-metric codes attain the Gilbert-Varshamov bound with high probability for the length of the code tending to infinity [32]. This aligns with famous and well studied results in the Hamming [70] and the rank metric [87]. In addition, the characterization of constant Lee weight codes, initiated by Wood [136], has only recently been completed in [34]. Recently, a first Lee metric signature scheme has been proposed in [108]. Even though, the scheme has been broken using lattice-attacks, the Lee metric is still an interesting candidate for cryptographic applications. The strong connection to lattices could possibly be a powerful tool for Lee metric schemes, for instance, in deriving a first code-based fully homomorphic encryption scheme.

In this chapter we formally introduce the Lee metric over a general integer residue ring $\mathbb{Z}/q\mathbb{Z}$ for any positive integer q. We define the Lee weight of an element as well as of an n-tuple. By abuse of notation we will call an n-tuple a vector, even though $(\mathbb{Z}/q\mathbb{Z})^n$ is not necessarily a vector space. With the scope of studying vectors of a given Lee weight in the course of this thesis and with an eye on the syndrome decoding problem in the Lee metric, we introduce the spheres and balls of a given dimension n and radius t and discuss their properties. An important task of classical coding theory is to bound the minimum distance of a code, we also discuss the Singleton bound analogues in the Lee metric as well as the sphere-packing bound and the Gilbert-Varshamov bound.

3.1 Codes over Integer Residue Rings

Different to classical coding theory over finite fields (see Section 2.3), the Lee metric is defined over integer residue rings $\mathbb{Z}/q\mathbb{Z}$, for a positive integer q. Therefore, in this section we introduce block codes whose underlying alphabet is an integer residue ring $\mathbb{Z}/q\mathbb{Z}$ and, especially, $\mathbb{Z}/p^s\mathbb{Z}$ for a positive integer s and a prime number p. Equivalently to the classical case over a finite field, codes over rings can be characterized by a *generator matrix* and a *parity-check matrix*. We will formally define ring-linear codes in this section and study their structure and parameters. Codes over rings have been introduced in 1963 by Assmus and Harold [9]. We refer to [67, 118] for more background on codes over rings.

In the classical coding theory setting over finite fields, a code is a subspace of a given finite field. The analogue of subspaces when working over rings are modules, or submodules. The following definition of a ring-linear code works for any finite ring $\mathcal{R}$. However, in the course of this thesis we will consider only the integer residue rings $\mathbb{Z}/q\mathbb{Z}$ and $\mathbb{Z}/p^s\mathbb{Z}$.

Definition 3.1.1 (Ring-Linear Code)**.** Let $\mathcal{R}$ be a finite ring. A *ring-linear code of length* n is an $\mathcal{R}$-submodule $\mathcal{C}$ of $\mathcal{R}^n$.

Assume that the underlying finite ring is of size q, then we define the rate of the code $\mathcal{C}$, similar to the Hamming case, by

$$R := \frac{\log_q(|\,\mathcal{C}\,|)}{n}.$$

Similar to the classical case over finite fields, a ring-linear code $\mathcal{C}$ can be represented by a generator matrix and a parity-check matrix which define $\mathcal{C}$ in terms of their image and kernel, respectively.

Codes over rings have mainly been studied over $\mathbb{Z}/q\mathbb{Z}$, where q is a positive integer, or in the more specific case where $q = p^s$ is a power s of a prime p [28, 29, 71, 115, 123]. Due to its algebraic structure, cyclic codes over finite chain rings were considered [35, 98]. Choosing $q = p^s$ yields more structure on the finite integer ring and hence, yields more structure for codes defined over $\mathbb{Z}/p^s\mathbb{Z}$. Hence, let us now focus on the integer residue ring $\mathbb{Z}/p^s\mathbb{Z}$ for a prime number p and a positive integer s. Compared to any integer residue ring $\mathbb{Z}/q\mathbb{Z}$, the ring $\mathbb{Z}/p^s\mathbb{Z}$ is a chain ring, meaning that its ideals form a chain of inclusions. Let us denote the minimal ideal containing the element p^i by

$$\langle p^i \rangle := p^i(\mathbb{Z}/p^s\mathbb{Z}).$$

We then observe the following chain of inclusions

$$\langle p^{s-1} \rangle \subseteq \langle p^{s-2} \rangle \subseteq \ldots \subseteq \langle p \rangle \subseteq \mathbb{Z}/p^s\mathbb{Z}. \tag{3.1}$$

This additional structure allows us to determine more parameters of a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ and to define a systematic form of generator matrices and parity-check matrices, respectively. By the fundamental theorem of finite Abelian groups, a $\mathbb{Z}/p^s\mathbb{Z}$-submodule $\mathcal{C}$ can be uniquely decomposed into a finite direct sum of $\mathbb{Z}/p^s\mathbb{Z}$-submodules, i.e., there exist s nonnegative integers $k_0, \ldots, k_{s-1}$ such that

$$\mathcal{C} \simeq \bigoplus_{i=0}^{s-1} (\mathbb{Z}/p^{s-i}\mathbb{Z})^{k_i}. \tag{3.2}$$

This implies that the cardinality $|\,\mathcal{C}\,|$ of the code is given by

$$|\,\mathcal{C}\,| = \prod_{i=0}^{s-1} (p^{s-i})^{k_i} = p^{\sum_{i=0}^{s}(s-i)k_i}. \tag{3.3}$$

In literature the cardinality of a ring-linear code is sometimes referred to the *type* of the code. In order not to confuse it with the type introduced in Section 2.2 we will omit this name. However, we call $(k_0, \ldots, k_{s-1})$ the *subtype* of the code.

Recall from classical coding theory, that the dimension of a code over a finite field determined, together with the size of the field, the cardinality of the code. In this sense and owing to (3.3), we define the $\mathbb{Z}/p^s\mathbb{Z}$-analogue of the dimension of a code over a finite field as

$$k := \log_{p^s}|\,\mathcal{C}\,| = \sum_{i=0}^{s-1} \frac{s-i}{s} k_i.$$

We call this value k the $\mathbb{Z}/p^s\mathbb{Z}$-dimension of the code in order not to confuse with the notion of a dimension. In contrast to the dimension over a finite field, the $\mathbb{Z}/p^s\mathbb{Z}$-dimension of a code may not always be an integer. Furthermore, we define the rank of $\mathcal{C}$ by

$$K := \sum_{i=0}^{s-1} k_i.$$

Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K and $\mathbb{Z}/p^s\mathbb{Z}$-dimension k. Note that $K = k$ if and only if $(k_0, k_1, \ldots, k_{s-1}) = (k_0, 0, \ldots, 0)$, i.e., $K = k = k_0$. Codes with this property define a specific class of codes over finite chain rings.

Definition 3.1.2. Given a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K and subtype $(k_0, \ldots, k_{s-1})$. We call k_0 the *free rank* of $\mathcal{C}$. Moreover, we say that $\mathcal{C}$ is *free* if its rank coincides with its free rank, i.e., if $K = k_0$.

With the decomposition given in (3.2), we observe that if $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ is a free code it admits a $\mathbb{Z}/p^s\mathbb{Z}$-basis and is isomorphic to

$$\mathcal{C} \simeq (\mathbb{Z}/p^s\mathbb{Z})^{k_0}.$$

As mentioned, the notion of generator matrices and parity-check matrices can be adapted to codes over finite rings as well. In the case of codes over a finite chain ring $\mathbb{Z}/p^s\mathbb{Z}$, the rank K and the subtype $(k_0, \ldots, k_{s-1})$ allow us to define a systematic form of both types of matrices.

Definition 3.1.3. Consider a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K and free rank k_0. A matrix $G \in (\mathbb{Z}/p^s\mathbb{Z})^{K\times n}$ is called a *generator matrix* of $\mathcal{C}$ if the rows of G span $\mathcal{C}$. A *parity-check matrix* H is an $(n - k_0) \times n$ matrix over $\mathbb{Z}/p^s\mathbb{Z}$ whose kernel coincides with $\mathcal{C}$.

Proposition 3.1.4. *Let $\mathcal{C}$ be a linear code in $\mathbb{Z}/p^s\mathbb{Z}$ of subtype $(k_0, \ldots, k_s)$ and rank K. Then $\mathcal{C}$ is permutation equivalent to a code having a generator matrix $G_{\mathsf{sys}} \in (\mathbb{Z}/p^s\mathbb{Z})^{K\times n}$ of the form*

$$G_{\mathsf{sys}} = \begin{pmatrix} \mathbb{I}_{k_0} & A_{1,2} & A_{1,3} & \cdots & A_{1,s} & A_{1,s+1} \\ 0 & p\mathbb{I}_{k_1} & pA_{2,3} & \cdots & pA_{2,s} & pA_{2,s+1} \\ 0 & 0 & p^2\mathbb{I}_{k_2} & \cdots & p^2A_{3,s} & p^2A_{3,s+1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & p^{s-1}\mathbb{I}_{k_{s-1}} & p^{s-1}A_{s,s+1} \end{pmatrix}, \tag{3.4}$$

where $A_{i,s+1} \in (\mathbb{Z}/p^{s+1-i}\mathbb{Z})^{k_{i-1}\times(n-K)}$, $A_{i,j} \in (\mathbb{Z}/p^{s+1-i}\mathbb{Z})^{k_{i-1}\times k_j}$ for $j \leq s$. In addition, the code $\mathcal{C}$ is permutation equivalent to a code having a parity-check matrix $H \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k_0)\times n}$ of the form

$$H_{\mathsf{sys}} = \begin{pmatrix} B_{1,1} & B_{1,2} & \cdots & B_{1,s-1} & B_{1,s} & \mathbb{I}_{n-K} \\ pB_{2,1} & pB_{2,2} & \cdots & pB_{2,s-1} & p\mathbb{I}_{k_{s-1}} & 0 \\ p^2B_{3,1} & p^2B_{3,2} & \cdots & p^2\mathbb{I}_{k_{s-2}} & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ p^{s-1}B_{s,1} & p^{s-1}\mathbb{I}_{k_1} & \cdots & 0 & 0 & 0 \end{pmatrix}, \tag{3.5}$$

where $B_{1,j} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K)\times k_{j+1}}$, $B_{i,j} \in (\mathbb{Z}/p^{s+1-i}\mathbb{Z})^{k_{s-i+1}\times k_{j+1}}$ for $i > 1$.

We call the forms in (3.4) and (3.5) the *systematic form* of a generator matrix and a parity-check matrix, respectively. Notice that the systematic form of a generator matrix of a code over a finite chain ring is quite different to the one over a finite chain ring. However, if $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ is a free code the systematic form of a generator matrix coincides with (2.6), i.e., there is a matrix $A \in (\mathbb{Z}/p^s\mathbb{Z})^{k_0\times(n-k_0)}$ such that

$$G_{\mathsf{sys}} = \begin{pmatrix} \mathbb{I}_{k_0} & A \end{pmatrix}.$$

The subtype $(k_0, \ldots, k_s)$ of a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ hence indicates the number of rows of a generator matrix G lying in the ideal $\langle p^i \rangle$. That is, there are k_i rows of G lying in $\langle p^i \rangle$ but not in $\langle p^j \rangle$ for any $j > i$. Additionally, to the subtype of a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$, we can define a similar parameter going over the columns of a generator matrix of $\mathcal{C}$.

Definition 3.1.5. Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K. For each $j \in \{1, \ldots, n\}$ consider the j-th coordinate map

$$\begin{array}{rccc} \pi_j : & (\mathbb{Z}/p^s\mathbb{Z})^n & \longrightarrow & \mathbb{Z}/p^s\mathbb{Z} \\ & (c_1, \ldots, c_n) & \longmapsto & c_j \end{array}.$$

The *support subtype* of $\mathcal{C}$ is defined to be an $(s+1)$-tuple $(n_0(\mathcal{C}), \dots, n_s(\mathcal{C}))$, where $n_i(\mathcal{C})$ counts the number coordinates $j \in \{1, \dots, n\}$ belonging to ideal $\langle p^i \rangle$, i.e.,

$$n_i(\mathcal{C}) := \left| \left\{ j \in \{1, \dots, n\} \mid \langle \pi_j(\mathcal{C}) \rangle = \langle p^i \rangle \right\} \right|.$$

A code with $n_s(\mathcal{C}) = 0$ is called *non-degenerate*.

If the code $\mathcal{C}$ is clear form the context, we will write n_i instead of $n_i(\mathcal{C})$.

Example 3.1.6. Let $\mathcal{C} \subset (\mathbb{Z}/9\mathbb{Z})^4$ be defined by the generator matrix

$$G_{\mathsf{sys}} = \begin{pmatrix} 1 & 0 & 4 & 3 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 3 & 6 \end{pmatrix}.$$

The code $\mathcal{C}$ has subtype $(k_0, k_1) = (2, 1)$. In fact, the first two rows of G_{sys} lie in the ideal $\langle p^0 \rangle = \langle 1 \rangle$ but not in the ideal $\langle p \rangle$ whereas the elements of the last row are all contained in $\langle p \rangle$. The rank K and the $\mathbb{Z}/p^s\mathbb{Z}$-dimension k are computed as

$$K = \sum_{i=0}^{2} k_i = 3 \quad \text{and} \quad k = \sum_{i=0}^{2} \frac{2-i}{2} k_i = k_0 + \frac{1}{2} k_1 = 2.5.$$

To compute the support subtype we calculate for each column the minimal ideal $\langle p^i \rangle$ that contains all nonzero the entries of the respective column. Clearly, for the first two columns this is the ideal generated by $1 = p^0$. The same holds for the third column. In the last column all nonzero entries are contained in the ideal $\langle 1 \rangle$ and $\langle 3 \rangle$ but not in $\langle 9 \rangle$. Hence, we assign this column to the ideal $\langle 3 \rangle$. As we have three columns assigned to $\langle p^0 \rangle$ and one to $\langle p \rangle$, the support subtype of $\mathcal{C}$ is

$$(n_0, n_1, n_2) = (3, 1, 0).$$

Over finite fields we classically endow the ambient space with the Hamming metric, as discussed in Section 2.3.1. Over a finite integer ring $\mathbb{Z}/q\mathbb{Z}$ we can define the Hamming weight of an n-tuple $x \in (\mathbb{Z}/q\mathbb{Z})^n$ in the exact same way as in Definition 2.3.6. Note, that Proposition 2.3.8 is independent of the ambient space and thus the same error-correction capability holds for finite integer rings in the Hamming metric. Understanding the error-correction performance is hence dependening on the minimum Hamming distance of the code. Over finite chain rings, we can deduce the same Singleton bound on the minimum Hamming distance as in the field case.

Proposition 3.1.7. *Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a code of $\mathbb{Z}/p^s\mathbb{Z}$-dimension k. Then its minimum Hamming distance can be upper bounded by*

$$\mathrm{d}_{\mathsf{H}}(\mathcal{C}) \leq n - k + 1.$$

This result can be further tightened using the rank K instead of the $\mathbb{Z}/p^s\mathbb{Z}$-dimension k.

Proposition 3.1.8 ([50, 52])**.** *Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K. Then*

$$\mathrm{d}_{\mathsf{H}}(\mathcal{C}) \leq n - K + 1.$$

Similarly to classical coding theory, we call a linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ a MDS code *with respect to the $\mathbb{Z}/p^s\mathbb{Z}$-dimension (or the rank)* if $\mathcal{C}$ meets the bound in Proposition 3.1.7 (or Proposition 3.1.8, respectively) with equality. In Section 3.4 we introduce Singleton-like bounds for the Lee metric based on a similar puncturing argument as in the Hamming metric (see the proof of Theorem 2.3.9). In contrast to the Hamming metric Singleton bound, we will see that its Lee-metric analogous is far from being tight.

3.2 Basic Definitions and Results

In the following we consider a positive integer q and the integer residue ring $\mathbb{Z}/q\mathbb{Z}$ of q elements. The integer residue ring $\mathbb{Z}/q\mathbb{Z}$ can be interpreted and represented in various ways. We mainly use the representation of the set of the first q integers $\{0, 1, \ldots, q-1\}$.

Definition 3.2.1. We define the *Lee weight* of an element $a \in \mathbb{Z}/q\mathbb{Z}$ interpreted as an integer in $\{0, \ldots, q-1\}$ in the following way:

$$\mathrm{wt}_{\mathsf{L}}(a) := \min\{a, q-a\}.$$

Similarly, the Lee weight of a vector $x \in (\mathbb{Z}/q\mathbb{Z})^n$ of length n is the sum the Lee weights of each entry of x, i.e.

$$\mathrm{wt}_{\mathsf{L}}(x) = \sum_{i=1}^{n} \mathrm{wt}_{\mathsf{L}}(x_i).$$

Let us give an intuitive picture to establish the Lee weight of the elements of $\mathbb{Z}/q\mathbb{Z}$. Consider the elements of $\mathbb{Z}/q\mathbb{Z}$ on a circle with equal distances between them. Then the Lee weight of $a \in \mathbb{Z}/q\mathbb{Z}$ is the minimal number of arcs separating a from zero. Figure 3.1 illustrates this over the ring $\mathbb{Z}/9\mathbb{Z}$.

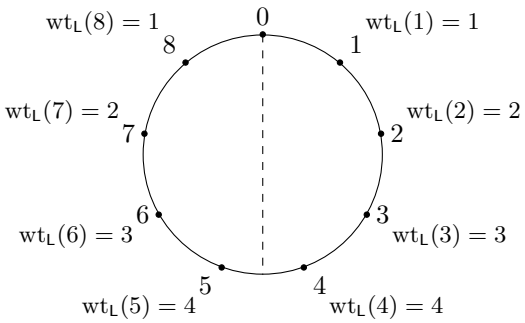

FIGURE 3.1: Circular representation of $\mathbb{Z}/9\mathbb{Z}$ indicating the respective Lee weight of its elements.

This yields the following symmetry property of the Lee weight,

$$\mathrm{wt}_{\mathsf{L}}(a) = \mathrm{wt}_{\mathsf{L}}(q-a). \tag{3.6}$$

Equation (3.6) implies that the Lee weight of any element in $\mathbb{Z}/q\mathbb{Z}$ can never exceed $\lfloor q/2 \rfloor$. Furthermore, we observe that the Lee weight of $a \in \mathbb{Z}/q\mathbb{Z}$ is always lower bounded by its Hamming weight (see Definition 2.3.6). Equality between the two weights holds if and only $q \in \{2, 3\}$. Hence, for a vector $x \in (\mathbb{Z}/q\mathbb{Z})^n$ we have

$$\mathrm{wt}_{\mathsf{H}}(x) \leq \mathrm{wt}_{\mathsf{L}}(x) \leq \mathrm{wt}_{\mathsf{H}}(x)\,\lfloor q/2 \rfloor \leq n\,\lfloor q/2 \rfloor.$$

Similarly to the Hamming distance (Definition 2.3.6), we can define the Lee distance of two vectors as follows.

Definition 3.2.2. Let $x, y \in (\mathbb{Z}/q\mathbb{Z})^n$. The *Lee distance* between x and y, $\mathrm{d}_{\mathsf{L}}(x, y)$ is the Lee weight of their difference, i.e.,

$$\mathrm{d}_{\mathsf{L}}(x, y) := \mathrm{wt}_{\mathsf{L}}(x - y).$$

Analogously, the minimum Lee distance $\mathrm{d}_{\mathsf{L}}(\mathcal{C})$ of a code $\mathcal{C}$ is lower bounded by the minimum Hamming distance $\mathrm{d}_{\mathsf{H}}(\mathcal{C})$ of the same code and upper bounded by the $\lfloor q/2 \rfloor$-fold of the minimum Hamming distance, i.e., for any $x, y \in \mathbb{Z}/q\mathbb{Z}$ it holds

$$\mathrm{d}_{\mathsf{H}}(x, y) \leq \mathrm{d}_{\mathsf{L}}(x, y) \leq \lfloor q/2 \rfloor\, \mathrm{d}_{\mathsf{H}}(x, y).$$

Let us assume that we pick an element a uniformly at random from $\mathbb{Z}/q\mathbb{Z}$. Lemma 3.2.3 then states the expected Lee weight of that random element a.

Lemma 3.2.3 ([112, Problem 10.15])**.** *Let A be a uniformly distributed random variable over $\mathbb{Z}/q\mathbb{Z}$. The expected Lee weight of A is*

$$\delta_q := \mathbb{E}(\mathrm{wt_L}(A)) = \begin{cases} \frac{q^2-1}{4q} & \text{if } q \text{ is odd} \\ \frac{q}{4} & \text{if } q \text{ is even} \end{cases}.$$

Proof. As $A \in \mathbb{Z}/q\mathbb{Z}$ is chosen uniformly at random, we have $\mathbb{P}(A = i) = \frac{1}{q}$ for every $i \in \mathbb{Z}/q\mathbb{Z}$. The proof then follows by computing

$$\mathbb{E}(\mathrm{wt_L}(A)) = \sum_{i=0}^{q-1} \mathrm{wt_L}(i)\mathbb{P}(A = i) = \frac{1}{q}\sum_{i=0}^{q-1} \mathrm{wt_L}(i).$$

□

3.3 Spheres and Balls

A natural question arising is: given a vector $x \in (\mathbb{Z}/q\mathbb{Z})^n$ of fixed Lee weight $\mathrm{wt_L}(x) = t$, what is the distribution of the entries of x? In Section 5.1.2 we answer this question in the asymptotic regime in terms of large block length. The expected Lee weight δ_q will serve for the derivation of the distribution. Knowing the size of an n-dimensional sphere or ball of a given Lee-radius is crucial to determine bounds like, for instance, the Gilbert-Varshamov bound [10].

Generally, there is an exact formula on how to compute the size of an ℓ-dimensional sphere of fixed radius t. Namely, for the radius t we compute all its integer partitions where each part of the partition has a size of at most the maximal possible weight in the corresponding metric. These will represent the decomposition of the nonzero entries of the elements in the sphere. To get the size of the sphere we sum over all integer partitions adding up the number of elements that have a weight decomposition corresponding to the integer partition. Although this procedure provides the exact value of the sphere size, it often does not give an intuitive or practical understanding of the sphere size or how this size changes as the parameters change. For large parameters it is even impractical to compute the size in this way. Hence, it can be useful to have practical upper bounds and lower bounds on this formula. A current method of obtaining both upper and lower bounds is, for instance, to only consider the partition attaining the maximum number of elements. Another method is to bound the size of an ℓ-dimensional ball of radius t, since clearly every upper bound on its size is a valid upper bound on the size of the sphere too. On a more complex analytic side, sizes of spheres and balls can be described using generating functions, and their limit for n going to infinity can be computed using the saddle point technique used in [60]. In 1994, Löliger described a general method to derive bounds on the size of a discrete ball given any additive metric [86]. A similar approach specifically for the Lee metric has then been done in [27]. All these quantities can be described using generating functions, and their limit for n going to infinity can be computed using the saddle point technique used in [60]. In the complexity analysis of certain decoding algorithms (for instance, information set decoding), we are interested in the asymptotic size of Lee spheres, Lee balls, and some types of restricted Lee spheres.

Generally, we consider two functions $f(x)$ and $g(x)$ both not depending on n and define a generating function

$$\Phi(x) = f(x)^n g(x).$$

With an eye on Lemma 3.3.2, we are usually interested in the coefficients of generating functions. The goal now is to estimate the coefficient $\mathrm{coeff}\left[\Phi(z)^n, z^k\right]$ of the term z^k in the function $\Phi(x)^n$ for some fixed $k \in \mathbb{N}$. The following gives an asymptotic result on the growth rate of this coefficient.

Lemma 3.3.1 ([60, Corollary 1]). *Let $\Phi(x) = f(x)^n g(x)$ with $f(0) \neq 0$, and $t(n)$ be a function in n. Set $T := \lim_{n\to\infty} t(n)/n$ and set ρ to be the solution to*

$$\Delta(x) := \frac{xf'(x)}{f(x)} = T.$$

If $\Delta'(\rho) > 0$, and the modulus of any singularity of $g(x)$ is larger than ρ, then for large n

$$\frac{1}{n}\log_{p^s}\left(\operatorname{coeff}\left[\Phi(x), x^{t(n)}\right]\right) \approx \log_{p^s}(f(\rho)) - T\log_{p^s}(\rho) + o(1).$$

In alliance with the question mentioned, we are interested in the n-dimensional sphere (respectively, the n-dimensional ball) of Lee-radius t which we define as follows, respectively.

$$\mathcal{S}_{t,q}^{(n)} := \{x \in (\mathbb{Z}/q\mathbb{Z})^n \mid \mathrm{wt}_{\mathsf{L}}(x) = t\},$$
$$\mathcal{B}_{t,q}^{(n)} := \{x \in (\mathbb{Z}/q\mathbb{Z})^n \mid \mathrm{wt}_{\mathsf{L}}(x) \leq t\}.$$

Spheres and balls have a natural relation to each other. In fact, we can describe any ball of a given radius t by the sum of all spheres of radius up to t. This holds as well in the Lee metric, i.e.,

$$\mathcal{B}_{t,q}^{(n)} = \sum_{r=0}^{t} \mathcal{S}_{r,q}^{(n)}. \tag{3.7}$$

According to the definition of generating functions, Definition 2.0.1, we define $s_{j,q}^{(n)} = \left|\mathcal{S}_{j,q}^{(n)}\right|$ and let $S_q^{(n)}(z) := \sum_j s_{j,q}^{(n)} z^j$ denote the generating function of the size of n-dimensional spheres in the Lee metric. By Taylor's theorem on generating functions (see Theorem 2.0.2) we deduce for every nonnegative integer $j \in \mathbb{N}$

$$s_{j,q}^{(n)} = \frac{1}{j!}\frac{\mathrm{d}^j}{\mathrm{d}\, z^j}\left(S_q^{(n)}(z)\right)_{z=0}. \tag{3.8}$$

Rewriting Equation (3.7) for the volume of a sphere and using (3.8) yields

$$\left|\mathcal{B}_{t,q}^{(n)}\right| = \sum_{i=0}^{r} \frac{1}{i!}\frac{\mathrm{d}^i}{\mathrm{d}\, z^i}\left(S_q^{(n)}(z)\right)_{z=0}. \tag{3.9}$$

Next, we would like to find a closed form expression for $S_q^{(n)}$. Since the Lee distance of a vector is additive, we have the same computation for every vector with a given Lee weight. Thus, the generating function $S_q^{(n)}(z)$ is multiplicative over the coordinates, which means that $S_q^{(n)}(z) = \left(S_q^{(1)}(z)\right)^n$. It suffices to find a closed form for the generating function $S_q^{(1)}$ of the sequence of integers $s_{i,q}^{(1)}$ defined to be the number of elements in $\mathbb{F}_q$ having Lee weight exactly i. By the symmetric property of the Lee weight shown in (3.6), we know that $s_{0,q}^{(1)} = 1$ and $s_{i,q}^{(1)} = 2$ for every $j \in \{1, \ldots, \lfloor q/2\rfloor - 1\}$. For the number of elements in $\mathbb{Z}/q\mathbb{Z}$ with Lee weight $\lfloor q/2\rfloor$ we have two options, depending on whether q is even or odd. If q is odd there is an even number of nonzero elements and hence $s_{\lfloor q/2\rfloor,q}^{(1)} = 2$. Analogously, if q is even, the number of nonzero elements in $\mathbb{Z}/q\mathbb{Z}$ is odd and thus $s_{\lfloor q/2\rfloor,q}^{(1)} = 1$. Therefore, we deduce the following closed form.

$$S_q^{(1)}(z) = \sum_{i=0}^{\lfloor q/2\rfloor} s_{i,q}^{(1)} z^i = \begin{cases} 1 + 2z + \cdots + 2z^{(q-1)/2} & q \text{ odd}, \\ 1 + 2z + \cdots + 2z^{(q-2)/2} + z^{q/2} & q \text{ even}. \end{cases} \tag{3.10}$$

Hence, the corresponding closed form for $S_q^{(n)}$ is obtained by raising the equations in (3.10) to the n-th power. Hence, using Lemma 3.3.1 and the relation (3.9) allows us to derive the following result.

Lemma 3.3.2 (Surface of the spheres)**.** *The cardinalities of the n-dimensional Lee-sphere $\mathcal{S}_{d,q}^{(n)}$ and Lee-ball $\mathcal{B}_{d,q}^{(n)}$, respectively, of radius d over $\mathbb{Z}/q\mathbb{Z}$ are given by*

$$\left|\mathcal{S}_{d,q}^{(n)}\right| = \operatorname{coeff}\left[\left(S_q^{(1)}(z)\right)^n, z^d\right] \quad \text{and} \quad \left|\mathcal{B}_{d,q}^{(n)}\right| = \operatorname{coeff}\left[\frac{\left(S_q^{(1)}(z)\right)^n}{1-x}, z^d\right],$$

where $S_q^{(1)}(z)$ is given as in Equation (3.10).

There is an exact double-binomial formula for the n-dimensional Lee-sphere $\mathbb{Z}/q\mathbb{Z}$ if the radius t does not exceed $q/2$.

Proposition 3.3.3. [112, Proposition 10.10] *The size of an n-dimensional Lee-sphere over $\mathbb{Z}/q\mathbb{Z}$ with radius $t \leq q/2$ is given by*

$$\left|\mathcal{S}_{t,q}^{(n)}\right| = \sum_{i=0}^{n} 2^i \binom{n}{i}\binom{t}{i}.$$

Furthermore, we introduce the normalized logarithmic surface (respectively volume) spectra by

$$\sigma_{\delta n}^{(n)} := \frac{1}{n}\log_2\left|\mathcal{S}_{\delta n,q}^{(n)}\right| \quad \text{and} \quad \nu_{\delta n}^{(n)} := \frac{1}{n}\log_2\left|\mathcal{B}_{\delta n,q}^{(n)}\right|$$

while their asymptotic counterparts are denoted by

$$\sigma_\delta := \lim_{n\to\infty}\frac{1}{n}\log_2\left|\mathcal{S}_{\delta n,q}^{(n)}\right| \quad \text{and} \quad \nu_\delta := \lim_{n\to\infty}\frac{1}{n}\log_2\left|\mathcal{B}_{\delta n,q}^{(n)}\right|.$$

3.4 Bounds on Lee-Metric Codes

Several bounds, such as the Gilbert-Varshamov, Plotkin, Elias and Singleton bounds, for codes in the Lee metric have been established using various techniques [5, 24, 42, 119, 137]. In [10] the author derived asymptotic versions of the above-mentioned types of bounds.

One of the most famous bounds in classical coding theory is the Singleton bound (Theorem 2.3.9). Even though the Lee metric is a rather old metric, a Singleton-like bound has only been found in 2000 by Shiromoto [119].

Theorem 3.4.1 (Singleton-Like Bound in the Lee Metric, [119])**.** *Consider a linear code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ of $\mathbb{Z}/p^s\mathbb{Z}$-dimension k. Then the following bound holds*

$$\left\lfloor \frac{\mathrm{d_L}(\mathcal{C}) - 1}{\lfloor q/2 \rfloor} \right\rfloor \leq n - k.$$

This bound follows the same puncturing argument as for the classical case in the Hamming metric (see proof of Theorem 2.3.9). Since the Lee weight of an element can exceed its Hamming weight, and thus exceed the value 1, having a given Lee weight t does not imply that there are t nonzero positions. Hence, to follow the proof provided by Singleton for the Hamming metric, we would have to normalize by the maximal possible Lee weight in an integer residue ring $\mathbb{Z}/q\mathbb{Z}$ given by $\lfloor q/2 \rfloor$. Even though the existence of a nontrivial code attaining this bound has been shown by the example $\mathcal{C} = \langle (1,2) \rangle \subset (\mathbb{Z}/5\mathbb{Z})^2$, the authors in [34] showed that this code is actually the only nontrivial linear code attaining Shiromoto's Singleton-like bound in the Lee metric. Thus, studying further techniques to derive tighter bounds in the Lee metric is crucial.

Note also that Shiromot's bound implies that $\mathrm{d_L}(\mathcal{C}) \leq \lfloor q/2 \rfloor (n-k) + a$, where $a \in \{1, \ldots, \lfloor q/2 \rfloor\}$. Alderson and Huntemann managed to tighten this bound being able to amit the integer a under the assumption that the $\mathbb{Z}/p^s\mathbb{Z}$-dimension k of the code is a positive integer.

Theorem 3.4.2 ([5]). *For any code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ of $\mathbb{Z}/p^s\mathbb{Z}$-dimension k, with $1 < k < n$ is a positive integer, we have that*

$$\mathrm{d_L}(\mathcal{C}) \leq \lfloor q/2 \rfloor (n-k).$$

This bound is tighter than Shiromoto's bound. However, codes attaining this bound are still sparse as q or n tend to infinity [32]. In Chapter 4 we give an alternative bound based on a puncturing argument, and we study new bounds on the minimum Lee distance using different techniques instead.

Another interesting question in coding theory is the following: given a code of length n and minimum distance d, what is its maximal cardinality? This question has been captured in several bounds such as the sphere-packing bound (sometimes referred to as Hamming bound), the Plotkin bound, the Gilbert-Varshamov bound and the Elias bound. In the Hamming metric these bounds together with their asymptotic versions are well-known (see Section 2.3.2 for a recap on the Singleton-bound, the sphere-packing bound and the Gilbert-Varshamov bound). In the Lee metric similar bounds exist and most involve the size of the n-dimensional Lee-ball of a given radius. Their asymptotic counterparts can be computed by using generating functions (see Definition 2.0.1) or the saddle-point technique [60] as a part of complex analysis.

The sphere-packing bound in the Lee metric is stated as follows.

Theorem 3.4.3 (Sphere-packing, [24]). *Let $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ be a linear code with minimum Lee distance $\mathrm{d_L}(\mathcal{C}) = d$. For any $t = \lfloor \frac{\mathrm{d_L}(\mathcal{C})-1}{2} \rfloor$ it holds that*

$$|\,\mathcal{C}\,| \leq \frac{q^n}{\left|\,\mathcal{B}_{t,q}^{(n)}\,\right|}.$$

For the asymptotic form of this bound, we use the information rate R of the best code instead of the minimum Lee distance or the maximum size of a code.

Theorem 3.4.4 (Asymptotic sphere-packing, [86]). *Given a code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ of minimum Lee distance $\mathrm{d_L}(\mathcal{C}) = d$ and maximal information rate R. For any $t = \lfloor \frac{\mathrm{d_L}(\mathcal{C})-1}{2} \rfloor$ it holds that*

$$\lim_{n\longrightarrow\infty} \sup R \leq \lim_{n\longrightarrow\infty} \left(1 - \frac{1}{n}\log_q\left(\left|\,\mathcal{B}_{t,q}^{(n)}\,\right|\right)\right)$$

A lower bound on the asymptotic rate is captured in the Gilbert-Varshamov bound. Let us first state the bound in its finite length setting. There exist two versions of the Gilbert-Varshamov bound: The classical bound is a lower bound on the maximal cardinality that any (not necessarily linear) code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ can achieve. The second version of the bound is an existence bound, stating sufficient conditions for the existence of a linear code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$. In the course of this thesis we are mainly interested in the classical statement of the bound.

Theorem 3.4.5 (Gilbert-Varshamov in the Lee metric, [24]). *Let $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ be a linear code of minimum Lee distance $\mathrm{d_L}(\mathcal{C}) = d$. Then,*

$$|\,\mathcal{C}\,| \geq \frac{q^n}{\left|\,\mathcal{B}_{d-1,q}^{(n)}\,\right|}$$

Theorem 3.4.6 (Asymptotic Gilbert-Varshamov, [86]). *Given a code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ of minimum Lee distance $\mathrm{d_L}(\mathcal{C}) = d$ and maximal information rate R. Then,*

$$\lim_{n\longrightarrow\infty} \inf R \geq \lim_{n\longrightarrow\infty} \left(1 - \frac{1}{n}\log_q\left(\left|\,\mathcal{B}_{(d-1)/2,q}^{(n)}\,\right|\right)\right).$$

The Gilbert-Varshamov bound is additionally of interest in applications like information set decoding and the underlying syndrome decoding problem (see Chapter 7 for more details). In this application, the Gilbert-Varshamov bound provides a threshold for the weight of the error vector. If the weight of the error vector is below the threshold a unique solution to

the syndrome decoding problem exists. Otherwise, there are many possible solutions to the problem and finding all the solution might result in a more expensive computation.

Chapter 4

Bounds on the Minimum Lee Distance

The minimum distance of a code is in one-to-one correspondence with its error-correction capability. More precisely, the higher the minimum distance, the better the error-correction capability (see Proposition 2.3.8). The study of optimal codes in terms of the error-correction performance is an important task in classical coding theory. The most famous bound is the Singleton bound which provides a trade-off between the minimum distance of a code and its dimension. The Singleton bound has been introduced for the Hamming metric by Singleton [121] in 1964 and has already been studied by Komamiya [80]. Ever since, codes attaining the Singleton bound have been studied extensively. Letting the field size q grow large, it is well-known that codes attaining the Singleton bound over $\mathbb{F}_q$ are dense, meaning that if we pick a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ uniformly at random among all codes of the same parameters then $\mathcal{C}$ attains the Singleton bound almost surely in the limit of q.

Similar results have been found for the rank metric, introduced in 1978 by Delsarte [48] and reintroduced by Gabidulin [56] and Roth [111]. Gabidulin in [56] derived a Singleton-like bound for rank metric codes already in 1985, and it has been shown in [95] that linear codes over $\mathbb{F}_{q^m}$ attaining Gabidulin's Singleton bound are dense in the limit of q and m. Considering, however, $\mathbb{F}_q$-linear codes endowed with the rank metric implies sparsity for q tending to infinity [68] except for some special cases where m or n are 2 [6, 33, 62].

If we change the ambient space and consider a finite chain ring $\mathbb{Z}/p^s\mathbb{Z}$ and endow it with the Lee metric, the situation is different. In Section 3.4, we have seen that Singleton-like bounds for Lee-metric codes over a chain ring $\mathbb{Z}/p^s\mathbb{Z}$ have been derived by Shiromoto [119] and Alderson and Huntemann [5] (see Theorem 3.4.1 and 3.4.2, respectively). However, codes attaining these bounds are extremely sparse as p, s and n tend to infinity [32]. Hence, using a puncturing argument for Lee-metric codes is not suitable. This is inter alia due to the fact that for a puncturing argument in the Lee metric, we have to normalize by the maximum Lee weight $\lfloor p^s/2 \rfloor$.

In this chapter we tackle the problem of finding tighter bounds for the minimum Lee distance of a linear code over $\mathbb{Z}/p^s\mathbb{Z}$ using generalized weights. The idea of using generalized weights stems from the Hamming metric case [106]. We therefore start by introducing this concept over the Hamming metric first, and we then adapt it to the Lee metric in Section 4.1. We discuss their advantages and disadvantages in Sections 4.2 and 4.3 and derive novel bounds on the minimum Lee distance with respect to the novel definitions of a Lee-support. In a second step, we give a novel definition of generalized Lee distances making use of the algebraic structure of the chain ring. The results and bounds on the minimum Lee distance of a code presented in this chapter have been studied in [19] in collaboration with Violetta Weger.

A New Puncturing Bound

Before introducing supports and generalized weights in the Lee metric, we give an improved version of Shiromoto's Singleton bound still using a puncturing argument. Recall from Section 3.4 that Shiromoto's Singleton bound for the Lee metric is far from being a tight bound. One reason is that the Lee weight of an element is upper bounded by the maximal weight $\lfloor q/2 \rfloor$ in a given integer residue ring $\mathbb{Z}/q\mathbb{Z}$. However, if $q = p^s$ is a prime power a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$

can be decomposed into a finite sum of $\mathbb{Z}/p^s\mathbb{Z}$-submodules (as shown in (3.2)). In each of the submodules $\mathbb{Z}/p^i\mathbb{Z}$, the elements can achieve a different maximum Lee weight which, for simplicity, we define as

$$M_i := \max\left\{\mathrm{wt}_{\mathsf{L}}(a) \mid a \in \langle p^i \rangle\right\} = \left\lfloor \frac{p^{s-i}}{2} \right\rfloor p^i. \tag{4.1}$$

Hence, knowing the number of coordinates of $\mathcal{C}$ belonging to a submodule $\mathbb{Z}/p^i\mathbb{Z}$ can be fruitful for the understanding of the maximum possible Lee weight in these coordinates. Recall from Definition 3.1.5 that the support subtype $(n_0, \ldots, n_s)$ of a code $\mathcal{C}$ captures this number for each submodule $\mathbb{Z}/p^i\mathbb{Z}$. Using the support subtype, we can easily derive Lee-metric Singleton-like bound from the puncturing argument.

Theorem 4.0.1. *Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K and support subtype given by $(n_0, \ldots, n_{s-1}, 0)$. Define for all $k \in \{0, \ldots, s\}$*

$$B_k = \sum_{i=k}^{s-1} n_i \quad \textit{and} \quad A_k = \sum_{i=k}^{s-1} n_i M_i.$$

Let $j \in \{1, \ldots, s-1\}$ be the smallest positive integer such that $A_j < \mathrm{d}_{\mathsf{L}}(\mathcal{C})$, then

$$K \leq n - B_j - \left\lfloor \frac{\mathrm{d}_{\mathsf{L}}(\mathcal{C}) - A_j - 1}{M_{j-1}} \right\rfloor.$$

Proof. We start by puncturing the code in the positions of the smallest possible Lee weight. To identify these positions, we use the support subtype. Clearly, in the ideal $\langle p^i \rangle$, we have as largest possible Lee weight $M_i = \lfloor \frac{p^{s-i}}{2} \rfloor p^i$, and thus we would start puncturing in the positions, where all codewords lie within $\langle p^{s-1} \rangle$, i.e., in the positions belonging to the support subtype n_{s-1}. We hence assume that the minimum distance between two distinct tuples decreases by $A_{s-1} = n_{s-1}M_{s-1}$. If this is still smaller than the minimum Lee distance, we can continue puncturing in the next ideal, namely $\langle p^{s-2} \rangle$. We continue in this fashion, every time puncturing in $n_i M_i$ positions, until $A_j = \sum_{i=j}^{s-1} n_i M_i$ has reached the minimum Lee distance. At this point we are left with codewords that are at least $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) - A_j$ apart, thus we can continue puncturing in $\left\lfloor \frac{\mathrm{d}_{\mathsf{L}}(\mathcal{C}) - A_j - 1}{M_{j-1}} \right\rfloor$ positions living in $\langle p^{j-1} \rangle$, i.e., belonging to the support subtype n_{j-1}, and still be sure that the punctured code has the same size as the original code. In this case, we have the new length of the punctured code, being $n - B_j - \left\lfloor \frac{\mathrm{d}_{\mathsf{L}}(\mathcal{C}) - A_j - 1}{M_{j-1}} \right\rfloor$, for $B_j = \sum_{i=j}^{s-1} n_i$. □

Example 4.0.2. Let us consider $\mathcal{C} \subseteq (\mathbb{Z}/9\mathbb{Z})^4$ generated by

$$G = \begin{pmatrix} 1 & 0 & 2 & 3 \\ 0 & 3 & 6 & 0 \\ 0 & 0 & 3 & 6 \end{pmatrix}.$$

The $\mathbb{Z}/9\mathbb{Z}$-dimension of this code is $k = 2$, the minimum Lee distance of this code is $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) = 6$, and the support subtype is given by $(2, 2, 0)$.

If we puncture in the second and the last column (both belonging to the ideal $\langle 3 \rangle$), we get $n_1 M_1 = 6 \not< \mathrm{d}_{\mathsf{L}}(\mathcal{C})$. Hence, we identify $j = s = 2$, and we puncture in only one of the columns corresponding to the support subtype $n_1 = 2$. In fact, $\left\lfloor \frac{\mathrm{d}_{\mathsf{L}}(\mathcal{C}) - 0 - 1}{3} \right\rfloor = 1$. That is, the bound in Theorem 4.0.1 is attained as

$$K = 3 = 4 - 0 - \left\lfloor \frac{6 - 0 - 1}{3} \right\rfloor = n - B_j - \left\lfloor \frac{\mathrm{d}_{\mathsf{L}}(\mathcal{C}) - A_j - 1}{M_{j-1}} \right\rfloor.$$

The bound from Theorem 3.4.1 would instead give

$$\left\lfloor \frac{\mathrm{d}_{\mathsf{L}}(\mathcal{C}) - 1}{\lfloor 9/2 \rfloor} \right\rfloor = \left\lfloor \frac{6 - 1}{4} \right\rfloor = 1 < 2 = n - k.$$

Since we are also in the case where k is an integer strictly larger than 1, we can also apply the bound from Theorem 3.4.2, and get

$$\mathrm{d_L}(\mathcal{C}) = 6 < 8 = (4-2)\cdot 4 = (n-k)\lfloor 9/2 \rfloor .$$

We can rewrite the bound from Theorem 4.0.1 as upper bound on the minimum Lee distance: Let j be the smallest positive integer with $A_j < \mathrm{d_L}(\mathcal{C})$, then

$$\mathrm{d_L}(\mathcal{C}) \leq M_{j-1}\left(\sum_{i=0}^{j-1} n_i - K\right) + \sum_{i=j}^{s-1} n_i M_i + \alpha,$$

for some $\alpha \in \{1, \ldots, M_{j-1}\}$. However, the condition to find the smallest positive integer j such that $A_j < \mathrm{d_L}(\mathcal{C})$ renders the bound impractical, as usually we do not know the minimum Lee distance of a code and thus want to bound it from above.

4.1 Defining Lee-supports and Generalized Lee Weights over Chain Rings

We start this section by recapping the definition of generalized Hamming weights over finite fields. We discuss their definition of a support of a vector and a code, as well as their properties, and will then discuss the adaption of these definitions to the case of the Lee metric.

Generalized Hamming weights have originally been introduced in [73, 131] over finite fields and have been studied in various areas such as [38, 51, 65, 66, 106]. In [64] the authors defined the generalized Hamming weights of ring-linear codes by considering the join-Hamming support of a code. Originally, generalized Hamming weights are based on the minimal support of a subcode $\mathcal{D} \subseteq \mathcal{C}$ of dimension $r \leq k$. Thus, let us recap the definition of a support of a vector and a code in the classical case of a finite field.

Definition 4.1.1. Given a finite field $\mathbb{F}_q$ of q elements. Let $x \in \mathbb{F}_q^n$ be a vector of length n and $\mathcal{C} \subseteq \mathbb{F}_q^n$ a code of dimension k. The *Hamming support* of x and $\mathcal{C}$, respectively, is defined by

$$\begin{aligned} \mathrm{supp_H}(x) &:= \{i = 1, \ldots, n \mid x_i \neq 0\}, \\ \mathrm{supp_H}(\mathcal{C}) &:= \{i = 1, \ldots, n \mid \exists c \in \mathcal{C} \text{ with } c_i \neq 0\}. \end{aligned}$$

Notice that it immediately follows from Definition 4.1.1 that the Hamming weight of a vector x corresponds to the cardinality of the Hamming support of x, i.e.,

$$\mathrm{wt_H}(x) = |\,\mathrm{supp_H}(x)\,|.$$

Similarly, we define the *Hamming weight of a code* as

$$\mathrm{wt_H}(\mathcal{C}) := |\,\mathrm{supp_H}(\mathcal{C})\,|.$$

In the classical case, the r-th generalized Hamming weights are then defined in the following way.

Definition 4.1.2. Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a linear code of dimension k. Then for any $r \in \{1, \ldots, k\}$ the *r-th generalized Hamming weight* is given by

$$\mathrm{wt_H^r}(\mathcal{C}) = \min\{\mathrm{wt_H}(\mathcal{D}) \mid \mathcal{D} \subseteq \mathcal{C}, \dim(\mathcal{D}) = r\}.$$

It is easy to see that the first generalized Hamming weight of a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ corresponds to the minimum Hamming distance of $\mathcal{C}$. Similarly, the k-th generalized Hamming weight of $\mathcal{C}$ is equal to its Hamming weight. Furthermore, generalized Hamming weights fulfill an increasing property [131]. That is, given a linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ of dimension k, then for every

$1 \leq r \leq k$ it holds

$$\mathrm{d}_{\mathsf{H}}(\mathcal{C}) = \mathrm{wt}_{\mathsf{H}}^1(\mathcal{C}) < \mathrm{wt}_{\mathsf{H}}^2(\mathcal{C}) < \ldots < \mathrm{wt}_{\mathsf{H}}^k(\mathcal{C}) = \mathrm{wt}_{\mathsf{H}}(\mathcal{C}). \tag{4.2}$$

As the inequalities in 4.2 are strict inequalities we can easily deduce the Singleton bound for non-degenerate codes, for which $\mathrm{wt}_{\mathsf{H}}(\mathcal{C}) = n$, by subtracting $(k-1)$ from the weight of the code, i.e.,

$$\mathrm{d}_{\mathsf{H}}(\mathcal{C}) < \mathrm{wt}_{\mathsf{H}}(\mathcal{C}) - (k-1) = n - k + 1.$$

Hence, when moving to finite integer rings endowed with the Lee metric, we would like to define generalized Lee weights in such a way that the Property (4.2) holds or at least a similar property. One question that arises is how to define a support in the Lee metric. In fact, viewing the support as an index set in the Lee metric is not convenient, since we would have to define $\lfloor p^s/2 \rfloor$ many support sets for each Lee weight which is not optimal. In [64] the authors interpreted the Hamming support of a vector of length n as an n-tuple, where each position of the support is given by the Hamming weight of the vector at this position, i.e., for $x \in \mathbb{F}_q^n$

$$\mathrm{supp}_{\mathsf{H}}(x) = (\mathrm{wt}_{\mathsf{H}}(x_1), \ldots, \mathrm{wt}_{\mathsf{H}}(x_n)).$$

As we are now working with tuples of length n, we will introduce additional notation. We define the *cardinality* of a tuple s as the sum of its entries, i.e.,

$$|\, s \,| := \sum_{i=1}^{n} s_i. \tag{4.3}$$

Considering two n-tuples $s, t \in \mathbb{N}^n$, we define their *join* and *meet*, respectively, as

$$\begin{aligned} s \vee t &:= (\max\{s_1, t_1\}, \ldots, \max\{s_n, t_n\}), \\ s \wedge t &:= (\min\{s_1, t_1\}, \ldots, \min\{s_n, t_n\}). \end{aligned}$$

Note that viewing the support as a tuple and defining its cardinality as in (4.3) ensures $\mathrm{wt}_{\mathsf{H}}(x) = |\,\mathrm{supp}_{\mathsf{H}}(x)\,|$, and works in the exact same fashion for the Lee metric too. That is, for $x \in (\mathbb{Z}/p^s\mathbb{Z})^n$ we define the *Lee support* of x as

$$\mathrm{supp}_{\mathsf{L}}(x) = (\mathrm{wt}_{\mathsf{L}}(x_1), \ldots, \mathrm{wt}_{\mathsf{L}}(x_n)).$$

In order to extend this to the support of codes, we have several options. One of those, is the join-support, as considered in [64]: for $\mathcal{C} \subseteq \mathcal{R}^n$ a linear code, we define its *join-support* as

$$\mathrm{supp}_{\mathsf{join}}(\mathcal{C}) := \left(\max_{c \in \mathcal{C}} \mathrm{wt}(c_1), \ldots, \max_{c \in \mathcal{C}} \mathrm{wt}(c_n) \right) = \bigvee_{c \in \mathcal{C}} \mathrm{supp}(c).$$

Note that another possibility would be to define the *meet-support*, as follows

$$\begin{aligned} \mathrm{supp}_{\mathsf{meet}}(\mathcal{C}) &:= \left(\min_{c \in \mathcal{C}} \{\max\{\mathrm{wt}(c_1), 0\}\}, \ldots, \min_{c \in \mathcal{C}} \{\max\{\mathrm{wt}(c_n), 0\}\} \right) \\ &= \bigwedge_{c \in \mathcal{C}} (\mathrm{supp}(c) \vee 0). \end{aligned}$$

As the Hamming weight of nonzero elements equals one, we observe that the join-support coincides with the meet-support of a code $\mathcal{C}$ in the Hamming metric, i.e.,

$$\mathrm{supp}_{\mathsf{H},\mathsf{join}}(\mathcal{C}) = \mathrm{supp}_{\mathsf{H},\mathsf{meet}}(\mathcal{C}).$$

Example 4.1.3. Let us consider a code over $\mathbb{F}_3$ generated by

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

With the usual definition of the Hamming support in Definition 4.1.1, we have that

$$\operatorname{supp}_{\mathsf{H}}(\mathcal{C}) = \{1, 2, 3, 4\}.$$

With the join-support, we are considering the maximal value of the weight of the entries of a codeword in each position, that is

$$\operatorname{supp}_{\mathsf{H},\mathsf{join}}(\mathcal{C}) = (1, 1, 1, 1, 0).$$

For the meet-support, we take the minimum nonzero value of the weight of the entries of a codeword in each position which also gives $(1, 1, 1, 1, 0)$.
By applying the corresponding definition of the weight of a code we observe that all the three support definitions of $\mathcal{C}$ yield the same weight

$$\begin{aligned} \operatorname{wt}_{\mathsf{H}}(\mathcal{C}) &= |\operatorname{supp}_{\mathsf{H}}(\mathcal{C})| = 4, \\ \operatorname{wt}_{\mathsf{H},\mathsf{join}}(\mathcal{C}) &= \operatorname{wt}_{\mathsf{H},\mathsf{meet}}(\mathcal{C}) = 4. \end{aligned}$$

In the following Sections 4.2 and 4.3 we study the adaption of the Hamming supports as a tuple in the case of the Lee metric.

For the definition of a generalized weight over finite integer rings, we have to exchange the fixed dimension of the subcodes with a ring-analogue parameter. A natural choice would be the $\mathbb{Z}/p^s\mathbb{Z}$-dimension, but as this value is not necessarily an integer and there might not exist subcodes of $\mathcal{C}$ of certain fixed smaller rational number as the $\mathbb{Z}/p^s\mathbb{Z}$-dimension, we choose to discard this option.

In [51], the authors chose to exchange the dimension with the subtype. In fact, in the same paper the authors defined generalized Lee weights for $\mathbb{Z}/4\mathbb{Z}$. This particular case is, however, not of interest for us, as the Lee-metric Singleton-like bound over $\mathbb{Z}/4\mathbb{Z}$ directly follows from the Gray isometry [67]. Following the idea of [51], a first attempt on defining generalized weights over $\mathbb{Z}/p^s\mathbb{Z}$ would be the following.

Definition 4.1.4. Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of subtype $(k_0, \dots, k_{s-1})$. Then for any $(r_0, \dots, r_{s-1})$ with $r_i \leq k_i$ for all $i \in \{0, \dots, s-1\}$ the $(r_0, \dots, r_{s-1})$-th generalized weight is given by

$$\mathrm{d}^{(r_0,\dots,r_{s-1})}(\mathcal{C}) = \min\{\operatorname{wt}(\mathcal{D}) \mid \mathcal{D} \subseteq \mathcal{C}, \mathcal{D} \text{ has subtype } (r_0, \dots, r_{s-1})\}.$$

Note that this definition is not considering all possible subcodes or all possible subtypes of subcodes. To allow for a comparison between two different subtypes $(r_0, \dots, r_{s-1})$ and $(r'_0, \dots, r'_{s-1})$ which might have $r_i < r'_i$ for some i but $r_j > r'_j$ for some j, a natural choice is to impose a lexicographical order, i.e., we consider the order

$$(k_0, \dots, k_{s-1}) > (k_0 - 1, \dots, k_{s-1}) > \cdots > (0, k_1, \dots, k_{s-1}) > \cdots > (0, \dots, 0, 1).$$

However, the property $\mathrm{d}(\mathcal{C}) = \mathrm{d}^{(0,\dots,0,1)}(\mathcal{C})$ is then not guaranteed. In fact, a minimum Lee weight codeword will lie within a subcode having subtype one of the standard vectors e_i. Thus, we have $\mathrm{d}(\mathcal{C}) = \mathrm{d}^{e_i}(\mathcal{C})$ for some unknown i. Observing that this just means to fix the rank of the subcode to 1, we choose to directly fix the rank instead.

Definition 4.1.5. Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K. Then for any $r \in \{1, \dots, K\}$ the r-th generalized weight is given by

$$\mathrm{d}^r(\mathcal{C}) = \min\{\operatorname{wt}(\mathcal{D}) \mid \mathcal{D} \leq \mathcal{C}, \operatorname{rk}(\mathcal{D}) = r\}.$$

4.2 Generalized Join-Lee Weight

With an eye on the definition of generalized weights in the Hamming metric seen in Section 4.1, we introduce and discuss in this section generalized Lee weights with respect to the join-support derived from the Hamming metric. Recall from Section 4.1 that defining a support in terms of an index set in the Lee metric is challenging. Therefore, a support in the Lee metric will always be a tuple storing the Lee weights. In the Hamming metric, we observed that the Hamming support of a vector $x \in (\mathbb{Z}/p^s\mathbb{Z})^n$ as an n-tuple of Lee weights has two equivalent descriptions, namely the join-support and the meet-support defined in 4.1 and 4.1.

We want to define the Lee support and hence the generalized Lee weights similar to the Hamming metric case. For $x \in (\mathbb{Z}/p^s\mathbb{Z})^n$ we view the Lee support as an n-tuple and define it analogously to the Hamming support, i.e.,

$$\operatorname{supp}_{\mathsf{L}}(x) := (\operatorname{wt}_{\mathsf{L}}(x_1), \dots, \operatorname{wt}_{\mathsf{L}}(x_n)).$$

As we want to proceed as in the Hamming metric, to define a Lee support for a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ we have two options: the join-Lee support and the meet-Lee support. Owing to our ultimate goal of deriving a bound on the minimum Lee distance, and hence on defining generalized Lee weights that satisfy a property similar to (4.2), we now quickly discuss why, in the Lee metric, the meet-support is not a suitable choice.

Definition 4.2.1. For a code $\mathcal{C} \subset (\mathbb{Z}/p^s\mathbb{Z})^n$ we define the *meet-Lee support* as the minimal (if possible) nonzero Lee weight in each position among all codewords, meaning that

$$\operatorname{supp}_{\mathsf{L},\mathsf{meet}}(\mathcal{C}) := \left(\min_{c\in\mathcal{C}}\{\max\{\operatorname{wt}_{\mathsf{L}}(c_1), 0\}\}, \dots, \min_{c\in\mathcal{C}}\{\max\{\operatorname{wt}_{\mathsf{L}}(c_n), 0\}\} \right).$$

As the meet support is defined over the entries of the codewords, we can describe the meet-Lee weight of the code using its support subtype.

Proposition 4.2.2. *For $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of support subtype $(n_0, \dots, n_{s-1}, 0)$, we have that*

$$\left|\operatorname{supp}_{\mathsf{L},\mathsf{meet}}(\mathcal{C})\right| = \operatorname{wt}_{\mathsf{L},\mathsf{meet}}(\mathcal{C}) = \sum_{i=0}^{s-1} n_i p^i.$$

Proof. The meet-Lee support asks to take the smallest nonzero Lee weight in position j and then to sum over all entries $j \in \{1, \dots, n\}$. Since any position belonging to the support subtype n_i lies in the ideal $\langle p^i \rangle$, this position has as smallest nonzero Lee weight p^i. □

We can then define the r-th generalized meet-Lee weights.

Definition 4.2.3. Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K. For $r \in \{1, \dots, K\}$, we define the r-th generalized meet-Lee weight as

$$\mathrm{d}^r_{\mathsf{L},\mathsf{meet}}(\mathcal{C}) = \min\left\{ \left|\operatorname{supp}_{\mathsf{L},\mathsf{meet}}(\mathcal{D})\right| \mid \mathcal{D} \leq \mathcal{C}, \operatorname{rk}(\mathcal{D}) = r \right\}.$$

Unfortunately, this definition of a generalized Lee weight does not allow us to deduce a bound on the minimum Lee distance of a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ as the desired property (4.2) does not always hold. In fact, already the first generalized meet-Lee weight does not fulfil the property as

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq \mathrm{d}^1_{\mathsf{L},\mathsf{meet}}(\mathcal{C}).$$

Example 4.2.4. Consider the code $\mathcal{C} = \langle (1, 2) \rangle \subseteq (\mathbb{Z}/9\mathbb{Z})^2$. It is easy to see that $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) = 3$ given by the minimum Lee weight codewords $(1, 2)$ and $(8, 7)$. However, the first generalized meet-Lee weight is $\mathrm{d}^1_{\mathsf{L},\mathsf{meet}}(\mathcal{C}) = 2$ given by the minimal Lee-support $\operatorname{supp}_{\mathsf{L},\mathsf{meet}}\langle (1, 2) \rangle = (1, 1)$.

We therefore focus on the definition of the join-support in the Lee metric as it was also promoted in [64].

Definition 4.2.5. For a code $\mathcal{C} \subset (\mathbb{Z}/p^s\mathbb{Z})^n$ its *join-Lee support* is defined as the maximal possible Lee weight in each position among all codewords, i.e.,

$$\operatorname{supp}_{\mathsf{L},\mathsf{join}}(\mathcal{C}) := (\max\{\operatorname{wt}_{\mathsf{L}}(c_1) \mid c \in \mathcal{C}\}, \ldots, \max\{\operatorname{wt}_{\mathsf{L}}(c_n) \mid c \in \mathcal{C}\}).$$

Similarly to the meet support, we can express the join-Lee weight of a code using its support subtype. Notice, that the join-Lee support asks for the maximum Lee weight in a given column. Since, similarly to the meet-Lee support, we compute the minimal ideal containing all entries of the column, we need the maximum possible Lee weight in each ideal.

Proposition 4.2.6. *For $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of support subtype $(n_0, \ldots, n_{s-1}, 0)$, we have that*

$$\left|\operatorname{supp}_{\mathsf{L},\mathsf{join}}(\mathcal{C})\right| = \operatorname{wt}_{\mathsf{L},\mathsf{join}}(\mathcal{C}) = \sum_{i=0}^{s-1} n_i M_i.$$

Proof. In each index $j \in \{1, \ldots, n\}$, we can check in which minimal ideal this coordinate of the code lies. Let us assume that this is $\langle p^i \rangle$, for some $i \in \{0, \ldots, s-1\}$. Since the support of the code takes the maximum over all codewords in the code, we will reach in this entry the maximal Lee weight of the ideal $\langle p^i \rangle$, which is given by $M_i = \lfloor \frac{p^{s-i}}{2} \rfloor p^i$. Since we know the support subtype of the code, we know that we have n_i many of these entries. □

The r-th generalized join-Lee weight is then defined as follows.

Definition 4.2.7. Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K. For $r \in \{1, \ldots, K\}$, we define the r-th generalized join-Lee weight as

$$\mathrm{d}^r_{\mathsf{L},\mathsf{join}}(\mathcal{C}) = \min\{\operatorname{wt}_{\mathsf{L},\mathsf{join}}(\mathcal{D}) \mid \mathcal{D} \leq \mathcal{C}, \operatorname{rk}(\mathcal{D}) = r\}.$$

Let us consider an example which also shows the differences between the meet-Lee support and the join-Lee support.

Example 4.2.8. Let us consider a code $\mathcal{C} \subseteq (\mathbb{Z}/9\mathbb{Z})^4$ generated by

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

which has support subtype $(4, 0, 0)$ and minimum Lee distance 2 (given, for instance, by the codeword $(1, 0, 0, 8)$). For the generalized meet-Lee weights we compute

$$\begin{aligned} \mathrm{d}^1_{\mathsf{L},\mathsf{meet}}(\mathcal{C}) &= \operatorname{wt}_{\mathsf{L},\mathsf{meet}}(\langle (0,1,2,0) \rangle) = 2, \\ \mathrm{d}^2_{\mathsf{L},\mathsf{meet}}(\mathcal{C}) &= \operatorname{wt}_{\mathsf{L},\mathsf{meet}}\left(\left\langle \begin{pmatrix} 1 & 0 & 3 & 2 \\ 0 & 1 & 2 & 0 \end{pmatrix} \right\rangle\right) = 4, \\ \mathrm{d}^3_{\mathsf{L},\mathsf{meet}}(\mathcal{C}) &= \operatorname{wt}_{\mathsf{L},\mathsf{meet}}(\langle G \rangle) = 4 = \operatorname{wt}_{\mathsf{L},\mathsf{meet}}(\mathcal{C}). \end{aligned}$$

Therefore, we observe

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \geq \mathrm{d}^1_{\mathsf{L},\mathsf{meet}}(\mathcal{C}) \leq \mathrm{d}^2_{\mathsf{L},\mathsf{meet}}(\mathcal{C}) = \mathrm{d}^3_{\mathsf{L},\mathsf{meet}}(\mathcal{C}) = \operatorname{wt}_{\mathsf{L},\mathsf{meet}}(\mathcal{C}).$$

For the generalized join-Lee weights we have that

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq \mathrm{d}^1_{\mathsf{L},\mathsf{join}}(\mathcal{C}) < \mathrm{d}^2_{\mathsf{L},\mathsf{join}}(\mathcal{C}) < \mathrm{d}^3_{\mathsf{L},\mathsf{join}}(\mathcal{C}) \leq \operatorname{wt}_{\mathsf{L},\mathsf{join}}(\mathcal{C}),$$

since

$$\begin{aligned} \mathrm{d}^1_{\mathsf{L},\mathsf{join}}(\mathcal{C}) &= \operatorname{wt}_{\mathsf{L},\mathsf{join}}(\langle (0,0,3,3) \rangle) = 6, \\ \mathrm{d}^2_{\mathsf{L},\mathsf{join}}(\mathcal{C}) &= \operatorname{wt}_{\mathsf{L},\mathsf{join}}\left(\left\langle \begin{pmatrix} 0 & 0 & 3 & 3 \\ 3 & 0 & 0 & 6 \end{pmatrix} \right\rangle\right) = 9, \\ \mathrm{d}^3_{\mathsf{L},\mathsf{join}}(\mathcal{C}) &= \operatorname{wt}_{\mathsf{L},\mathsf{join}}(\mathcal{C} \cap \langle 3 \rangle) = 12, \\ \operatorname{wt}_{\mathsf{L},\mathsf{join}}(\mathcal{C}) &= 16. \end{aligned}$$

This example already gives an idea about the relation of the r-th generalized join-Lee weights. To understand their properties better, we prove that subcodes of rank r attaining the r-th generalized join-Lee weight all lie within the $\mathcal{C} \cap \mathbb{Z}/p^{s-1}\mathbb{Z}$ which we refer to as the *socle of the code*.

Proposition 4.2.9. *The subcodes which attain the r-th generalized join-Lee weights all lie within the socle $\mathcal{C}_{s-1} = \mathcal{C} \cap \langle p^{s-1} \rangle$.*

Proof. By contradiction, assume that $\mathcal{D} \leq \mathcal{C}$ of rank r achieves the r-th generalized Lee weight $\mathrm{d}^r_{\mathsf{L},\mathsf{join}}(\mathcal{C})$ and $\mathcal{D}$ does not lie within the socle. That is, if $\mathcal{D}$ has support subtype $(n_0, \ldots, n_{s-1}, 0)$, then for some $i < s-1$ we have $n_i \neq 0$. Thus,

$$\mathrm{d}^r_{\mathsf{L},\mathsf{join}}(\mathcal{C}) = \left| \operatorname{supp}_{\mathsf{L},\mathsf{join}}(\mathcal{D}) \right| \leq \sum_{i=1}^{s-1} n_i M_i.$$

By considering the subcode $\mathcal{D}_{s-1} = \mathcal{D} \cap \langle p^{s-1} \rangle$, which is still of rank r, we observe that its support subtype is $(0, \ldots, 0, n_0 + \cdots + n_{s-1}, 0)$. Then, by Proposition 4.2.6,

$$\mathrm{wt}_{\mathsf{L},\mathsf{join}}(\mathcal{D}_{s-1}) = M_{s-1}(n_0 + \cdots + n_{s-1}) < \sum_{i=1}^{s-1} n_i M_i,$$

since $M_{s-1} < M_i$ for all $i < s-1$. This gives a contradiction to the minimality of the subcode $\mathcal{D}$. □

By Proposition 4.2.9, it is hence enough to only consider the generalized join-Lee weights of the socle $\mathcal{C} \cap \langle p^{s-1} \rangle$.

Corollary 4.2.10. *Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K. Then for all $r \in \{1, \ldots, K\}$ we have*

$$\mathrm{d}^r_{\mathsf{L},\mathsf{join}}(\mathcal{C}) = \mathrm{d}^r_{\mathsf{L},\mathsf{join}}(\mathcal{C} \cap \langle p^{s-1} \rangle).$$

This property gives us an immediate relation to the generalized Hamming weights. In fact, the socle can be considered as a code over $\mathbb{F}_p$ and the subcodes which attain the minimal join-Lee support are then those which attain the minimal Hamming support.

Corollary 4.2.11. *Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K. Then for all $r \in \{1, \ldots, K\}$ we have*

$$\mathrm{d}^r_{\mathsf{L},\mathsf{join}}(\mathcal{C}) = \mathrm{d}^r_{\mathsf{H}}(\mathcal{C}) M_{s-1}.$$

We can the use the properties of the generalized Hamming weights to derive the following properties of the generalized join-Lee weights.

Proposition 4.2.12. *Let $\mathcal{C} \subset (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K. Then we have*

1. $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq \mathrm{d}^1_{\mathsf{L},\mathsf{join}}(\mathcal{C})$.
2. $\mathrm{d}^r_{\mathsf{L},\mathsf{join}}(\mathcal{C}) < \mathrm{d}^{r+1}_{\mathsf{L},\mathsf{join}}(\mathcal{C})$ *for every* $1 \leq r < K$.
3. $\mathrm{d}^K_{\mathsf{L},\mathsf{join}}(\mathcal{C}) \leq \mathrm{wt}_{\mathsf{L},\mathsf{join}}(\mathcal{C})$.

Proof. The first property follows immediately from the definition of the join-Lee support of a tuple x. It can be tight, whenever the minimal Lee weight codeword is in the socle, which is not necessary. For the second property we simply use Corollary 4.2.11 and the third property simply follows from the definition of join-Lee support. □

We want to note here that we do not recover the exact properties of the generalized Hamming weight codes. In fact, we do not have $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) = \mathrm{d}^1_{\mathsf{L},\mathsf{join}}(\mathcal{C})$ and $\mathrm{wt}_{\mathsf{L},\mathsf{join}}(\mathcal{C}) = \mathrm{d}^K_{\mathsf{L},\mathsf{join}}(\mathcal{C})$. This seems to be the price we have to pay in order to drop the absolute homogeneity property

and to be able to consider the Lee metric. However, unlike the meet-Lee support we get a nice chain of inequalities

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq \mathrm{d}^1_{\mathsf{L},\mathsf{join}}(\mathcal{C}) < \mathrm{d}^2_{\mathsf{L},\mathsf{join}}(\mathcal{C}) < \cdots < \mathrm{d}^K_{\mathsf{L},\mathsf{join}}(\mathcal{C}) \leq \mathrm{wt}_{\mathsf{L},\mathsf{join}}(\mathcal{C}),$$

which gives us a new Lee-metric Singleton-like bound.

Theorem 4.2.13. *Let $\mathcal{C} \subset (\mathbb{Z}/p^s\mathbb{Z})^n$ be a (non-degenerate) linear code of rank K. Then we have*

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq M_{s-1}(n-K+1) = \left\lfloor \frac{p}{2} \right\rfloor p^{s-1}(n-K+1).$$

Proof. Using the properties 1.-3. from Proposition 4.2.12 we know that

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq \mathrm{d}^K_{\mathsf{L},\mathsf{join}}(\mathcal{C}) - \sum_{i=1}^{K-1} \mathrm{d}^i_{\mathsf{L},\mathsf{join}}(\mathcal{C}) - \mathrm{d}^{i-1}_{\mathsf{L},\mathsf{join}}(\mathcal{C}).$$

Let us denote $x_i = \mathrm{d}^i_{\mathsf{L},\mathsf{join}}(\mathcal{C}) - \mathrm{d}^{i-1}_{\mathsf{L},\mathsf{join}}(\mathcal{C})$. Then, by Corollary 4.2.11, we know that

$$x_i = M_{s-1}.$$

Assuming that the code is non-degenerate, we get the claim by using that

$$\mathrm{d}^K_{\mathsf{L},\mathsf{join}}(\mathcal{C}) = \sum_{i=0}^{s-1} n_i M_{s-1} = nM_{s-1}.$$

□

Note that we could have gotten this bound also by directly using

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq \mathrm{d}^1_{\mathsf{L},\mathsf{join}}(\mathcal{C}) = \mathrm{d}^1_{\mathsf{H}}(\mathcal{C})M_{s-1} = \mathrm{d}_{\mathsf{H}}(\mathcal{C})M_{s-1} \leq (n-K+1)M_{s-1}.$$

This new Singleton bound is sharper than the previously known Lee-metric Singleton-like bounds, for example the bound from Theorem 3.4.1.

Note that for MDS codes, we actually know all r-th generalized Hamming weights: let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a linear code of dimension k, then for all $r \in \{1, \ldots, k\}$

$$\mathrm{d}^r_{\mathsf{H}}(\mathcal{C}) = n - k + r. \tag{4.4}$$

Thus, a natural question that arises, is whether the optimal codes with respect to the newly defined Lee-metric Singleton-like bound have a similar behaviour. That is, we are interested in an expression for the r-th generalized join-Lee weight $\mathrm{d}^r_{\mathsf{L},\mathsf{join}}(\mathcal{C})$ for every $r \in \{1, \ldots, K\}$. Indeed, such an expression does exist and is given in Proposition 4.2.14. The closed form expression can immediately be derived from the result on the r-th generalized Hamming weight given in (4.4).

Proposition 4.2.14. *Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be code of rank K attaining the bound in Theorem 4.2.13. Then, for each $r \in \{1, \ldots, K\}$, the r-th generalized join-Lee weight is given by*

$$\mathrm{d}^r_{\mathsf{L},\mathsf{join}}(\mathcal{C}) = M_{s-1}(n-K+r).$$

Proof. This immediately follows from Corollary 4.2.11, as

$$\mathrm{d}^r_{\mathsf{L},\mathsf{join}}(\mathcal{C}) = M_{s-1}\,\mathrm{d}^r_{\mathsf{H}}(\mathcal{C}) = M_{s-1}(n-K+r).$$

□

4.3 Generalized Column-Lee Weight

We observe that in order to compute the r-th generalized Hamming weight of a code $\mathcal{C}$, we consider a generator matrix G and count the number of nonzero columns, i.e., the column weight. However, since G is not unique, choosing r rows of G which attain the minimal column weight for G does not immediately give rise to the r-th generalized Hamming weight (as we will see in Example 4.3.3). To compute the r-th generalized Hamming weight we would, hence, choose r rows of a generator matrix of minimal column weight.

Let us denote by $\mathcal{R}$ any ring. For a matrix $A \in \mathcal{R}^{K\times n}$ we will denote by $S_r(A) \in \mathcal{R}^{r\times n}$ all the submatrices of A of size $r \times n$.

Definition 4.3.1. Consider a matrix $A = (a_1^\top \cdots a_n^\top) \in \mathcal{R}^{K\times n}$. We define the *column weight*, $\mathrm{wt}_{\mathsf{col}}(A)$, of A by the number of nonzero columns of A, i.e.,

$$\mathrm{wt}_{\mathsf{col}}(A) := \left|\left\{i \in \{1,\dots,n\} \mid a_i \neq 0 \in \mathcal{R}^K\right\}\right|.$$

The *column support*, $\mathrm{supp}_{\mathsf{col}}(A)$, of A is given by

$$\mathrm{supp}_{\mathsf{col}}(A) := (\max\{\mathrm{supp}(a_1)\},\dots,\max\{\mathrm{supp}(a_n)\}).$$

Again we have the nice property that $|\,\mathrm{supp}_{\mathrm{col}}(A)\,| = \mathrm{wt}_{\mathsf{col}}(A)$. In fact,

$$\mathrm{wt}_{\mathsf{col}}(A) = |\,\mathrm{supp}_{\mathrm{col}}(A)\,| = \sum_{i=1}^{n} \max\{\mathrm{supp}(a_i)\}.$$

Thus, we can define the column support, column weight and the generalized column weights of a code.

Definition 4.3.2. Let $\mathcal{C} \subseteq \mathcal{R}^n$ be a linear code of rank K. The *column support* of $\mathcal{C}$ is given by the minimal column support of any generator matrix, i.e.,

$$\mathrm{supp}_{\mathrm{col}}(\mathcal{C}) = \min_{G:\langle G\rangle=\mathcal{C}} \mathrm{supp}_{\mathrm{col}}(G).$$

The *column weight* of a code is then given by the size of the column support, i.e.,

$$\mathrm{wt}_{\mathsf{col}}(\mathcal{C}) = |\,\mathrm{supp}_{\mathrm{col}}(\mathcal{C})\,|.$$

Finally, the *r-th generalized column weight* of $\mathcal{C}$ is defined as

$$\mathrm{d}^r_{\mathsf{col}}(\mathcal{C}) = \min\{\mathrm{wt}_{\mathsf{col}}(\mathcal{D}) \mid \mathcal{D} \leq \mathcal{C}, \mathrm{rk}(\mathcal{D}) = r\}. \tag{4.5}$$

Note that the definition of the r-th generalized column weight of a linear code $\mathcal{C} \subset \mathcal{R}^n$ of rank K is equivalent to

$$\mathrm{d}^r_{\mathsf{col}}(\mathcal{C}) = \min\{\mathrm{wt}_{\mathsf{col}}(S_r(G)) \mid \mathrm{rk}(\langle S_r(G)\rangle) = r, \langle G\rangle = \mathcal{C}\}.$$

The difficulty of this new definition lies in the choice of the generator matrix instead of the choice of the subcode. This is the only difference to the usual definition of join support and join weight. However, this task is equivalently hard.

Let us show the dependency on the choice of generator matrix in the following example.

Example 4.3.3. Let us consider $\mathcal{C} \subseteq \mathbb{F}_2^5$ generated by

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

If we were to compute the column (Hamming) weights of $S_r(G)$, we would get for $S_1(G)$

$$\mathrm{wt}_{\mathsf{col}}\left(\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \end{pmatrix}\right) = 3.$$

However, this is not the first generalized Hamming weight of the code. There exists a generator matrix G', such that $S_r(G')$ attains the r-th generalized Hamming weights as column weights, for each $r \in \{1, \dots, k\}$:

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

Now we can read of the r-th generalized Hamming weights easily:

$$\begin{aligned} \mathrm{d}^1_{\mathsf{col}}(\mathcal{C}) &= \mathrm{wt}_{\mathsf{col}}\left(\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \end{pmatrix}\right) = 2, \\ \mathrm{d}^2_{\mathsf{col}}(\mathcal{C}) &= \mathrm{wt}_{\mathsf{col}}\left(\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}\right) = 3, \\ \mathrm{d}^3_{\mathsf{col}}(\mathcal{C}) &= \mathrm{wt}_{\mathsf{col}}\left(\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}\right) = 5. \end{aligned}$$

Thus, the definition is not independent of the choice of generator matrix. Let us now adapt the definitions to the Lee weight.

Definition 4.3.4. Consider a matrix $A = \begin{pmatrix} a_1^\top & \cdots & a_n^\top \end{pmatrix} \in \mathcal{R}^{K \times n}$. Its *column Lee support* is given by the n-tuple

$$\mathrm{supp}_{\mathsf{L},\mathsf{col}}(A) = (\max\{\mathrm{supp}_{\mathsf{L}}(a_1)\}, \dots, \max\{\mathrm{supp}_{\mathsf{L}}(a_n)\}).$$

The *column Lee weight* of A is given by

$$\mathrm{wt}_{\mathsf{L},\mathsf{col}}(A) = \left|\, \mathrm{supp}_{\mathsf{L},\mathsf{col}}(A) \,\right| = \sum_{i=1}^{n} \max\{\mathrm{supp}_{\mathsf{L}}(a_i)\}.$$

Note that this definition asks us to choose in each column the entry of maximal Lee weight.

Example 4.3.5. Let us consider the matrix

$$G = \begin{pmatrix} 1 & 0 & 3 & 2 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 3 & 3 \end{pmatrix} \in (\mathbb{Z}/9\mathbb{Z})^{3\times 4}.$$

Then, the column Lee support and the column Lee weight of G are given by

$$\mathrm{supp}_{\mathsf{L},\mathsf{col}}(G) = (1, 1, 3, 3) \text{ and } \mathrm{wt}_{\mathsf{L},\mathsf{col}}(G) = 8.$$

We are now able to extend the definitions of column Lee support and column Lee weight to a linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K.

Definition 4.3.6. Consider a linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K. We define its *column Lee support* by the minimal column Lee weight of any generator matrix of $\mathcal{C}$, i.e.,

$$\mathrm{supp}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \min_{G:\langle G\rangle = \mathcal{C}} \mathrm{supp}_{\mathsf{L},\mathsf{col}}(G).$$

The *column Lee weight* of $\mathcal{C}$ is then given by the size of its column Lee support, i.e.,

$$\mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \left|\, \mathrm{supp}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) \,\right|.$$

As in the case for the Hamming metric, the definition is not independent on the choice of generator matrix. For this, we introduce the following matrix, called *reduced systematic* generator matrix.

Definition 4.3.7. Consider a matrix $G \in (\mathbb{Z}/p^s\mathbb{Z})^{K\times n}$ as given in (3.4). We say that G is in *reduced systematic form* if for every entry a of $A_{i,j} \in (\mathbb{Z}/p^{s+1-i}\mathbb{Z})^{k_i \times k_j}$ with $i < j \leq s$ it holds that $\mathrm{wt}_{\mathsf{L}}(a) \leq p^{j-1}$.

We will denote a matrix G in reduced systematic form by G_{rsys}. Let us give an example to clarify Definition 4.3.7.

Example 4.3.8. Consider $G \in (\mathbb{Z}/27\mathbb{Z})^{3\times 4}$

$$G = \begin{pmatrix} 1 & 14 & 11 & 0 \\ 0 & 9 & 18 & 0 \\ 0 & 0 & 9 & 18 \end{pmatrix}.$$

Note that G is in systematic form as defined in (3.4). By elementary row reduction, i.e., by subtracting suitable multiples of the rows r_j from row r_i with $1 \leq i < j \leq 3$, we obtain a matrix G_{rsys} in reduced systematic form

$$G_{\mathsf{rsys}} = \begin{pmatrix} 1 & 5 & 20 & 0 \\ 0 & 9 & 9 & 9 \\ 0 & 0 & 9 & 18 \end{pmatrix}.$$

By a similar argument used to prove Proposition 3.1.4 we observe the following.

Proposition 4.3.9. *Consider a linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of subtype $(k_0, \ldots, k_{s-1})$ and rank K. The code $\mathcal{C}$ is permutation equivalent to a code having a generator matrix in reduced systematic form.*

This new systematic form yields a natural upper bound on the column Lee weight of a code $\mathcal{C}$. For this let us now consider the support subtype outside an information set of size K of the code. Since we can always find a permutation-equivalent code which has an information set in the first K positions, we can assume that we only consider the last $n-K$ columns of a generator matrix in reduced systematic form. In order not to confuse it with the support subtype $(n_0, \ldots, n_s)$ of the entire generator matrix, we will denote the support subtype of the last $n-K$ columns of a generator matrix in reduced systematic form by $(\mu_0, \ldots, \mu_s)$.

Proposition 4.3.10. *Consider a linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of subtype $(k_0, \ldots, k_{s-1})$ and rank K, and let $(\mu_0, \ldots, \mu_{s-1})$ be the support subtypes in the last $n-K$ columns of a generator matrix in reduced systematic form. Then the column Lee weight of $\mathcal{C}$ is upper bounded by*

$$\mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) \leq \sum_{i=0}^{s-1} p^i k_i + \sum_{i=0}^{s} \mu_i M_i.$$

Proof. By Definition 4.3.6 we have

$$\mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \left| \min_{G:\langle G\rangle = \mathcal{C}} \mathrm{supp}_{\mathsf{L},\mathsf{col}}(G) \right|.$$

Furthermore, by Proposition 4.3.9, $\mathcal{C}$ admits a generator matrix $G_{\mathsf{rsys}} \in (\mathbb{Z}/p^s\mathbb{Z})^{K\times n}$ in reduced systematic form. Hence, the column Lee weight of G_{rsys} is a natural upper bound to the column Lee weight of the code, i.e.,

$$\mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) \leq \mathrm{wt}_{\mathsf{L},\mathsf{col}}(G_{\mathsf{rsys}}).$$

Due to the form of G_{rsys}, we observe that the maximum Lee weight in the first K columns is given by the entry $(G_{\mathsf{rsys}})_{i,i}$ for $i \in \{1, \ldots, K\}$. For the last $n-K$ columns we have to assume the maximal Lee weight. The support subtype $(\mu_0 \ldots, \mu_s)$ in these columns immediately tells us, how many columns are contained in which ideal. Hence, for each column lying in $\langle p^i\rangle$ (where i is maximal for this column) the maximal Lee weight is M_i. This yields the desired result. □

Let us now introduce the r-th generalized column Lee weights of a code $\mathcal{C}$.

Definition 4.3.11. Given a linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of subtype $(k_0, \ldots, k_{s-1})$ and rank K. The *r-th generalized column Lee weight* of $\mathcal{C}$ is defined as

$$\mathrm{d}^r_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \min\{\mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{D}) \mid \mathcal{D} \leq \mathcal{C}, \mathrm{rk}(\mathcal{D}) = r\}.$$

Similarly to Definition (4.5), the r-th generalized column Lee weight is equivalent to

$$\mathrm{d}^r_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \min\{\mathrm{wt}_{\mathsf{L},\mathsf{col}}(S_r(G)) \mid \mathrm{rk}(\langle S_r(G)\rangle) = r, \langle G\rangle = \mathcal{C}\}.$$

As in the Hamming-metric case, the difficulty lies now in finding a generator matrix attaining the r-th generalized column Lee weights. To visualize this, let us return to our previous example for the Lee-metric support.

Example 4.3.12. Let us consider the code $\mathcal{C} \subseteq (\mathbb{Z}/9\mathbb{Z})^4$ generated by

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

which has support subtype $(4, 0, 0)$ and minimum Lee distance 2. If we compute the minimal column weights of submatrices of G we get

$$\begin{aligned} \mathrm{wt}_{\mathsf{L},\mathsf{col}}(\begin{pmatrix} 0 & 1 & 2 & 0 \end{pmatrix}) &= 3, \\ \mathrm{wt}_{\mathsf{L},\mathsf{col}}\left(\begin{pmatrix} 0 & 1 & 2 & 0 \\ 0 & 0 & 3 & 3 \end{pmatrix}\right) &= 7, \\ \mathrm{wt}_{\mathsf{L},\mathsf{col}}(G) &= 8. \end{aligned}$$

However, there is a generator matrix of the code which is not in systematic form and which attains smaller column Lee weights:

$$G' = \begin{pmatrix} 8 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \\ 0 & 8 & 1 & 3 \end{pmatrix}.$$

The r-th generalized Lee weights are then

$$\begin{aligned} \mathrm{d}^1_{\mathsf{L},\mathsf{col}}(\mathcal{C}) &= \mathrm{wt}_{\mathsf{L},\mathsf{col}}(\begin{pmatrix} 8 & 0 & 0 & 1 \end{pmatrix}) = 2 = \mathrm{d}_{\mathsf{L}}(\mathcal{C}), \\ \mathrm{d}^2_{\mathsf{L},\mathsf{col}}(\mathcal{C}) &= \mathrm{wt}_{\mathsf{L},\mathsf{col}}\left(\begin{pmatrix} 8 & 0 & 0 & 1 \\ 0 & 1 & 8 & 0 \end{pmatrix}\right) = 4, \\ \mathrm{d}^3_{\mathsf{L},\mathsf{col}}(\mathcal{C}) &= \mathrm{wt}_{\mathsf{L},\mathsf{col}}\left(\begin{pmatrix} 8 & 0 & 0 & 1 \\ 0 & 1 & 8 & 0 \\ 0 & 0 & 3 & 0 \end{pmatrix}\right) = 6 = \mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}). \end{aligned}$$

Note that both matrices within this example are of reduced systematic form.

Lemma 4.3.13. *Let $\mathcal{C} \in (\mathbb{Z}/p^s\mathbb{Z})^n$ be code of rank K and let $G^{(i)} \in (\mathbb{Z}/p^s\mathbb{Z})^{i\times n}$ of a rank $i \in \{1, \ldots, K-1\}$ be a generator matrix of a subcode of $\mathcal{C}$ attaining $\mathrm{d}^i_{\mathsf{L},\mathsf{col}}(\mathcal{C})$. Consider $c \in \mathcal{C}$ such that $\begin{pmatrix} G^{(i)} \\ c \end{pmatrix}$ is a generator matrix of a subcode of rank $i+1$. Then,*

$$\mathrm{wt}_{\mathsf{L},\mathsf{col}}\left(\begin{pmatrix} G^{(i)} \\ c \end{pmatrix}\right) > \mathrm{wt}_{\mathsf{L},\mathsf{col}}(G^{(i)}).$$

Proof. Let us define for all columns $j \in \{1, \ldots, n\}$ the maximal Lee weight of the j-th column in $G^{(i)}$ as $A^{(i)}_j$. We clearly have

$$\mathrm{wt}_{\mathsf{L},\mathsf{col}}(G^{(i)}) \leq \mathrm{wt}_{\mathsf{L},\mathsf{col}}\left(\begin{pmatrix} G^{(i)} \\ c \end{pmatrix}\right).$$

By contradiction, let us assume that equality holds. Then,

$$\sum_{j=1}^{n} A^{(i)}_j = \sum_{j=1}^{n} \max\left\{A^{(i)}_j, \mathrm{wt}_{\mathsf{L}}(c_j)\right\} \tag{4.6}$$

and so for all $j \in \{1, \dots, n\}$ we have $\mathrm{wt}_{\mathsf{L}}(c_j) \leq A_j^{(i)}$. However, as $G^{(i)}$ attains $\mathrm{d}^i_{\mathsf{L},\mathsf{col}}(\mathcal{C})$, the sum in Equation (4.6) is minimal among all rank i subcodes of $\mathcal{C}$. Hence, there is no index $j \in \{1, \dots, n\}$ for which $\mathrm{wt}_{\mathsf{L}}(c_j) < A_j^{(i)}$ and thus for all j we have $\mathrm{wt}_{\mathsf{L}}(c_j) = A_j^{(i)}$. This implies $c_j = \pm A_j^{(i)}$.

This means that c has in every position the maximal Lee weight over all rows of $G^{(i)}$. Thus, for every row g_ℓ of $G^{(i)}$ with $\ell \in \{1, \dots, i\}$ for which $\mathrm{wt}_{\mathsf{L}}(g_\ell) > \mathrm{wt}_{\mathsf{L}}(c)$, we can add and/or subtract c to decrease its weight. For each row $\ell \in \{1, \dots, i\}$ let us define the sets

$$I_\ell^- = \{j \in \{1, \dots, n\} \mid \mathrm{wt}_{\mathsf{L}}(c_j - g_{\ell j}) < \mathrm{wt}_{\mathsf{L}}(c_j)\},$$
$$I_\ell^+ = \{j \in \{1, \dots, n\} \mid \mathrm{wt}_{\mathsf{L}}(c_j + g_{\ell j}) \leq \mathrm{wt}_{\mathsf{L}}(c_j)\}.$$

For a fixed row $\ell \in \{1, \dots, i\}$, if

$$\sum_{j \in I_\ell^-} \mathrm{wt}_{\mathsf{L}}(c_j) < \sum_{j \in I_\ell^+} \mathrm{wt}_{\mathsf{L}}(c_j),$$

we add c to the row g_ℓ. If however,

$$\sum_{j \in I_\ell^+} \mathrm{wt}_{\mathsf{L}}(c_j) \leq \sum_{j \in I_\ell^-} \mathrm{wt}_{\mathsf{L}}(c_j),$$

we subtract c from that row g_ℓ.

We consider now the new row $g'_\ell := c \pm g_\ell$ which has a strictly smaller Lee weight than c. Since the cases are similar, assume that for g_ℓ the first case is true, i.e., $\sum_{j \in I_\ell^-} \mathrm{wt}_{\mathsf{L}}(c_j) < \sum_{j \in I_\ell^+} \mathrm{wt}_{\mathsf{L}}(c_j)$, and thus we add the row c, getting $g'_\ell := c + g_\ell$. Clearly, for each position j in I_ℓ^- we added a Lee weight of at most $A_j^{(i)}$, while in each position j in I_ℓ^+ we subtracted a Lee weight of at most $A_j^{(i)}$, thus

$$\begin{aligned}
\mathrm{wt}_{\mathsf{L}}(g'_\ell) &= \sum_{j \in I_\ell^+} \mathrm{wt}_{\mathsf{L}}(g_{\ell j} + c_j) + \sum_{j \in I_\ell^-} \mathrm{wt}_{\mathsf{L}}(g_{\ell j} + c_j) \\
&< \sum_{j \in I_\ell^+} \mathrm{wt}_{\mathsf{L}}(g_{\ell j} + c_j) + \sum_{j \in I_\ell^-} A_j^{(i)} + \sum_{j \in I_\ell^-} \mathrm{wt}_{\mathsf{L}}((c_j) \\
&< \sum_{j \in I_\ell^+} \mathrm{wt}_{\mathsf{L}}(c_j) - \sum_{j \in I_\ell^+} A_j^{(i)} + \sum_{j \in I_\ell^+} A_j^{(i)} + \sum_{j \in I_\ell^-} \mathrm{wt}_{\mathsf{L}}((c_j) \\
&= \mathrm{wt}_{\mathsf{L}}(c).
\end{aligned}$$

Repeating this procedure for every row of the matrix $G^{(i)}$, obtaining the new matrix $G'^{(i)}$ of rank i, we have

$$\mathrm{wt}_{\mathsf{L},\mathsf{col}}(G'^{(i)}) < \mathrm{wt}_{\mathsf{L},\mathsf{col}}(G^{(i)}),$$

since in every row we now reduced the Lee weight, but this is a contradiction to G attaining $\mathrm{d}^i_{\mathsf{L},\mathsf{col}}(\mathcal{C})$. □

Finally, we are able to prove the desired properties for the generalized column Lee weights.

Proposition 4.3.14. *Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K. Then*

1. $\mathrm{d}^1_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \mathrm{d}_{\mathsf{L}}(\mathcal{C})$.
2. $\mathrm{d}^r_{\mathsf{L},\mathsf{col}}(\mathcal{C}) < \mathrm{d}^{r+1}_{\mathsf{L},\mathsf{col}}(\mathcal{C})$ *for all* $r < K$.
3. $\mathrm{d}^K_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C})$.

Proof. For the first property, note that the column Lee weight of a $1 \times n$ matrix is equal to the Lee weight of that n-tuple. Since a minimal Lee-weight codeword c is a rank 1 subcode of $\mathcal{C}$ with the smallest column Lee weight, it attains $\mathrm{wt}_{\mathsf{L},\mathsf{col}}(c) = \mathrm{d}^1_{\mathsf{L},\mathsf{col}}(\mathcal{C})$.

The second property follows from Lemma 4.3.13. Any $G^{(i+1)} \in (\mathbb{Z}/p^s\mathbb{Z})^{(i+1)\times n}$ attaining $\mathrm{d}^{i+1}_{\mathsf{L},\mathsf{col}}(\mathcal{C})$ we can write $G^{(i+1)} = \begin{pmatrix} G^{(i)} \\ g' \end{pmatrix}$ for some $G^{(i)} \in (\mathbb{Z}/p^s\mathbb{Z})^{i\times n}$ of rank i. Then we either have that $G^{(i)}$ already attained $\mathrm{d}^{i}_{\mathsf{L},\mathsf{col}}(\mathcal{C})$ and hence

$$\mathrm{d}^{i}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \mathrm{wt}_{\mathsf{L},\mathsf{col}}(G^{(i)}) < \mathrm{wt}_{\mathsf{L},\mathsf{col}}(G^{(i+1)}) = \mathrm{d}^{i+1}_{\mathsf{L},\mathsf{col}}(\mathcal{C}),$$

or, if $G^{(i)}$ did not attain $\mathrm{d}^{i}_{\mathsf{L},\mathsf{col}}(\mathcal{C})$, then

$$\mathrm{d}^{i}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) < \mathrm{wt}_{\mathsf{L},\mathsf{col}}(G^{(i)}) \leq \mathrm{wt}_{\mathsf{L},\mathsf{col}}(G^{(i+1)}).$$

In either case, we get that $\mathrm{d}^{i}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) < \mathrm{d}^{i+1}_{\mathsf{L},\mathsf{col}}(\mathcal{C})$.
Lastly, the third property follows immediately from the definition of the column Lee weight of a code $\mathcal{C}$. □

The properties in Proposition 4.3.14 allow us to deduce a natural Singleton-like bound for the Lee metric.

Theorem 4.3.15. *Given a linear code $\mathcal{C} \in (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K. The minimum distance of $\mathcal{C}$ is upper bounded by*

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq \mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - K + 1.$$

Proof. Using the properties given in Proposition 4.3.14 we note that

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) = \mathrm{d}^{1}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) \leq \mathrm{d}^{K}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - \sum_{i=2}^{K} \left(\mathrm{d}^{i}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - \mathrm{d}^{i-1}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) \right). \tag{4.7}$$

By the strict inequality between the generalized column Lee weights, we have a difference of at least one, i.e.,

$$\mathrm{d}^{i}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - \mathrm{d}^{i-1}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) \geq 1.$$

Since $\mathrm{d}^{K}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C})$, the desired bound follows. □

As for increasing parameters of a linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K and subtype $(k_0, \ldots, k_{s-1})$ it becomes harder to compute $\mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C})$, applying Proposition 4.3.10 we obtain a direct consequence of Theorem 4.3.15 which requires no computational effort.

Corollary 4.3.16. *Given a linear code $\mathcal{C} \in (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K. The minimum distance of $\mathcal{C}$ is upper bounded by*

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq \sum_{i=0}^{s-1} p^i k_i + \sum_{i=0}^{s} \mu_i M_i - K + 1.$$

The bounds given in Theorem 4.3.15 and Corollary 4.3.16 improve the Singleton bound by Shiromoto [119] and the one by Alderson and Huntemann [5]. In the proof of Theorem 4.3.15 we bounded the differences $\mathrm{d}^{i}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - \mathrm{d}^{i-1}_{\mathsf{L},\mathsf{col}}(\mathcal{C})$ by one for every $i = 2, \ldots, K$. However, for a relatively small rank K this bound is not very tight. The sum in Equation (4.7) is a telescoping sum, meaning that

$$\sum_{i=2}^{K} \left(\mathrm{d}^{i}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - \mathrm{d}^{i-1}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) \right) = \mathrm{d}^{K}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - \mathrm{d}^{1}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - \mathrm{d}_{\mathsf{L}}(\mathcal{C}).$$

Hence, the goal is now to derive a lower bound on the difference $\mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - \mathrm{d}_{\mathsf{L}}(\mathcal{C})$ allowing us to further tighten the Singleton-like bound.

In the following let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K and subtype $(k_0, \dots, k_{s-1})$. Let us introduce the maximal subtype $i \in \{0, \dots, s-1\}$ for which k_i is nonzero, that is

$$\sigma := \max\{i \in \{0, \dots, s-1\} \mid k_i \neq 0\}.$$

Proposition 4.3.17. *Let p be an odd prime. For a linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K and subtype $(k_0, \dots, k_{s-1})$ and maximal subtype k_σ, we get the following lower bound*

$$\mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - \mathrm{d}_{\mathsf{L}}(\mathcal{C}) \geq \sum_{i=0}^{\sigma-1} \left(\sum_{j=0}^{i} k_j \right) \lfloor p/2 \rfloor \, p^i + (k_\sigma - 1)p^\sigma.$$

Proof. Let us start by focusing on the generalized column Lee weights. Assume that $c_1 \in \mathcal{C}$ is such that $\mathrm{d}^1_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \mathrm{wt}_{\mathsf{L},\mathsf{col}}(\langle c_1 \rangle)$. By Lemma 4.3.13, we know that the generalized column Lee weights can be obtained in an iterative fashion. Hence, to find a subcode $\mathcal{D}_2$ of rank 2 we are looking for a codeword $c_2 \in \mathcal{C}$ such that $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$ is of rank 2 and such that it minimizes $\mathrm{wt}_{\mathsf{L},\mathsf{col}}\left(\begin{pmatrix} c_1 \\ c_2 \end{pmatrix}\right)$. We continue with this process until we obtain a matrix

$$G_K := \begin{pmatrix} c_1 \\ \vdots \\ c_K \end{pmatrix}$$

of rank K such that $\mathrm{wt}_{\mathsf{L},\mathsf{col}}(G_K) = \mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) = \mathrm{d}^K_{\mathsf{L},\mathsf{col}}(\mathcal{C})$.

Since the code $\mathcal{C}$ is of subtype $(k_0, \dots, k_\sigma, 0, \dots, 0)$, the rows of the matrix G_K each correspond to one of the σ blocks formed by the systematic form G_{sys} of G_K. To understand the difference of $\mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C})$ and the first generalized column Lee weight $\mathrm{d}^1_{\mathsf{L},\mathsf{col}}(\mathcal{C})$ we can think of successively removing rows from G_K until we are only left with the minimum weight codeword c_1. Thinking in the block-wise structure of a generator matrix in systematic form, at some point we will have cancelled k_i rows corresponding to the i-th block of G_{sys}. Hence, the minimal difference subtracted is

$$M_{i-1} - M_i = \lfloor p/2 \rfloor \, p^{i-1}.$$

Doing this successively for every k_i, with $i \in \{0, \dots, \sigma\}$, gives

$$\sum_{i=0}^{\sigma-1} \left(\sum_{j=0}^{i} k_j \right) \lfloor p/2 \rfloor \, p^i.$$

At this point we are left only with a block corresponding to the rows belonging to the maximal subtype k_σ. The minimal difference between the rows of the same block is given by p^σ. Hence, cancelling $(k_\sigma - 1)$ rows yields a difference of $p^\sigma(k_\sigma - 1)$ and the desired result follows. □

A natural consequence (combining Propositions 4.3.10 and 4.3.17) is the next bound on the minimum Lee distance $\mathrm{d}_{\mathsf{L}}(\mathcal{C})$ of a code $\mathcal{C}$ of given rank and subtype.

Corollary 4.3.18. *Consider a linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$, where p is an odd prime. Let $\mathcal{C}$ be of rank K and subtype $(k_0, \dots, k_{s-1})$ with maximal subtype k_σ and having support subtype $(\mu_0, \dots, \mu_{s-1})$ in the last $n - K$ positions. Then the following upper bound on the minimum Lee distance of $\mathcal{C}$ holds*

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq \sum_{i=0}^{s-1} p^i k_i + \sum_{i=0}^{s} \mu_i M_i - \left[\sum_{i=0}^{\sigma-1} \left(\sum_{j=0}^{i} k_j \right) \lfloor p/2 \rfloor \, p^i + (k_\sigma - 1)p^\sigma \right].$$

Let us give an example over $\mathbb{Z}/9\mathbb{Z}$.

Example 4.3.19. Consider again the code $\mathcal{C}$ generated by

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

over $\mathbb{Z}/9\mathbb{Z}$. In the last column, the code $\mathcal{C}$ has support subtype $(1,0,0)$ and minimum Lee distance 2. Furthermore, we observe that $\sigma = 1$ and support subtype $(1,0)$. Hence, by Corollary 4.3.18,

$$\mathrm{d_L}(\mathcal{C}) \leq 2 + 3 + 1 \cdot 4 - [2 \cdot 1 + (1-1)3] = 7.$$

Similarly to the join-support, examples of codes attaining this bound are codes generated by matrices $G = \begin{pmatrix} p^{s-1}\mathbb{I}_K & p^{s-1}A \end{pmatrix}$ for $A \in (\mathbb{Z}/p^s\mathbb{Z})^{K\times(n-K)}$, where $p = 3$. In fact, for any odd p these codes have a minimum Lee distance $d = p^{s-1}(n-K+1)$. Furthermore, we note that in the last $n-K$ columns we have support subtype $(0,\ldots,0,n-K)$ and $M_{s-1} = \lfloor p/2 \rfloor p^{s-1}$. Hence, inserting these values in the bound given in Corollary 4.3.18 gives

$$\mathrm{d_L}(\mathcal{C}) \leq p^{s-1}(1 + (n-K)\lfloor p/2 \rfloor).$$

This is equal to $\mathrm{d_L}(\mathcal{C})$ exactly if $p \in \{2,3\}$. Consider again the code $\mathcal{C} \subseteq (\mathbb{Z}/9\mathbb{Z})^4$ of rank $K = 3$ with generator matrix

$$G = \begin{pmatrix} 3 & 0 & 0 & 3 \\ 0 & 3 & 0 & 6 \\ 0 & 0 & 3 & 6 \end{pmatrix}.$$

This code has minimum Lee distance $\mathrm{d_L}(\mathcal{C}) = 6$ and subtype $(k_0, k_1) = (0,3)$. Hence, we also have $\sigma = 1$. The support subtype in the last $n-K = 1$ column is $(0,1,0)$ and $M_1 = 3$. Computing the bound in Corollary 4.3.18 gives then

$$\sum_{i=0}^{1} p^i k_i + \sum_{i=0}^{2} \mu_i M_i - (k_1 - 1)p = 3 \cdot 3 + 1 \cdot 3 - (3-1)3 = 2 \cdot 3 = 6$$

and thus this code is optimal with respect to the bound in Corollary 4.3.18. A further analysis on the density of optimal codes with respect to the column-Lee support (Corollary 4.3.18) can be found in Section 4.5.2.

4.4 Generalized Lee Distances

The resulting Lee-metric Singleton-like bounds in Theorem 4.2.13 and Corollary 4.3.18 are improving the previously known bounds. However, their optimal codes are sparse and the column-Lee weight of a code is computationally difficult to compute. We thus ask if fixing the rank of the subcode is the correct direction. In fact, a ring-linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K has very natural subcodes to consider which are all of rank K.

Definition 4.4.1. Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K. For each $i \in \{0,\ldots,s-1\}$ we define the *i-th filtration subcode* $\mathcal{C}_i$ of $\mathcal{C}$ as the intersection of $\mathcal{C}$ with the ideal $\langle p^i \rangle$, i.e.,

$$\mathcal{C}_i := \mathcal{C} \cap \langle p^i \rangle.$$

The $(s-1)$-st filtration $\mathcal{C}_{s-1}$ is commonly known as the *socle* of the code $\mathcal{C}$.

Recall from Equation (3.1) that a finite chain ring $\mathbb{Z}/p^s\mathbb{Z}$ has a natural chain of ideals. Hence, the filtration subcodes naturally form a chain of inclusions, namely

$$\mathcal{C}_{s-1} \subseteq \mathcal{C}_{s-2} \subseteq \ldots \subseteq \mathcal{C}_1 \subseteq \mathcal{C}_0 = \mathcal{C}. \tag{4.8}$$

We then define a new class of generalized Lee weights, or more concretely generalized Lee distances, coming from filtration subcodes.

Definition 4.4.2. Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code. For each $r \in \{1, \dots, s\}$ we define the *r-th generalized minimum Lee distance* of the code $\mathcal{C}$ to be the minimum distance of the filtration subcode $\mathcal{C}_{r-1}$, that is

$$\mathrm{d}_{\mathsf{L}}^r(\mathcal{C}) = \mathrm{d}_{\mathsf{L}}(\mathcal{C}_{r-1}).$$

The generalized minimum Lee distances have some natural properties that are summarized in the following.

Proposition 4.4.3. *Given a linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of subtype $(k_0, \dots, k_{s-1})$ and rank K. Let $\sigma := \max\{i \in \{0, \dots, s-1\} \mid k_i \neq 0\}$. Then the generalized minimum Lee distances satisfy*

1. $\mathrm{d}_{\mathsf{L}}^1(\mathcal{C}) = \mathrm{d}_{\mathsf{L}}(\mathcal{C})$,
2. $\mathrm{d}_{\mathsf{L}}^r(\mathcal{C}) \leq \mathrm{d}_{\mathsf{L}}^{r+1}(\mathcal{C})$ *for every* $r \in \{1, \dots s-1\}$,
3. $\mathrm{d}_{\mathsf{L}}^r(\mathcal{C}) \leq p^{r-1} + (n-k)M_{r-1}$ *for every* $r \in \{\sigma+1, \dots, s\}$.

Proof. The first and second property immediately follow from (4.8).
For the third property we observe that for every $r \in \{\sigma+1, \dots, s\}$, by applying elementary row operations, we can bring a generator matrix G_{r-1} of $\mathcal{C}_{r-1}$ in the form

$$G_{r-1} = \begin{pmatrix} p^{r-1}\mathbb{I}_K & A \end{pmatrix}, \tag{4.9}$$

where $A \in \left(p^{r-1}\mathbb{Z}/p^s\mathbb{Z}\right)^{K\times(n-K)}$. The r-th minimum Lee distance is upper bounded by the Lee weight of any row of G_{r-1}. For each row, the first K positions have a Lee weight of exactly p^{r-1}. In the last $n-K$ positions of each row we assume the maximal Lee weight given by $M_{r-1} := \left\lfloor p^{s-(r-1)}/2 \right\rfloor p^{r-1}$ and hence the inequality follows. □

Note that the upper bound on the generalized minimum Lee distances in property 3. of Proposition 4.4.3 is relatively loose. This is due to the fact, that we have assumed no knowledge about the matrix A given in (4.9).

Due to property 2. in Proposition 4.4.3, we cannot use the usual Singleton-like argument and decrease the weight of the whole code. Instead, we note that any $\mathrm{d}_{\mathsf{L}}^r(\mathcal{C})$ is a direct upper bound on the minimum Lee distance, i.e., for any $r \in \{1, \dots, s\}$ we have $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq \mathrm{d}_{\mathsf{L}}^r(\mathcal{C})$. The only question that remains, is how far we have to go down in the filtration to expect the lowest minimum Lee distance $\mathrm{d}_{\mathsf{L}}^r(\mathcal{C})$. In the following we identify several parameters of the code, that are easy to read off from any generator matrix of the code, that indicates which filtration subcode gives an appropriately low upper bound on $\mathrm{d}_{\mathsf{L}}(\mathcal{C})$.

As computing the minimum Lee distance of every subcode $\mathcal{C}_i$ is an exhausting task, especially if there is no knowledge about the structure of A, we would like to introduce some more parameters regarding A for the first filtration subcode of $\mathcal{C}$ admitting a generator matrix of the form (4.9). That is the filtration subcode $\mathcal{C}_\sigma$ with a generator matrix of the form $G_\sigma = \begin{pmatrix} p^\sigma\mathbb{I}_K & A \end{pmatrix}$, for some matrix $A \in (p^\sigma\mathbb{Z}/p^s\mathbb{Z})^{K\times(n-k)}$. Let a_{ij} denote the entry of A lying in row i and column j. For each row of A, we determine the maximal power of p appearing, and we denote it by

$$\ell_i := \max\left\{k \in \{\sigma, \dots, s-1\} \mid \exists a_{ij} : \langle a_{ij}\rangle = \langle p^k\rangle,\ K+1 \leq j \leq n\right\}.$$

Clearly, $\ell_i \geq \sigma$. Let n_i' denote the number of entries of the i-th row of A that lie within the ideal $\langle p^{\ell_i}\rangle$, i.e.,

$$n'_{\ell_i} := \left|\left\{j \in \{K+1, \dots, n\} \mid a_{ij} \in \langle p^{\ell_i}\rangle\right\}\right|.$$

For a given linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$, these parameters help us to understand the evolution of the matrix A in the generator matrices of the filtration subcodes $\mathcal{C}_{r-1}$, for $r \in \{\sigma+1, \dots, s\}$. In fact, given a generator matrix G_σ of the filtration subcode $\mathcal{C}_\sigma$ in the form (4.9), the parameters ℓ_i and n'_{ℓ_i} for a row $i \in \{1, \dots, K\}$ allow understanding at which point in the

filtration these positions become zero. More precisely, knowing ℓ_i and n'_{ℓ_i} implies that in $\mathcal{C}_{s-\ell_i+\sigma}$ there are n'_{ℓ_i} many zero entries in i-th row of A.

Knowing the number of entries turning into zero in a certain filtration subcode is a huge advantage in bounding the minimum distance of a code. Therefore, we define by $n'^{(r-1)}$ the maximal number of zeros we can get in the last $n-K$ positions of a row of a generator matrix of the filtration subcode $\mathcal{C}_{r-1}$. That is, for every $r \in \{\sigma+1, \dots, s\}$,

$$n'^{(r-1)} := \max\left\{n'_{\ell_i} \mid \ell_i > s-r+\sigma,\ i \in \{1, \dots, K\}\right\}.$$

If there is no ℓ_i with $\ell_i > s-r+\sigma$, we will set $n'^{(r-1)} = 0$. Furthermore, let $\ell^{(r-1)}$ be the corresponding value ℓ_i to $n'^{(r-1)}$, i.e.,

$$\ell^{(r-1)} := \max\left\{\ell_i \mid n'_{\ell_i} = n'^{(r-1)},\ i \in \{1, \dots, K\}\right\}.$$

We can hence refine the third property in Proposition 4.4.3 as follows.

Lemma 4.4.4. *Given a linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of subtype $(k_0, \dots, k_{s-1})$ with maximal subtype k_σ. Then, for every $r \in \{\sigma+1, \dots, s\}$, the r-th generalized Lee distance can be upper bounded by*

$$\mathrm{d}_{\mathsf{L}}^r(\mathcal{C}) = \mathrm{d}_{\mathsf{L}}(\mathcal{C}_{r-1}) \leq p^{r-1} + \left(n - K - n'^{(r-1)}\right) M_{r-1}.$$

Proof. This follows similarly as Proposition 4.4.3; by focusing on the row with the maximal number of zeros in the last $n-K$ columns of $\mathcal{C}_{r-1}$ which is captured in $n'^{(r-1)}$. Hence, the remaining $(n-K-n'^{(r-1)})$ positions are bounded by the maximal Lee weight in the considered ideal, which is given by M_{r-1}. □

Example 4.4.5. Let us consider a free code $\mathcal{C} \in (\mathbb{Z}/27\mathbb{Z})^5$ spanned by the rows of the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 21 & 6 \\ 0 & 1 & 0 & 10 & 7 \\ 0 & 0 & 1 & 18 & 8 \end{pmatrix} =: \begin{pmatrix} \mathbb{I}_3 & A \end{pmatrix}.$$

We compute

$$\begin{aligned} \ell_1 &= 1 \quad \text{and} \quad n'_{\ell_1} = 2, \\ \ell_2 &= 0 \quad \text{and} \quad n'_{\ell_2} = 2, \\ \ell_3 &= 2 \quad \text{and} \quad n'_{\ell_3} = 1. \end{aligned}$$

Let us now consider the filtration subcodes $\mathcal{C}_1$ and $\mathcal{C}_2$ in order to compute the bound given in Proposition 4.4.3. Note that in this case $\sigma = 0$ as the code is free. For $\mathcal{C}_\sigma = \mathcal{C}_0 = \mathcal{C}$, the values ℓ_i and n'_i are given above. As $\ell_3 = 2$ and $n'_3 = 1$, at the filtration subcode $\mathcal{C}_{3-2+0} = \mathcal{C}_1$ there is one entry equal to zero. Indeed, $\mathcal{C}_1 = \mathcal{C} \cap \langle 3 \rangle$ has a generator matrix of the form

$$\begin{pmatrix} 3 & 0 & 0 & 9 & 18 \\ 0 & 3 & 0 & 3 & 21 \\ 0 & 0 & 3 & 0 & 24 \end{pmatrix},$$

where the last row contains one zero element in the last 2 columns. Note that $n'^{(1)} = n'_{\ell_3} = 1$ and hence, $\mathrm{d}_{\mathsf{L}}^2(\mathcal{C}) \leq 3 + (5-3-1)12 = 15$.

Similarly, at the filtration subcode $\mathcal{C}_2 = \mathcal{C} \cap \langle 9 \rangle$ we observe two zero entries in the first row, as

$$\begin{pmatrix} 9 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 9 & 9 \\ 0 & 0 & 9 & 0 & 18 \end{pmatrix}.$$

Here we notice that $n'^{(2)} = n'_{\ell_1} = 2$ and thus $\mathrm{d}_{\mathsf{L}}^3(\mathcal{C}) \leq 9 + (5-3-2)9 = 9$.

By Proposition 4.4.3, we know that the r-th generalized Lee distances are in non-decreasing order. Therefore, for any $r \in \{\sigma+1, \dots, s\}$ the bound in Lemma 4.4.4 is a valid upper bound for the minimum Lee distance of a code $\mathcal{C}$. However, as visible in Example 4.4.5, the bounds on the r-th minimum Lee distances do not have to follow the same non-decreasing order. As they all hold as an upper bound to the minimum Lee distance of the code, the following bound is a direct consequence of Lemma 4.4.4 by choosing the smallest among the bounds given in the statement.

Corollary 4.4.6. *Given a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of subtype $(k_0, \dots, k_{s-1})$. For each $r \in \{\sigma+1, \dots, s\}$ let $\ell \geq 1$ and (ℓ, n') be the pair $(\ell^{(r-1)}, n'^{(r-1)})$ minimizing*

$$p^{s-\ell^{(r-1)}+\sigma} + \left(n - K - n'^{(r-1)}\right) M_{s-\ell^{(r-1)}+\sigma}.$$

Then the codes minimum distance is bounded by

$$\mathrm{d_L}(\mathcal{C}) \leq p^{s-\ell+\sigma} + (n - K - n')\, M_{s-\ell+\sigma}.$$

As for large rank K it is infeasible to compute this minimum, we can also derive a slightly weaker bound depending on the maximal value ℓ_i, which is easy to compute.

Corollary 4.4.7. *Given a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of subtype $(k_0, \dots, k_{s-1})$ of maximal subtype k_σ. For each $r \in \{\sigma+1, \dots, s\}$ let $\ell := \max\{\ell_i \mid i = 1, \dots, K\}$ and define the corresponding value $n' := \max\{n_{\ell_i} \mid \ell_i = \ell, \text{ for } i = 1, \dots, K\}$ Then, the minimum distance is bounded by*

$$\mathrm{d_L}(\mathcal{C}) \leq \begin{cases} p^{s-\ell+\sigma} + (n - K - n')\, M_{s-\ell+\sigma} & \text{if } \ell \geq 1, \\ p^\sigma + (n-K) M_\sigma & \text{else.} \end{cases}$$

In fact, we can identify conditions, leading to four different cases for the bound provided in Corollary 4.4.7. For this very last observation, leading to the very last Lee-metric Singleton-like bound, we first need one last definition. Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of maximal subtype k_σ and assume that $\mathcal{C}_\sigma$ is generated by $(p^\sigma \mathbb{I} \quad A)$.

Let us denote the entries of A as $a_{i,j}$, for $i \in \{1, \dots, K\}$ and $j \in \{K+1, \dots, n\}$. We define

$$N' := \max\{j \in \{K+1, \dots, n\} \mid \text{for every } i \in \{1, \dots, K\} :\ p \mid a_{i,j}\}.$$

That is N' is the maximal number of entries in a row of A, which are divisible by p.

Example 4.4.8. Let us consider the code over $\mathbb{Z}/27\mathbb{Z}$ generated by

$$G = \begin{pmatrix} 1 & 0 & 3 & 6 \\ 0 & 1 & 18 & 1 \end{pmatrix}.$$

The previous bound from Corollary 4.4.7 would take $\ell = 2$ and $n' = 1$. Instead of having $N' = 2$, as in the first row of A we have two entries that are divisible by p. In fact, this indicates the minimum Hamming weight codeword lies within the socle, in this case of Hamming weight 1. Clearly, if N' is large, it is beneficial to go until the socle.

1. Case $\ell = \sigma$ or $n'/2 \leq \frac{p^{s-\ell}-1}{p^{s-\sigma}-1}$. In this case we stay in $\mathcal{C}_\sigma$:

$$\mathrm{d_L}(\mathcal{C}) \leq p^\sigma + (n-K) M_\sigma.$$

2. Case $\ell = s$. In this case we also stay in $\mathcal{C}_\sigma$, but observed some zero entries:

$$\mathrm{d_L}(\mathcal{C}) \leq p^\sigma + (n - k - n') M_\sigma.$$

3. Case $\ell \neq \sigma$ or $\ell \neq s$ and $n'/2 \geq \frac{p^{s-\ell}-1}{p^{s-\sigma}-1}$. In this case we can move to $\mathcal{C}_{s-\ell+\sigma}$:

$$\mathrm{d_L}(\mathcal{C}) \leq p^{s-\ell+\sigma} + (n - k - n') M_{s-\ell+\sigma}.$$

4. Case: if $n' \leq N' \frac{p^{\ell-\sigma}-p^{\ell-\sigma-1}}{p^{\ell-\sigma}-1} + (n-K-2)\frac{p^{\ell-\sigma-1}-1}{p^{\ell-\sigma}-1}$. In this case we go to the socle:

$$\mathrm{d_L}(\mathcal{C}) \leq p^{s-1} + (n-K-N')M_{s-1}.$$

Note also, that instead of taking the filtration subcodes $\mathcal{C}_i = \mathcal{C} \cap \langle p^i \rangle$, we could have also considered the torsion subcodes.

Definition 4.4.9. Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$. For $i \in \{0, \ldots, s-1\}$, we call $\widetilde{\mathcal{C}}_i = \mathcal{C} \mod p^{s-i} \subseteq \left(\mathbb{Z}/p^{s-i}\mathbb{Z}\right)^n$ the *i-th torsion code*.

We can, however, immediately observe that the i-th torsion code represented as a code over the ambient space is naturally a subcode of the filtration subcode as

$$p^i\widetilde{\mathcal{C}}_i \subseteq \mathcal{C}_i \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n,$$

with $\mathrm{rk}(p^i\widetilde{\mathcal{C}}_i) = \sum_{j=0}^{i-1} k_j < \mathrm{rk}(\mathcal{C}_i) = K$.

In fact, any generator matrix of $\widetilde{\mathcal{C}}_i$ is a truncation of a generator matrix of G, i.e., we cut off the rows belonging to the subtypes $k_i, \ldots, k_{s-1}$. Thus, if we defined the r-th generalized Lee distances through the torsion subcodes, i.e., $\mathrm{d}_\mathsf{L}^r(\mathcal{C}) = \mathrm{d_L}(\widetilde{\mathcal{C}}_r)$, for $r \in \{0, \ldots, s-1\}$, then we would observe $\mathrm{d_L}(\mathcal{C}) \leq \mathrm{d_L}(\mathcal{C}_i) \leq \mathrm{d_L}(p^i\tilde{\mathcal{C}}_i)$. Thus, any upper bound on $\mathrm{d_L}(p^i\tilde{\mathcal{C}}_i)$ would serve as upper bound on $\mathrm{d_L}(\mathcal{C})$, but would be worse than taking directly bounds on the smaller $\mathrm{d_L}(\mathcal{C}_i)$.

Finally, we note that the same considerations also apply to the Hamming metric.

Corollary 4.4.10. *Given a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of subtype $(k_0, \ldots, k_{s-1})$ with maximal subtype k_σ. For each $r \in \{\sigma+1, \ldots, s\}$ let $\ell := \max\{\ell_i \mid i = 1, \ldots, K\}$ and define the corresponding value $n' := \max\{n_{\ell_i} \mid \ell_i = \ell$, for $i = 1, \ldots, K\}$. Then, the Hamming minimum distance is bounded by*

$$\mathrm{d_H}(\mathcal{C}) \leq \begin{cases} 1 + (n-K-n') & \text{if } \ell \geq 1, \\ 1 + (n-K) & \text{else.} \end{cases}$$

Note, that the Lee-metric version, that is Corollary 4.4.7, is not directly implied by the Hamming-metric bound. Such a direct bound would state

$$\mathrm{d_L}(\mathcal{C}) \leq \begin{cases} M(1 + (n-K-n')) & \text{if } \ell \geq 1, \\ M(1 + (n-K)) & \text{else.} \end{cases}$$

This bound is clearly worse than our Lee-metric Singleton-like bound of Corollary 4.4.7.

Let us consider now codes that achieve the bound on the minimum Lee distance based on filtration subcodes, i.e., Corollary 4.4.7, and check whether this fixes the r-th generalizes Lee distances. If $\mathcal{C}$ has maximal subtype k_σ and attains the bound in Corollary 4.4.7, then

$$\mathrm{d_L}(\mathcal{C}) = \mathrm{d}_\mathsf{L}^1(\mathcal{C}) = \cdots = \mathrm{d}_\mathsf{L}^{\sigma-1}(\mathcal{C}) = \mathrm{d_L}(\mathcal{C}_\sigma).$$

If $\sigma = s-1$ or if we are in case 4, i.e.,

$$n' \leq N' \frac{p^{\ell-\sigma}-p^{\ell-\sigma-1}}{p^{\ell-\sigma}-1} + (n-K-2)\frac{p^{\ell-\sigma-1}-1}{p^{\ell-\sigma}-1},$$

we consider the socle and hence all r-th generalized Lee distances $\mathrm{d}_\mathsf{L}^r(\mathcal{C})$ are equal. If we are not in case 4, the behaviour of the filtration subcodes $\mathcal{C}_r$ with $r \geq \sigma$ is more unpredictable.

As already discussed above there are codes with several properties which are attaining the bound in Corollary 4.4.7. One class of codes that we want to consider are those having $n' = n-K$. Assuming that such a code attains the bound, the following result gives us a closed expression for the r-th generalized Lee distances for all r.

Proposition 4.4.11. *Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K, subtype $(k_0, \ldots, k_{s-1})$ of maximal subtype k_σ, and tuple $(\ell, n-K)$, such that $\mathrm{d_L}(\mathcal{C}) = \mathrm{d}_\mathsf{L}^{s-\ell+\sigma}(\mathcal{C})$. Then the r-th generalized Lee distance*

is given by

$$\mathrm{d}_{\mathsf{L}}^{r}(\mathcal{C}) = \begin{cases} p^{s-\ell+\sigma} & \text{for every } r \leq s-\ell+\sigma, \\ p^{r} & \text{for every } r > s-\ell+\sigma. \end{cases}$$

Proof. Since $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) = \mathrm{d}_{\mathsf{L}}^{s-\ell+\sigma}(\mathcal{C})$ and since the r-th generalized Lee distances are increasing in r, we have $\mathrm{d}_{\mathsf{L}}^{r}(\mathcal{C}) = \mathrm{d}_{\mathsf{L}}^{s-\ell+\sigma}(\mathcal{C})$ for every $r \leq s-\ell+\sigma$. Hence, the first case is clear. For the second case, note that $\mathcal{C}_{s-\ell+\sigma}$ admits a generator matrix containing only zeros in the last $n-K$ columns. These entries remain zero for every filtration subcode $\mathcal{C}_r$ with $r > s-\ell+\sigma$. Hence, the minimum distance $\mathrm{d}_{\mathsf{L}}(\mathcal{C}_r)$ is always given by p^r. □

4.5 Comparison of the Bounds

At this point let us compare the bound of Corollary 4.4.7 to the bounds derived from the new puncturing argument (Theorem 4.0.1), to the join-Lee support (Theorem 4.2.13), to the Lee-column support (Corollary 4.3.18) and to the bounds provided by [5, 119]. We do so by providing first some examples that attain the bound from Corollary 4.4.7 and compare it to the other bounds.

Example 4.5.1. 1. Let $\mathcal{C} \subseteq (\mathbb{Z}/9\mathbb{Z})^4$ be the code generated by

$$G = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 6 \\ 0 & 0 & 1 & 4 \end{pmatrix}.$$

We observe that this code has minimum Lee distance $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) = 3$. For the last $n-K = 1$ column, we note, that all the entries lie within the ideal generated by 1. This means that $\ell = 0$ and $n' = n - K = 1$. Note that the bounds of Corollary 4.4.6 and 4.4.7 coincide. The bounds are computed as follows.

Filtration:	$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq 3$	(Corollary 4.4.6 and 4.4.7)
Join-Lee support:	$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq 6$	(Theorem 4.2.13)
Column-Lee support:	$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq 5$	(Corollary 4.3.18)
New puncturing:	$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq 8$	(Theorem 4.0.1)
Shiromoto:	$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq 8$	([119])
Alderson - Huntemann:	$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq 4$	([5])

2. Let $\mathcal{C} \subseteq (\mathbb{Z}/27\mathbb{Z})^5$ be the code generated by

$$G = \begin{pmatrix} 1 & 10 & 4 & 20 & 9 \\ 0 & 3 & 9 & 18 & 9 \end{pmatrix}.$$

The minimum Lee distance of this code is $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) = 9$. For the last $n-K = 3$ columns, we quickly compute $\ell' = 2$ and $n' = 1$. Then the bounds are computed as follows.

Filtration:	$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq 9$	(Corollary 4.4.6 and 4.4.7)
Join-Lee support:	$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq 36$	(Theorem 4.2.13)
Column-Lee support:	$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq 38$	(Corollary 4.3.18)
New puncturing:	$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq 48$	(Theorem 4.0.1)
Shiromoto:	$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq 40$	([119])
Alderson - Huntemann:	not existing	([5])

3. In this example let us consider the code $\mathcal{C} \subseteq (\mathbb{Z}/125\mathbb{Z})^6$ generated by

$$G = \begin{pmatrix} 1 & 0 & 25 & 50 & 75 & 100 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{pmatrix}.$$

(n, K, p^s, σ)	Alderson and Huntemann [5]	Shiromoto [119]	Join-Lee support (Theorem 4.2.13)	Filtration (Corollary 4.4.7) (ℓ, n')
$(6, 3, 9, 0)$	12	16	12	$(0,3)$: 13 $(1,1)$: 9 $(1,2)$: 6 $(1,3)$: 3 $(2,1)$: 9 $(2,2)$: 5 $(2,3)$: 1
$(6, 3, 9, 1)$	Not existing	16	12	$(1,\star)$: 12 $(2,1)$: 9 $(2,2)$: 6 $(2,3)$: 3
$(6, 3, 125, 0)$	186	248	200	$(0,3)$: 187 $(1,1)$: 125 $(1,2)$: 75 $(1,3)$: 25 $(2,1)$: 125 $(2,2)$: 65 $(2,3)$: 5 $(3,1)$: 125 $(3,2)$: 63 $(3,3)$: 1
$(6, 3, 125, 1)$	248 (only for subtype $(0,3,0)$)	248	200	$(1,\star)$: 185 $(2,1)$: 125 $(2,2)$: 75 $(2,3)$: 2 $(3,1)$: 125 $(3,2)$: 65 $(3,3)$: 5
$(6, 3, 125, 2)$	310 (only for subtype $(0,0,3)$) 248 (only for subtype $(1,1,1)$)	248	200	$(2,\star)$: 175 $(3,1)$: 125 $(3,2)$: 75 $(3,3)$: 25

TABLE 4.1: Comparison of the different bounds on the minimum Lee distance of a code of given parameters.

This code has minimum distance $\mathrm{d_L}(\mathcal{C}) = 5$. Note that the two bounds with respect to the filtration (Corollary 4.4.6 and 4.4.7) coincide. Hence, we obtain

Filtration:	$\mathrm{d_L}(\mathcal{C}) \leq 5$	(Corollary 4.4.6 and 4.4.7)
Join-Lee support:	$\mathrm{d_L}(\mathcal{C}) \leq 200$	(Theorem 4.2.13)
Column-Lee support:	$\mathrm{d_L}(\mathcal{C}) \leq 247$	(Corollary 4.3.18)
New puncturing:	$\mathrm{d_L}(\mathcal{C}) \leq 300$	(Theorem 4.0.1)
Shiromoto:	$\mathrm{d_L}(\mathcal{C}) \leq 249$	([119])
Alderson - Huntemann:	$\mathrm{d_L}(\mathcal{C}) \leq 248$	([5])

We now compare the bounds for different parameters. In Table 4.1, we do not consider the column-Lee support, i.e., Corollary 4.3.18, as we would need to consider too many different parameters which would not fit in the overview.

Let us focus first on a free code, i.e., $\sigma = 0$. Observe, that if the last $n - K$ columns of a generator matrix consist only of nonunits, i.e., $\ell = 0$, the bound by Alderson and Huntemann beats our bounds. However, as soon as $\ell \neq 0$ the new bound based on the minimum distance

of filtration subcodes (Corollary 4.4.7) always outperforms any other bound. In Table 4.1 we also observe, that the bound provided by Shiromoto is the loosest.

For nonfree codes, recall that the bound in [5] only works for integer $\mathbb{Z}/p^s\mathbb{Z}$-dimensions $k > 1$. Furthermore, we note that for a given $\sigma \geq 1$ we always have $\ell \geq \sigma$ and if $\ell = \sigma$ the filtration bound (Corollary 4.4.7) is the same for any n'. This is denoted by $n' = \star$ in Table 4.1. In any of the parameters presented, the bound based on the minimum Lee distance of a filtration subcode of the code (Corollary 4.4.7) outperforms all other bounds.

4.5.1 Invariance under Isometry in the Lee Metric

For the generalized Hamming weights of a linear k-dimensional code $\mathcal{C} \subseteq \mathbb{F}_q^n$, we also know that $\mathrm{d}_{\mathsf{H}}^r(\mathcal{C}) = \mathrm{d}_{\mathsf{H}}^r(\mathcal{C}')$, for any equivalent code $\mathcal{C}'$ and any $r \in \{1, \ldots, k\}$. We show here that the same holds true for all the three definitions of generalized Lee weights and distances, respectively. That is, we show that the generalized Lee weights and distances are invariant under isometries. Isometries are (usually bijective) maps between metric spaces preserving the distance properties. In our case we consider isometries from a linear Lee-metric code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ to another $\mathcal{C}' \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ preserving the Lee-distance properties of $\mathcal{C}$. By the definition of the Lee weight (see Definition 3.2.1) and the discussion on the scalar multiplication (see Section 5.3.2), the Lee-metric isometries only consist of permuting the positions and multiplying any position by 1 or -1.

Also, for the generalized join-Lee weights we have the same behaviour.

Proposition 4.5.2. *Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K, then $\mathrm{d}_{\mathsf{L,join}}^r(\mathcal{C}) = \mathrm{d}_{\mathsf{L,join}}^r(\mathcal{C}')$, for all $r \in \{1, \ldots, K\}$ and all $\mathcal{C}'$ which are equivalent to $\mathcal{C}$, under the Lee-metric isometries.*

Proof. All codewords of $\mathcal{C}'$ can be written as $c' = \sigma(c) \star v$, for some permutation $\sigma \in S_n$ and $v \in \{1, -1\}^n$, where $\star$ denotes the coordinate-wise multiplication and $c \in \mathcal{C}$. Now the claim follows immediately as

$$\begin{aligned}
\mathrm{d}_{\mathsf{L,join}}^r(\mathcal{C}) &= \min\{|(\max_{c\in\mathcal{C}}\{\mathrm{wt}_{\mathsf{L}}(c_1)\}, \ldots, \max_{c\in\mathcal{C}}\{\mathrm{wt}_{\mathsf{L}}(c_n)\})| \mid c \in \mathcal{D} \leq \mathcal{C}, \mathrm{rk}(\mathcal{D}) = r\} \\
&= \min\{|\sigma(\max_{c\in\mathcal{C}}\{\mathrm{wt}_{\mathsf{L}}(c_1)\}, \ldots, \max_{c\in\mathcal{C}}\{\mathrm{wt}_{\mathsf{L}}(c_n)\})| \mid c \in \mathcal{D} \leq \mathcal{C}, \mathrm{rk}(\mathcal{D}) = r\} \\
&= \mathrm{d}_{\mathsf{L,join}}^r(\mathcal{C}').
\end{aligned}$$

□

Similarly, we ask if the r-th generalized column-Lee weights are fixed under isometries.

Proposition 4.5.3. *Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code of rank K, then any equivalent code $\mathcal{C}' \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of $\mathcal{C}$, under the linear Lee-metric isometries is such that*

$$\mathrm{d}_{\mathsf{L,col}}^r(\mathcal{C}) = \mathrm{d}_{\mathsf{L,col}}^r(\mathcal{C}'),$$

for every $r \in \{1, \ldots, K\}$.

Proof. Recall that any generator matrix $G'^{(i)}$ of a subcode of rank i of a equivalent $\mathcal{C}'$ can be written as $G'^{(i)} = G^{(i)}P\mathrm{diag}(v)$, for some permutation matrix P, $v \in \{1, -1\}^n$ and some generator matrix $G^{(i)}$ of a subcode of rank i of $\mathcal{C}$. Both, $G'^{(i)}$ and $G^{(i)}$ have the same column weight. Now the claim follows immediately as

$$\begin{aligned}
\mathrm{d}_{\mathsf{L,col}}^r(\mathcal{C}) &= \min\{\mathrm{wt}_{\mathsf{L,col}}(G^{(r)}) \mid \langle G^{(r)}\rangle \leq \mathcal{C}, \mathrm{rk}(\langle G^{(r)}\rangle) = r\} \\
&= \min\{\mathrm{wt}_{\mathsf{L,col}}(G^{(r)}P\mathrm{diag}(v)) \mid \langle G^{(r)}\rangle \leq \mathcal{C}, \mathrm{rk}(\langle G^{(r)}\rangle) = r\} \\
&= \mathrm{d}_{\mathsf{L,col}}^r(\mathcal{C}').
\end{aligned}$$

□

Finally, we use a Lee-weight preserving isometry on $\mathbb{Z}/p^s\mathbb{Z}$ to observe that the r-th generalized Lee distance for a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ coincides with the r-th generalized Lee distance of a code $\mathcal{C}' \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ that is equivalent to $\mathcal{C}$.

Proposition 4.5.4. *Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K and let $\mathcal{C}' \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be equivalent to $\mathcal{C}$. Then, for every $r \in \{1, \dots, s\}$, we have*

$$\mathrm{d}_{\mathsf{L}}^r(\mathcal{C}) = \mathrm{d}_{\mathsf{L}}^r(\mathcal{C}').$$

Proof. Let ϕ denote an isometry preserving the Lee distance. Recall the r-th generalized Lee distance is given by the minimum Lee distance of the r-th filtration subcode $\mathcal{C}_{r-1}$ of $\mathcal{C}$, i.e.,

$$\mathrm{d}_{\mathsf{L}}^r(\mathcal{C}) = \mathrm{d}_{\mathsf{L}}(\mathcal{C}_{r-1}).$$

Since $\mathcal{C}'_{r-1} := \phi(\mathcal{C}_{r-1})$, we get that the minimum Lee distances of $\mathcal{C}_{r-1}$ and $\mathcal{C}'_{r-1}$ coincide and $\mathrm{d}_{\mathsf{L}}^r(\mathcal{C}) = \mathrm{d}_{\mathsf{L}}^r(\mathcal{C}')$. □

Hence, all three descriptions of generalized Lee weights and distances, respectively, are invariant under isometries in the Lee metric.

4.5.2 Density of Optimal Codes

Due to the one-to-one correspondence of the minimum distance of a code and its error-correction, one interesting quantity is the number of codes of maximum achievable Lee distance for given parameters. We call a such a code a *maximum Lee distance* (MLD) code. In this section we will discuss the density of MLD codes with respect to bounds derived for the different generalized weights.

Optimal Join-Lee Support Codes

Let us start by discussing the codes that attain the bound on the minimum Lee distance with respect to the join-Lee support presented in Theorem 4.2.13. Clearly, any code $\mathcal{C} \in (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K attaining this bound can be characterized by the following two properties:

1. The socle $\mathcal{C}_{s-1} = \mathcal{C} \cap \langle p^{s-1} \rangle$ is an MDS code over $\mathbb{F}_p$.
2. There exists a minimum Lee weight codeword in the socle.

The first property already implies sparsity as n tends to infinity and triviality for $p = 2$. Even the second property is problematic: $\mathrm{d}_{\mathsf{L}}(\mathcal{C}_{s-1}) = (n - K + 1)M_{s-1}$, implies that all nonzero entries of a minimal Hamming weight codeword in the socle must be of maximal Lee weight. Using the systematic form of the socle,

$$G_{s-1} = \begin{pmatrix} p^{s-1}\mathbb{I}_K & p^{s-1}A \end{pmatrix},$$

we can immediately see that any row g of G_{s-1} is also of minimal Hamming weight $n-K+1$. Thus, for g to have all nonzero entries of maximal Lee weight implies $p^{s-1} = M_{s-1}$, which will restrict optimal codes with respect to this bound to $p \in \{2, 3\}$ and any positive integer s. Assuming the MDS conjecture over $\mathbb{F}_3$ and $\mathbb{F}_2$, we must have a block length $n \leq 4$, respectively $n \leq 3$.

Example 4.5.5. The code $\mathcal{C} \subseteq (\mathbb{Z}/9\mathbb{Z})^4$ of rank $K = 3$ generated by

$$\begin{pmatrix} 3 & 0 & 0 & 3 \\ 0 & 3 & 0 & 6 \\ 0 & 0 & 3 & 6 \end{pmatrix}$$

attains the bound in Theorem 4.2.13. In fact, this code has $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) = 6$ and one can check that $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) = 6 = 3 \cdot (4 - 3 + 1) = M_{s-1}(n - K + 1)$.

We can drop the second condition, i.e., there exists a minimal Lee weight codeword in the socle, if we manage to estimate the difference

$$\mathrm{d}_{\mathsf{L},\mathsf{join}}^1(\mathcal{C}) - \mathrm{d}_{\mathsf{L}}(\mathcal{C}).$$

This task is, however, equally hard as bounding $\mathrm{d}_{\mathsf{L}}(\mathcal{C})$ itself.

Optimal Column-Lee Support Codes

We now discuss the density of codes attaining the bound provided in Corollary 4.3.18. Since for the join-support looking for a minimal Lee weight codeword is as hard as estimating the minimum distance of the code, the column-support based on generator matrices should give a similar result on the density of optimal codes. Recall that the bound is derived by

$$\mathrm{d}_{\mathsf{L}}(\mathcal{C}) \leq \mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - (\mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) - \mathrm{d}_{\mathsf{L}}(\mathcal{C})).$$

We upper bounded the column weight of the code by

$$\mathrm{wt}_{\mathsf{L},\mathsf{col}}(\mathcal{C}) \leq \sum_{i=0}^{s-1} p^i k_i + \sum_{i=0}^{s} \mu_i M_i.$$

Hence, in order to have codes attaining the bound on the minimum Lee distance, they must attain the bound on the column Lee weight too. That is, their generator matrix G must be in reduced systematic form. Furthermore, the support subtype of the last $n-K$ columns is $(\mu_0, \ldots, \mu_s)$, where in each of the μ_i positions the maximum Lee weight M_i is attained. For instance, a generator matrix may look as follows:

$$G_{\mathsf{rsys}} = \left[\begin{array}{c|ccccc} & & & & & \\ & U & & & & \\ & & & & & \\ & & \mu_0 & \mu_1 & \cdots & \mu_s \end{array}\right].$$

By the generating function (3.10), there are two options to attain a Lee weight M_i. Hence, the probability that a generator matrix is of this form is given by the number of such matrices divided by the number of all matrices, i.e.,

$$\begin{aligned} \prod_{i=0}^{s-1} \left(\frac{2(p^{s-i})^{(k-1)}}{(p^{s-i})^{(k-1)}(p^{s-i} - p^{s-i-1})} \right)^{\mu_i} &= \prod_{i=0}^{s-1} \left(\frac{2}{p^{s-i} - p^{s-i-1}} \right)^{\mu_i} \\ &= 2^{n-K} \prod_{i=0}^{s-1} \left(\frac{1}{p^{s-i}(1-1/p)} \right)^{\mu_i} \\ &= 2^{n-K} \prod_{i=0}^{s-1} \left(\frac{p^{i+1}}{p^s(p-1)} \right)^{\mu_i}. \end{aligned}$$

Note that $p^s(p-1) > p^{i+1}$ for every $i \in \{0, \ldots, s-1\}$. Hence, the fraction in the product is smaller than 1. Therefore, for $p \longrightarrow \infty$, the product tends to 0. The same argument holds if we let s tend to infinity. Similarly, as μ_i depends on n, we note that $\frac{2}{p^{s-i}-p^{s-i-1}} < 1$. This implies that if $n \longrightarrow \infty$ the product tends to zero as well. Thus, codes attaining the bound in Corollary 4.3.18 are sparse with respect to p, s and n.

Given an optimal code with respect to the Lee-metric Singleton-like bound 4.3.18, one could also ask if the r-th generalized column Lee weights are then fixed. Since the main problem of the column Lee weight of a code is the computational difficulty, we leave this as an open question.

Optimal Filtration Codes

We have already seen that codes attaining the bounds based on the Lee-join support and based on the Lee-column support are sparse as p, s and n tend to infinity. In this subsection we discuss the density of MLD codes with respect to the new Lee-metric Singleton-like bound in Corollary 4.4.7 based on the filtration. If nothing else is stated we consider a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ of rank K and subtype $(k_0, \ldots, k_{s-1})$.

Recall that the bound from Corollary 4.4.7 is especially tight, if there are many zero positions in a row of a generator matrix of a filtration subcode. Given the rank K of a code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$, the probability that an entire row of A is zero, where A are the last $n-K$ columns of a generator matrix of a filtration $\mathcal{C}_{r-1}$ with $r \in \{\sigma+1, \ldots, s\}$, is depending on

σ, i.e., it depends on whether the code $\mathcal{C}$ is free or not. For n tending to infinity, it is known [32] that

$$\mathbb{P}(\mathcal{C} \text{ is free}) = \begin{cases} 1 & \text{if } R < 1/2, \\ 0 & \text{if } R > 1/2. \end{cases}$$

Hence, in this case we would have to distinguish again the two cases. On the contrary for p going to infinity, we know from [32], that the code $\mathcal{C}$ is free with high probability, which implies that $\sigma = 0$. In this case, we have

1. For every $i \in \{1, \dots, K\}$, $\ell_i = 0$. Thus, the bound in Corollary 4.4.7 can be reduced to

$$\mathrm{d_L}(\mathcal{C}) \leq 1 + (n - K)M,$$

   which coincides with the Singleton-like bound provided by [119].

2. There is an $i \in \{1, \dots, K\}$ with $\ell_i \neq 0$. In this case, we can find the pair (ℓ, n') as in Corollary 4.4.7 and the minimum Lee distance is bounded by

$$\mathrm{d_L}(\mathcal{C}) \leq p^{s-\ell} + (n - K - n')M_{s-\ell}.$$

The following Lemma shows that for $p \longrightarrow \infty$ the first case occurs with high probability.

Lemma 4.5.6. *For a free linear code $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$, as $p \longrightarrow \infty$, $\ell = 0$ almost surely.*

Proof. Note that $\mathbb{P}(\ell = 0)$ is the probability that there is no multiple of p contained in the last $n - K$ columns of a generator matrix G in systematic form of $\mathcal{C}$. More explicitly, it is the probability that all the entries in the last $n - K$ columns of G are units. That is,

$$\mathbb{P}(\ell = 0) = \left(\frac{(p-1)p^{s-1}}{p^s}\right)^{K(n-K)} = \left(1 - \frac{1}{p}\right)^{K(n-K)}.$$

Hence, letting p grow to infinity and keeping n and K fixed, yields the desired result. □

This means that, with high probability, MLD codes are sparse as $p \longrightarrow \infty$, as codes attaining the bound in Theorem 3.4.1 of Shiromoto are sparse.

Note that, letting s grow to infinity and keeping p fixed, we get that the probability $\mathbb{P}(\ell = 0)$ is a nonzero constant. Thus, codes attaining the bound on the minimum distance derived from filtration subcodes are not sparse for $s \longrightarrow \infty$.

We start by discussing the case, where the code $\mathcal{C}$ is a free code, hence $\sigma = 0$. Free codes have a generator matrix of the form $(\mathbb{I}_K \quad A)$, with $A \in (\mathbb{Z}/p^s\mathbb{Z})^{K\times(n-K)}$. If there is an $0 < \widetilde{\ell} < s$ such that $n' = n_{\widetilde{\ell}} = n - K$, the filtration subcode $\mathcal{C}_{s-\widetilde{\ell}}$ has an entire row equal to zero. This results in having an $(s - \widetilde{\ell})$-th generalized Lee distance of $p^{s-\widetilde{\ell}}$ and hence $\mathrm{d_L}(\mathcal{C}) \leq p^{s-\widetilde{\ell}}$.
Let us investigate the probability for A having a maximal $0 < \ell_i = \widetilde{\ell} < s$ with corresponding $n' = n - K$. This requires that all other rows of A are contained at most in the ideal $\langle p^{\widetilde{\ell}} \rangle$. The probability that A is of this form is therefore

$$\begin{aligned} \mathcal{P} &:= \frac{(p^{s-\widetilde{\ell}} - p^{s-\widetilde{\ell}-1})^{(n-K)}(p^{s-1} - p^{s-\widetilde{\ell}-1})^{(K-1)}(p^{s-\widetilde{\ell}} - p^{s-\widetilde{\ell}-1})^{(n-K-1)(K-1)}}{(p^s)^{(n-K)K}} \\ &= (p^{-\widetilde{\ell}} - p^{-\widetilde{\ell}-1})^{(n-K)}(p^{s-1} - p^{-\widetilde{\ell}-1})^{(K-1)}(p^{-\widetilde{\ell}} - p^{-\widetilde{\ell}-1})^{(n-K-1)(K-1)} \\ &= \left(\frac{1}{p^{\widetilde{\ell}}} - \frac{1}{p^{\widetilde{\ell}+1}}\right)^{(n-K-1)K+1}\left(\frac{1}{p} - \frac{1}{p^{\widetilde{\ell}+1}}\right)^{(K-1)}. \end{aligned}$$

This probability tends to zero as $n \longrightarrow \infty$, and thus MLD codes are sparse with respect to the bound given in Corollary 4.4.7 and $n \longrightarrow \infty$. However, since $\mathcal{P}$ does not depend on s and hence, as $s \longrightarrow \infty$, it is a nonzero constant. This implies neither sparsity nor density for $s \longrightarrow \infty$. In any case, we have with probability $\mathbb{P}(\mathcal{C} \text{ is free}) \cdot \mathcal{P}$, that the minimum distance of the code is bounded by $\mathrm{d_L}(\mathcal{C}) \leq p^{s-\widetilde{\ell}}$.

4.6 Summary and Outlook

Following a puncturing argument to derive a Singleton-like bound on the minimum Lee distance works not as smoothly as in the Hamming metric, meaning that the resulting bound proposed by Shiromoto [119] is rather loose and can only be achieved by one nontrivial linear code. On the search of new techniques to bound the minimum Lee distance of a code, we presented several novel definitions of a support in the Lee metric interpreting the support as a tuple of weights instead of an index set. This had the advantage to be able to define the cardinality of the support of a vector to be equal to its total Lee weight and led to desired properties. Using these new definitions of a support we defined the corresponding generalized Lee weights of subcodes with a fixed rank. By the increasing property of the generalized Lee weights, we derived new bounds on the minimum Lee distance of a code which, for some parameter cases, outperformed the existing bound by Shiromoto as well as the bound provided by Alderson and Huntemann [5]. More importantly, we showed that there is more than one nontrivial linear code attaining the bounds in Theorem 4.2.13 and Corollary 4.3.18. However, their optimal codes are still sparse for n, p or s tending to infinity.

We thus abandoned the idea of defining generalized Lee weights over the support of a subcode of fixed rank. Instead, we made use of the natural chain of inclusions of integer residue rings. For a code over $\mathbb{Z}/p^s\mathbb{Z}$ we defined the filtrations as intersections of the code with a corresponding element in the chain of residue rings and observed an increasing property in terms of the minimum Lee distances of the subcodes. This lead us to the definition of generalized Lee distances and to a novel approach of bounding the minimum Lee distance of a code. Since the new bound involves many more parameters of the code, it outperforms all other bounds by far. Even though this bound is still sparse in the limit of n and p, its optimal codes are not sparse for s tending to infinity.

As none of the bounds are dense in the limit of all n, p and s, one open question that remains is to derive bounds in the minimum Lee distance whose optimal codes are dense for one or even all the parameters. In the case of the join-Lee support and the column-Lee support we were able to identify a class of codes achieving this bound. However, there are possible other constructions of codes attaining the presented bounds. Hence, a further open task that remains is the construction of optimal codes in the Lee metric for the bounds presented.

Chapter 5

Channel Coding in the Lee Metric

The Lee metric has originally been introduced in 1958 by Lee [83] to cope with phase shift keying modulation. A first notion of a channel "matching" to the Lee metric under maximum likelihood decoding appeared [42], referring to Massey's original definition of a channel matching to a given metric [91]. That is, a memoryless channel whose decoding rule "decode the received vector to the nearest codeword" always gives the most probable codeword. In terms of the Lee metric, that means that given two error vectors e and e', the error e is more likely to occur with respect to e' if and only if it has a smaller Lee weight than e'. With this Chiang and Wolf were the first to define a symmetric, memoryless "Lee Channel" over $\mathbb{Z}/q\mathbb{Z}$ as follows. For every Lee weight $i \in \{0, 1, \ldots, \lfloor q/2 \rfloor\}$, assuming that $x = 0$ has been sent, the probability that the channel output y has Lee weight i is given by

$$p_i = \mathbb{P}(y = i \mid x = 0) = \mathbb{P}(y = -i \mid x = 0).$$

They proved that this channel model indeed matches to the Lee metric under maximum likelihood decoding if and only if the probabilities p_i are exponentially decreasing in i (see [42, Theorem 1]).

In this chapter we consider two channel models in the Lee metric: a discrete memoryless channel matched to the Lee metric and a constant Lee-weight channel. The first channel is an additive discrete memoryless channel, as introduced in [42], where the additive error term follows the Boltzmann distribution [30]. We refer to this channel as *memoryless Lee channel*. The channel model is motivated by two key observations. Firstly, it matches to the Lee metric under maximum likelihood decoding following the notation in [42, 91]. Secondly, the transition probability defining the channel law arises as the marginal distribution of the second channel model presented which is especially of interest for code-based cryptography. The second is a channel where a constant-weight error pattern is added to the transmitted message, where the error pattern is chosen uniformly from the set of vectors with fixed Lee weight and length equal to the block length. We show that, in the limit of large block length, and with Lee weights that are proportional to the block length, the marginal distribution of the additive error term follows the well-known Boltzmann distribution and hence indeed coincides with the channel law of the memoryless Lee channel. For both channels we provide finite length bounds on the error probability achievable by a linear code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ making use of the entropy of the Boltzmann-like marginal distribution in Section 5.2.

With an eye on the application to code-based cryptography, drawing an error vector uniformly at random from a set of fixed Lee weight is crucial to hide information on the structure of the error. In Section 5.3.1 we give two explicit algorithms to draw a vector $a \in (\mathbb{Z}/q\mathbb{Z})^n$ of fixed Lee weight $t \in \mathbb{N}$ uniformly among all such vectors. The security of a code-based cryptosystem relies heavily on the weight of the error vector. In the Lee metric the weight of a vector can easily be modified by multiplying the vector component-wise by a nonzero scalar. From a cryptographic point of view, being able to reduce the Lee weight of an error vector possibly leads to a lack in security of the corresponding cryptographic scheme. We refer to this problem as the *scalar multiplication problem* and discuss it in Section 5.3.2.

The results in this chapter were studied in [15, 16] and [17] in collaboration with Hannes Bartz, Gianluigi Liva and Joachim Rosenthal. Section 5.3.2 contains further results on the

scalar multiplication problem that go beyond the scope of [16].

5.1 Lee Channels

We start by introducing two channel models in the Lee metric. Both channel models are additive channels over the integer residue ring $\mathbb{Z}/q\mathbb{Z}$ where q is a positive integer. That is, we send a message $x \in (\mathbb{Z}/q\mathbb{Z})^n$ and observe at channel output a vector $y \in (\mathbb{Z}/q\mathbb{Z})^n$ possibly different from x, i.e., for some $e \in (\mathbb{Z}/q\mathbb{Z})^n$ we have $y = x + e$. The vector e is called the *error vector*. There are different ways to introduce errors. Either each symbol x_i of x is transmitted one after the other without any influence of the previous transmissions. In this case, we call the channel *memoryless*. Or we transmit the message block-wise, or as a whole vector, where the error vector is drawn randomly and of a given weight.

We first consider a discrete memoryless channel which we refer to the memoryless Lee channel. Recall from Definition 2.3.1 that channel is called *discrete memoryless*, if the input and output alphabet are discrete, finite sets and the output $Y = y$ at time t only depends on the input $X = x$ at that time t, i.e.,

$$\mathbb{P}(Y_1 = y_1, \ldots, Y_n = y_n \mid X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} \mathbb{P}(Y_i = y_i \mid X_i = x_i).$$

We define the channel law in such a way, that it matches to the Lee metric under maximum likelihood decoding introduced by [91].

Since the Lee metric has interesting applications to code-based cryptography, where errors are introduced on purpose and of a given Lee weight, in a second step we introduce a non-memoryless variant of the memoryless Lee channel which we refer to as the constant Lee-weight channel.

5.1.1 Memoryless Lee Channels

In the following we consider an additive discrete memoryless channel with input and output alphabet $\mathbb{Z}/q\mathbb{Z}$

$$y = x + e,$$

where y is the channel output, x is the channel input and e the additive error term. We will restrict to the case, where the additive error term $e \in \mathbb{Z}/q\mathbb{Z}$ is a realization of a random variable E whose distribution is proportional to an exponential function decreasing in the Lee weight of the error, i.e.,

$$P_E(e) \propto \exp(-\beta \, \mathrm{wt_L}(e)),$$

where $\beta > 0$ is a constant defining the channel. For the channel law that yields

$$P_{Y \mid X}(y \mid x) = \frac{1}{Z} \exp\left(-\beta \, \mathrm{d_L}(x, y)\right), \tag{5.1}$$

where $Z := \sum_{e=0}^{q-1} \exp(-\beta \, \mathrm{wt_L}(e))$ is the normalization constant. The probability defined in (5.1) satisfies the definition of a Lee channel given in (5.2). Therefore, from now on we refer to the Lee channel, the channel defined by (5.1).

Remark 5.1.1. The conditional probability satisfies the properties of a channel "matched to the Lee metric" under maximum likelihood decoding, introduced in [42, 91]. This is the channel whose maximum likelihood decoding rule reduces to finding the channel input x that minimizes the Lee distance from the channel output y;

The expected Lee weight of the additive error term of the channel (5.1) is given by

$$\delta := \mathbb{E}(\mathrm{wt_L}(E)) = -\frac{\mathrm{d} \log Z(\beta)}{\mathrm{d}\, \beta}$$

In the following subsection we consider a similar channel model. The main difference is that the channel introduce an error term of a fixed constant Lee weights.

Massey was the first one to introduce the notion of channels matching to a given metric. He defined it in the following way:

Definition 5.1.2. A metric and a discrete, memoryless channel are said to be matched for maximum likelihood decoding (MLD) if the decoding rule "decode the received vector to the nearest (or farthest) codeword" always gives a most probable codeword. More precisely, for two error vectors e and e' it must hold that

$$\mathrm{wt}_{\mathsf{L}}(e) < \mathrm{wt}_{\mathsf{L}}(e') \text{ if and only if } \mathbb{P}(e) > \mathbb{P}(e').$$

Define the "Lee Channel" over $\mathbb{Z}/q\mathbb{Z}$ as proposed by [42, Figure 2]:

$$p_i = \mathbb{P}(i \mid 0) = \mathbb{P}(-i \mid 0), \text{ for } i = 0, \ldots, \lfloor q/2 \rfloor . \tag{5.2}$$

Note, due to symmetry, we have $\mathbb{P}(i \mid j) = \mathbb{P}(i - j \mid 0)$, where $i - j$ is computed modulo q. Chiang and Wolf in [42] proved, that the channel described above strictly matches to the Lee metric for maximum likelihood decoding under some assumptions on the probabilities.

Theorem 5.1.3. [42, Theorem 1] *The channel described in (5.2) is strictly matched to the Lee metric for maximum likelihood decoding if and only if the following two properties hold.*

$$p_0 > p_1 \quad \text{and} \quad p_i = \frac{p_1^i}{p_0^{i-1}} \text{ for all } i = 2, \ldots, \lfloor q/2 \rfloor .$$

It follows that the channel distribution defined in (5.1), for $\beta > 0$ satisfies the conditions in Theorem 5.1.3 and hence, the memoryless Lee channel matches to the Lee metric under maximum likelihood decoding according to the definition introduced in [91].

5.1.2 Constant-Weight Lee Channel

We start by introducing an additive channel model over $\mathbb{Z}/q\mathbb{Z}$, that adds to a given codeword an error vector of a given fixed Lee weight. That is, given two positive integers $t, n \in \mathbb{Z}$ the channel output $y \in (\mathbb{Z}/q\mathbb{Z})^n$ is composed by a message $x \in (\mathbb{Z}/q\mathbb{Z})^n$ and an error vector $e \in (\mathbb{Z}/q\mathbb{Z})^n$ of Lee weight t chosen uniformly at random in $\mathcal{S}_{t,q}^{(n)}$, i.e.,

$$y = x + e \in (\mathbb{Z}/q\mathbb{Z})^n.$$

Hence, the channel transition probability for the constant Lee-weight channel is

$$P_{Y \mid X}(y \mid x) = \begin{cases} \left| \mathcal{S}_{t,q}^{(n)} \right|^{-1} & \text{if } \mathrm{d}_{\mathsf{L}}(y, x) = t, \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

The motivation for this model comes from cryptographic applications and more explicitly from the syndrome decoding problem (see Problem 1.0.1) that underlies most code-based cryptosystems. There, errors are introduced intentionally and of a given weight. The hardness of the problem, of course, relies on the weight of the error vector added to the codeword. Hence, the constant Lee-weight channel mimics such a scenario, where an error vector of given weight t is added to the codeword. Additionally, the error is drawn uniformly at random from the sphere of radius t in order not to reveal the structure, or the empirical distribution, of the error vector.

In this regard, for a vector $x \in (\mathbb{Z}/q\mathbb{Z})^n$ let $f(x) = (f_0(x), \ldots, f_{q-1}(x))$ denote the empirical distribution, meaning that

$$f_i(x) := \frac{1}{n} \left| \{ j \in \{1, \ldots, n\} \mid x_j = i \} \right| .$$

We call f the *type* (see Section 2.2.1 for more details on types) of the vector x. For a given composition φ, the set of vectors in $(\mathbb{Z}/q\mathbb{Z})^n$ with type φ is defined as

$$\mathcal{T}_{\varphi}^{(n)} := \{ x \in (\mathbb{Z}/q\mathbb{Z})^n \mid f(x) = \varphi \} .$$

By [45, Chapter 11.1], we observe that the cardinality of this set is exponentially equivalent to

$$\left|\mathcal{T}_{\varphi}^{(n)}\right| \doteq \exp(nH_{\mathrm{e}}(\varphi)),$$

which means that $\lim_{n\longrightarrow\infty} \frac{1}{n} \log\left(\frac{|\mathcal{T}_{\varphi}^{(n)}|}{\exp(nH_{\mathrm{e}}(\varphi))}\right) = 0$.

Marginal Channel Distribution

Recall from Lemma 3.2.3, when drawing an element a uniformly at random from an integer residue ring $\mathbb{Z}/q\mathbb{Z}$, its expected Lee weight is determined by δ_q and each value of $\mathbb{Z}/q\mathbb{Z}$ is equally likely to be drawn. It follows from the construction of a random vector in $\mathcal{S}_{t,q}^{(n)}$ that some underlying error partitions are more likely to occur than others.

We are interested in the marginal distribution of the channel law $P_{Y\,|\,X}(y \mid x)$ defined in (5.3) in the limit of $n \to \infty$. The marginal distribution plays an important role, for instance, in the initialization of iterative decoders of LDPC codes, when used over a constant Lee-weight channel [114]. While the focus here is in the asymptotic (in the block length n) case, the derived marginal distribution provides an excellent approximation of the true marginal down to moderate-length blocks (n in the order of a few hundreds).

In the following, we consider the normalized Lee weight $\delta := t/n$ of a vector in $\mathcal{S}_{t,q}^{(n)}$. The derivation follows by seeking the composition that dominates the set $\mathcal{S}_{\delta n,q}^{(n)}$. More specifically, we should look for the empirical distribution φ that maximizes the cardinality of $\mathcal{T}_{\varphi}^{(n)}$ under the constraint

$$\sum_{i=0}^{q-1} \mathrm{wt_L}(i)\varphi_i = \delta. \tag{5.4}$$

Lemma 5.1.4. *Assume that $x \in (\mathbb{Z}/q\mathbb{Z})^n$ has been drawn uniformly at random among all vectors of Lee weight t. Let X denote the random variable defining the realizations of an entry of x. As n grows large, for every $i \in \mathbb{Z}/q\mathbb{Z}$, the probability of X taking the value i is given by*

$$P_i^{\star} := \mathbb{P}(X = i) = \frac{1}{Z(\beta)} \exp\left(-\beta\, \mathrm{wt_L}(i)\right), \tag{5.5}$$

where β is the unique real solution to the weight constraint given in (5.4) and $Z(\beta)$ denotes the normalization constant.

Proof. Following [45, Chapter 12], we are looking for a distribution $P = (P_0, \ldots, P_{q-1})$ that maximizes the entropy function

$$H_{\mathrm{e}}(P) := -\sum_{\substack{i=0 \\ P_i \neq 0}}^{q-1} P_i \log P_i$$

under the constraint that the Lee weight of the vector is t, or equivalently, that the normalized Lee weight of the error vector is $\delta := t/n$, i.e.

$$\sum_{e=0}^{q-1} \mathrm{wt_L}(e) P_i = \delta.$$

Let us introduce a Lagrange multiplier $\beta > 0$, which is the solution to

$$\delta = \frac{(k-1)\mathrm{e}^{(k+1)\beta} - k\mathrm{e}^{k\beta} + \mathrm{e}^{\beta}}{(\mathrm{e}^{\beta k} - 1)(\mathrm{e}^{\beta} - 1)},$$

with $k = \lfloor q/2 \rfloor + 1$. Then the optimization problem has the following solution

$$P_i^{\star} = \kappa \exp\left(-\beta\, \mathrm{wt_L}(e)\right), \tag{5.6}$$

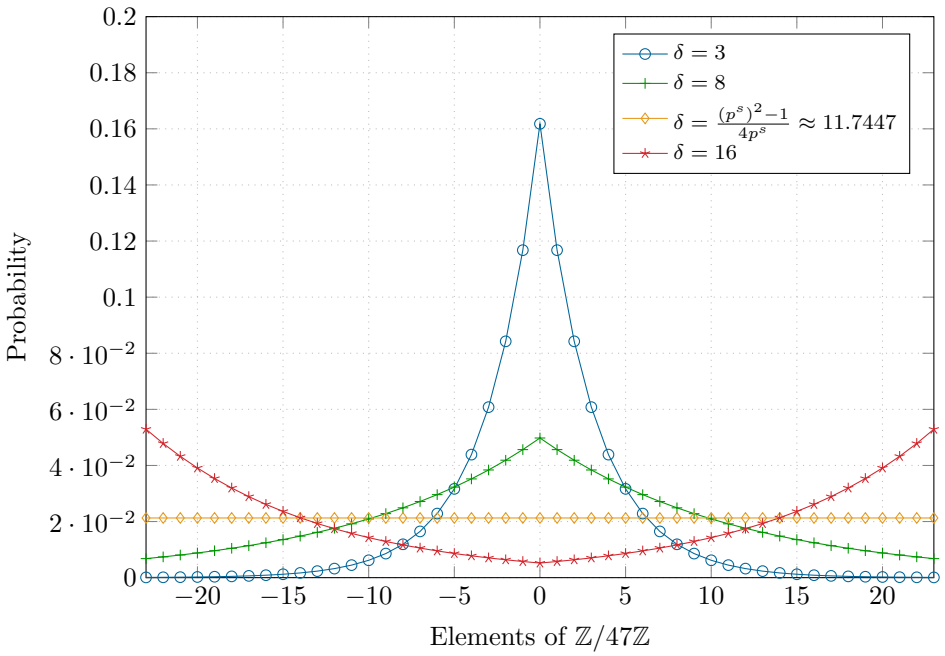

FIGURE 5.1: Marginal distribution for the elements in $\mathbb{Z}/47\mathbb{Z}$ for different values of $\delta = \lim_{n\to\infty} t(n)/n$.

where κ is a normalization constant enforcing $\sum_i P_i^\star = 1$. □

Note that if $\delta = \delta_q$, then X is distributed uniformly over $\mathbb{Z}/q\mathbb{Z}$ and hence $\beta = 0$. Moreover, $\beta > 0$ if and only if $\delta < \delta_q$. Furthermore, if $\beta > 0$ the relative weight becomes smaller. Since the marginal distribution (5.5) is an exponential function with negative exponent, it is decreasing in the weight. This means that for $\beta > 0$ the elements of the smallest Lee weight, i.e., 0, are the most probable, then elements of weight 1 until the least probable Lee weight $\lfloor q/2 \rfloor$. In the other case, where $\beta < 0$, the elements of the largest Lee weight, i.e., $\lfloor q/2 \rfloor$, are the most probable, followed by the elements of weight $\lfloor q/2 \rfloor - 1$, and so on, until the least probable of Lee weight 0. Figure 5.1 shows this behaviour over $\mathbb{Z}/47\mathbb{Z}$.

The distribution in (5.5) is closely related to the Boltzmann distribution [45, 30]. The Boltzmann distribution gives the probability that a system will be in a certain state depending on that states' energy and temperature. In statistical mechanics the distribution is used for systems of fixed compositions all being in a thermal equilibrium. Additionally, the distribution maximizes the entropy subject to a mean energy state. In our case the Lee weight may be interpreted as the energy value of a state $e \in \mathbb{Z}/q\mathbb{Z}$. Hence, we will refer to the distribution in (5.5) as *Boltzmann distribution*, and we will denote it by B_δ. Note also that for the channel law determined by Lemma 5.1.4, the optimal decoder will seek for the codeword at minimum Lee distance from the channel output y.

As a direct consequence of Lemma 5.1.4, we can give the probability of a random entry E having some given Lee weight

$$\mathbb{P}(\mathrm{wt_L}(X) = j) = \begin{cases} \mathbb{P}(X = j) & \text{if } (j = 0) \text{ or } (j = \lfloor q/2 \rfloor \text{ and } q \text{ is even}), \\ 2\mathbb{P}(X = j) & \text{else.} \end{cases} \tag{5.7}$$

Note that the constant Lee weight t grows linearly with n. Hence, in the following instead of saying that $x \in (\mathbb{Z}/q\mathbb{Z})^n$ has Lee weight t we will always relate to the average Lee weight $\delta = t/n$ of the entries of x. Analogously, instead of $\mathcal{S}_{t,q}^{(n)}$, let us consider the set E of probability distributions of vectors with an average Lee weight δ over $\mathbb{Z}/q\mathbb{Z}$, i.e.

$$E_{\delta,q} := \left\{ P = (P_0, \ldots, P_{q-1}) \,\middle|\, \sum_{i=0}^{q-1} P_i = 1 \text{ and } \sum_{i=0}^{q-1} P_i \,\mathrm{wt_L}(i) = \delta \right\}.$$

Lemma 5.1.5. *The set E is convex.*

Proof. By definition, $E_{\delta,q}$ is convex, if for every $P_1, P_2 \in E_{\delta,q}$ and every $\lambda \in (0,1)$ we have that $\lambda P_1 + (1-\lambda)P_2 \in E_{\delta,q}$. Take two arbitrary distributions P and Q of $E_{\delta,q}$ and an arbitrary $\lambda \in (0,1)$. Let $V := \lambda P + (1-\lambda)Q$. First, V is a probability distribution, since

$$\sum_{i=0}^{q-1} V_i = \sum_{i=0}^{q-1} P_i + \lambda(P_i - Q_i) = \sum_{i=0}^{q-1} P_i + \lambda \left(\sum_{i=0}^{q-1} P_i - \sum_{i=0}^{q-1} Q_i \right) = 1.$$

Similarly, we have

$$\sum_{i=0}^{q-1} V_i \operatorname{wt}_{\mathsf{L}}(i) = \sum_{i=0}^{q-1} P_i \operatorname{wt}_{\mathsf{L}}(i) + \lambda \left(\sum_{i=0}^{q-1} P_i \operatorname{wt}_{\mathsf{L}}(i) - \sum_{i=0}^{q-1} Q_i \operatorname{wt}_{\mathsf{L}}(i) \right) = \delta.$$

□

Hence, a straightforward application of the conditional limit theorem, Theorem 2.2.9, yields the following corollary.

Corollary 5.1.6. *Let x be a random vector over $\mathbb{Z}/q\mathbb{Z}$ of average Lee weight δ and with entries being i.i.d. distributed according to Q_x and let $P^\star$ denote the distribution given in Lemma 5.1.4. Then, for every $\varepsilon > 0$ it holds*

$$\mathbb{P}\left(D(Q_x \,||\, P^\star) \geq \varepsilon\right) \longrightarrow 0 \text{ as } n \longrightarrow \infty.$$

Proof. Let $x = (x_1, \ldots, x_n) \in (\mathbb{Z}/q\mathbb{Z})^n$ be a random vector whose entries are independent and uniformly distributed in $\mathbb{Z}/q\mathbb{Z}$. The distribution of x is uniform on $(\mathbb{Z}/q\mathbb{Z})^n$ (denoted by $U(\mathbb{Z}/q\mathbb{Z})$), and hence on $\mathcal{S}_{t,q}^{(n)}$. We have that

$$P^\star = \arg \min_{P \in E_{\delta,q}} D(P \,||\, U(\mathbb{Z}/q\mathbb{Z})).$$

Then, by Theorem 2.2.9, we obtain the desired result. □

5.2 Finite-Length Bounds for Lee Channels

In this section we are going to derive bounds on the error probability achievable by a code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ over both the constant Lee-weight channel and the memoryless Lee channel defined in Section 5.1.1. Several of the bounds on the minimum distance achievable by linear codes over $\mathbb{Z}/q\mathbb{Z}$ can be found in [10, 34, 86]. In the first case we will see an attainability bound in terms of a random coding union bound. For the memoryless Lee channel we will derive an upper bound again in terms of a random coding union bound as well as a converse bound, meaning a lower bound, achievable by any $[n,k]$ code in terms of a sphere-packing bound.

For both channel models we distinguish between maximum likelihood decoding and minimum distance decoding, that is, given a received word $y \in \mathbb{Z}/q\mathbb{Z}$, we consider the maximum likelihood decoding rule

$$\hat{x}_{ML} = \operatorname*{argmax}_{x \in \mathcal{C}} P_{Y \mid X}(y \mid x)$$

and the minimum distance decoding rule

$$\hat{x}_{\mathrm{MD}} = \operatorname*{argmin}_{x \in \mathcal{C}} \mathrm{d}_{\mathsf{L}}(y, x).$$

Note that the two decoding rules coincide over the memoryless Lee channel for $\delta \leq \delta_q$. In the constant Lee-weight channel, the maximum likelihood decoder gives a list of all codewords which are at distance δn from the received word y, and it outputs one of the codewords in this list randomly. Hence, the two decoding rules coincide for the constant Lee-weight channel whenever δn is within the decoding radius of the code $\mathcal{C}$.

5.2.1 Bounds on the Lee Spheres and Lee Balls

Before proceeding with the derivation of the error probability bounds, we first derive upper bounds on the size of a Lee-sphere and a Lee-ball, respectively. Recall, that spheres and balls in the Lee metric were introduced in Section 3.3 in terms of generating functions, and we will adapt the notation provided there. The knowledge of the size of an n-dimensional sphere of Lee-radius t, in coding theory, can be seen as the number of codewords of length n of Lee weight t and is crucial for bounds providing information on the possible code rates given the code's minimum distance. One of the most known results in the Hamming metric is that the size of the n-dimensional sphere of radius t over a q-ary alphabet is bounded by $q^{nH_q(t/n)}$, where H_q is the q-ary entropy function. In the Lee metric, similar arguments can be used to derive bounds using the entropy.

In fact, let $\mathcal{F}_\delta^{(n)}$ denote the set of empirical distributions of the sequences in $\mathbb{Z}/q\mathbb{Z}^n$ with normalized Lee weight δ, then the size of the n-dimensional Lee-sphere of normalized radius δ is given by

$$\sum_{f\in\mathcal{F}_\delta^{(n)}} \binom{n}{nf_0,\ldots,nf_{q-1}}.$$

Individuating the distribution $f^\star \in \mathcal{F}_\delta^{(n)}$ that maximizes the multinomial coefficient, the following two bounds on the sphere size follow immediately:

$$\sup_{f\in\mathcal{F}_\delta^{(n)}} \binom{n}{nf_0,\ldots,nf_{q-1}} \leq \left|\mathcal{S}_{\delta n,q}^{(n)}\right| < n^q \sup_{f\in\mathcal{F}_\delta^{(n)}} \binom{n}{nf_0,\ldots,nf_{q-1}}$$

The relation to a bound involving the entropy becomes clear, when recalling that [46, Lemma 2.1]

$$\binom{n}{nf_0,\ldots,nf_{q-1}} \doteq 2^{nH(f)}.$$

In [86], Löliger gave an asymptotically tight upper bound on the size of an n-dimensional ball of radius t given any additive-Lee weight (such as the Lee weight, the Hamming weight or even the sum-rank weight) based on the distribution given in (5.6). In the Lee metric, this bound is tight for a normalized weight $\delta \in [0, \delta_q]$. Note that for $\delta > \delta_q$ the size of the ball is saturated. Additionally, since the size of an n-dimensional sphere of radius t is naturally upper bounded by the size of the n-dimensional ball of the same radius, Löliger's bound also holds for the size of the sphere. Hence, in the following some results are known by [86]. However, we prove alternative proofs involving the entropy function of the Boltzmann-like distribution B_δ. The relation of the entropy function and the volume and surface of a sphere has been pointed out by Shannon in his seminal work [117].

We denote by $H_\delta := H(B_\delta)$ the entropy of the Boltzmann distribution with parameter δ, and we introduce the notation

$$H_\delta^+ := \begin{cases} H_\delta & 0 \leq \delta \leq \delta_q \\ \log_2(q) & \delta_q < \delta < r. \end{cases}$$

Lemma 5.2.1 (Growth rate of the surface spectrum)**.** *For any positive integer δn the surface spectrum is upper bounded by*

$$\sigma_{\delta n}^{(n)} \leq H_\delta.$$

In particular, as n grows large it holds that $\sigma_\delta = H_\delta$.

Proof. Let $X = (X_1, \ldots, X_n)$ be a finite sequence of random variables X_i chosen uniformly at random in the Lee-sphere $\mathcal{S}_{\delta n,q}^{(n)}$. Since X is uniformly distributed in the sphere, its entropy

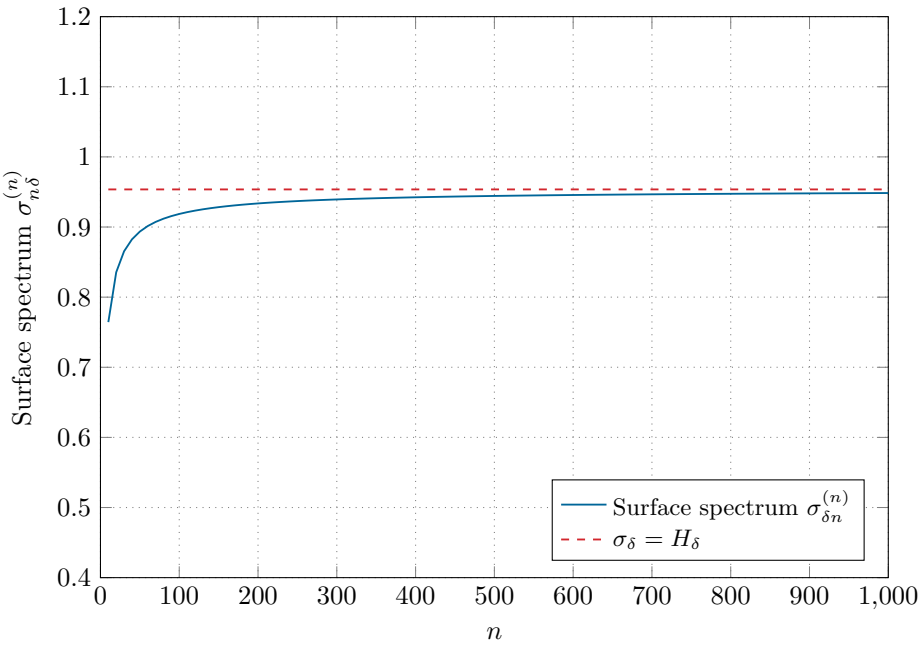

FIGURE 5.2: Convergence of $\sigma_{\delta n}^{(n)}$ to $\sigma_\delta = H_\delta$ as a function of n over $\mathbb{Z}/7\mathbb{Z}$ with $\delta = 0.2$.

is given by $H(X) = \log_2\left(\left|\mathcal{S}_{\delta n,q}^{(n)}\right|\right)$. Hence, the normalized logarithmic surface area is

$$\sigma_{\delta n}^{(n)} = \frac{1}{n} H(X).$$

The chain rule for the entropy (see Theorem 2.1.4) and the fact that the X_i´s are identically distributed, yield

$$H(X) \leq \sum_{i=1}^{n} H(X_i) = nH(X_1).$$

Since the Boltzmann distribution B_δ is the distribution of X_1 maximizing the entropy under the constraint that $\mathbb{E}(\mathrm{wt_L}(X_1)) = \delta$, the desired upper bound follows. To get the asymptotic result it suffices to take limits on both sides of the inequality. □

Figure 5.2 shows that the asymptotic limit is tightly approached already for n in the order of a few hundreds.

Lemma 5.2.2 (Growth Rate of the Volume Spectrum)**.** *For any positive integer δn the volume spectrum is upper bounded by*

$$\nu_{\delta n}^{(n)} \leq H_\delta^+.$$

In particular, as n grows large we have that $\nu_\delta = H_\delta^+$.

Proof. The proof follows similarly to the proof of the growth rate of the surface spectrum. Consider a random vector $X = (X_1, \ldots, X_n)$ chosen uniformly at random over $\mathcal{B}_{\delta n,q}^{(n)}$. Hence, $\mathrm{wt_L}(x) \leq \delta n$, where x denotes the realization of X. It holds that

$$\log_2\left(\left|\mathcal{B}_{\delta n,q}^{(n)}\right|\right) = H(X),$$

which implies, using again Theorem 2.1.4, that

$$\nu_{\delta n}^{(n)} = \frac{1}{n} H(X) \leq H(X_1).$$

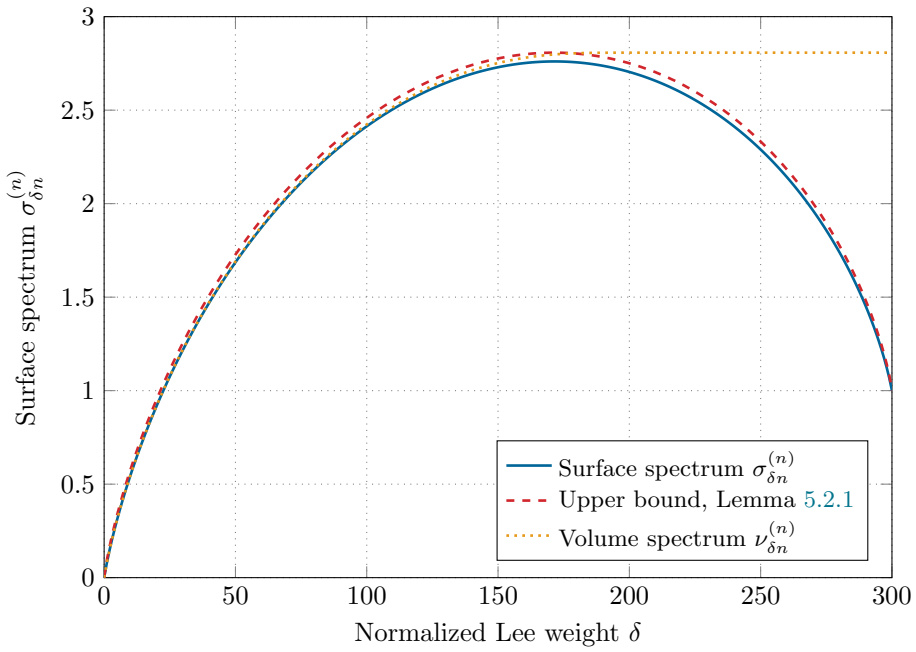

FIGURE 5.3: Comparison of exact surface spectrum $\sigma_{\delta n}^{(n)}$ and the volume spectrum $\nu_{\delta n}^{(n)}$ with the upper bound derived in Lemma 5.2.1 for $n = 100$ over $\mathbb{Z}/7\mathbb{Z}$.

Note that $H(X_1) \leq \log_2(q)$ for any parameter of $\delta \in [0, r]$. Hence, again since B_δ maximizes the entropy under the constraint $\mathbb{E}(\mathrm{wt}_{\mathrm{L}}(X_1)) \leq \delta$, we observe that $H(X_1) \leq H_\delta^+$ which yields the first statement of the lemma. To prove the latter statement it suffices to take the limit as n tends to infinity. □

Figure 5.3 shows the surface spectrum $\sigma_{\delta n}^{(n)}$ for $n = 100$ and $\mathbb{Z}/7\mathbb{Z}$. Observe that the sphere surface gets larger as the radius grows, till it reaches a maximum, and then it decreases. The upper bound from Lemma 5.2.1 is also provided. Note that the upper bound is tight, and it reaches a maximum at $H_\delta = \log_2(q)$ when $\delta = \delta^\star$. It is easy to check that in this case $\beta = 0$. On the same figure, we report the volume spectrum

$$\nu_{\delta n}^{(n)} = \frac{1}{n} \log_2 \left(\sum_{i=0}^{\delta} 2^{n\sigma_{in}^{(n)}} \right).$$

Note that the expression above can be efficiently computed via recursive application of the Jacobian logarithm.

5.2.2 Error Probability Bounds for the Constant Lee-Weight Channel

We consider a linear code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ of cardinality $|\mathcal{C}| = q^k =: M$, and we focus on the constant Lee-weight channel where the additive error term is of fixed Lee weight δn.

Theorem 5.2.3 (Random Coding Union Bound, ML Decoding)**.** *Let $\mathcal{C} \subset (\mathbb{Z}/q\mathbb{Z})^n$ be a random code of rate R_2. The average maximum likelihood decoding error probability of $\mathcal{C}$ used to transmit over a constant Lee-weight channel satisfies*

$$\mathbb{E}\left(P_B(\mathcal{C})\right) < 2^{-n\left[\log_2(q) - \sigma_{\delta n}^{(n)} - R_2\right]^+}.$$

Proof. Consider first the pairwise error probability $\mathsf{PEP}(x, y)$ for fixed x and y, where x is the transmitted codeword, y is the channel output and $\widetilde{X}$ is a random codeword distributed uniformly over $(\mathbb{Z}/q\mathbb{Z})^n$. By breaking ties always towards $\widetilde{X}$, we can upper bound the pairwise

error probability as

$$\begin{aligned}\mathsf{PEP}(x,y) &\leq \mathbb{P}\left(P_{Y\mid X}(y\mid x) = P_{Y\mid X}(y\mid \widetilde{X})\right)\\ &= \mathbb{P}\left(\mathrm{d_L}(y,\widetilde{X}) = \delta n\right)\\ &= \frac{\left|\mathcal{S}^{(n)}_{\delta n,q}\right|}{q^n}.\end{aligned}$$

The union bound on the block error probability is obtained by multiplying the result by $M-1$. By observing that the pairwise error probability does not depend on x, y, we get

$$\begin{aligned}\mathbb{E}\left(P_B(\mathcal{C})\right) &\leq \min\left(1, (M-1)\mathsf{PEP}(x,y)\right)\\ &< \min\left(1,\ M\frac{\left|\mathcal{S}^{(n)}_{\delta n,q}\right|}{q^n}\right)\\ &= 2^{-n\left[\log_2(q)-\sigma^{(n)}_{\delta n}-R_2\right]^+}.\end{aligned}$$

□

Owing to Lemma 5.2.1, the bound can be loosened yielding the simple form described in the following corollary.

Corollary 5.2.4. *The average maximum likelihood decoding error probability of a random code $\mathcal{C} \subset (\mathbb{Z}/q\mathbb{Z})^n$ of rate R_2 used to transmit over a constant Lee-weight channel satisfies*

$$\begin{aligned}\mathbb{E}(P_B(\mathcal{C})) &< 2^{-n[\log_2(q)-H_\delta-R_2]^+}\\ &= 2^{-n[KLB_q\mathcal{U}(\mathbb{Z}/q\mathbb{Z})-R_2]^+}.\end{aligned}$$

In terms of minimum distance decoding, the two results can be proven in a similar fashion, considering all codewords of distance up to δn, i.e., instead of working over the sphere $\mathcal{S}^{(n)}_{\delta n,q}$ we only extend to the ball $\mathcal{B}^{(n)}_{\delta n,q}$. Then the minimum distance counterparts of Theorem 5.2.3 and its consequence, Corollary 5.2.4, are given in the following two results.

Theorem 5.2.5 (Random Coding Union Bound, MD decoding)**.** *Let $\mathcal{C} \subset (\mathbb{Z}/q\mathbb{Z})^n$ be a random code of rate R_2. The average minimum distance decoding error probability of $\mathcal{C}$ used to transmit over a constant Lee-weight channel satisfies*

$$\mathbb{E}(P_B(\mathcal{C})) < 2^{-n\left[\log_2(q)-\nu^{(n)}_{\delta n}-R_2\right]^+}.$$

Proof. Consider first the pairwise error probability under the assumption that x is the transmitted codeword, y is the channel output and $\widetilde{X}$ is a random codeword distributed uniformly over $(\mathbb{Z}/q\mathbb{Z})^n$. By breaking ties always towards $\widetilde{X}$, we have

$$\begin{aligned}\mathsf{PEP}(x,y) &\leq \mathbb{P}\left(\mathrm{d_L}(y,x) \geq \mathrm{d_L}(y,\widetilde{X})\right)\\ &= \mathbb{P}(\mathrm{d_L}(y,\widetilde{X}) \leq \delta n)\\ &= \frac{\left|\mathcal{B}^{(n)}_{\delta n,q}\right|}{q^n}.\end{aligned}$$

The union bound on the block error probability can be obtained by multiplying the result by $M-1$. By observing that the pairwise error probability does not depend on x, y, we get

$$\begin{aligned}\mathbb{E}(P_B(\mathcal{C})) &\leq \min\left(1, (M-1)\mathsf{PEP}\right)\\ &< \min\left(1, M\frac{\left|\mathcal{B}_{\delta n,q}^{(n)}\right|}{q^n}\right)\\ &= 2^{-n\left[\log_2(q)-\nu_{\delta n}^{(n)}-R_2\right]^+}.\end{aligned}$$

□

Owing to Lemma 5.2.2, the bound can be loosened yielding the simple form described in Corollary 5.2.6.

Corollary 5.2.6. *The average minimum distance decoding error probability of a random code $\mathcal{C} \subset (\mathbb{Z}/q\mathbb{Z})^n$ of rate R_2 used to transmit over a constant Lee-weight channel satisfies*

$$\mathbb{E}(P_B(\mathcal{C})) < 2^{-n\left[\log_2(q)-H_\delta^+-R_2\right]^+}.$$

Figure 5.4 depicts the upper bounds given in Theorem 5.2.5 and Corollary 5.2.6 for minimum distance decoding, for $[500, 250]$ codes over $\mathbb{Z}/7\mathbb{Z}$. The bound of Corollary 5.2.6 is only slightly looser than the one provided by Theorem 5.2.5. A similar result holds for the bounds of Theorem 5.2.3 and Corollary 5.2.4, under maximum likelihood decoding.

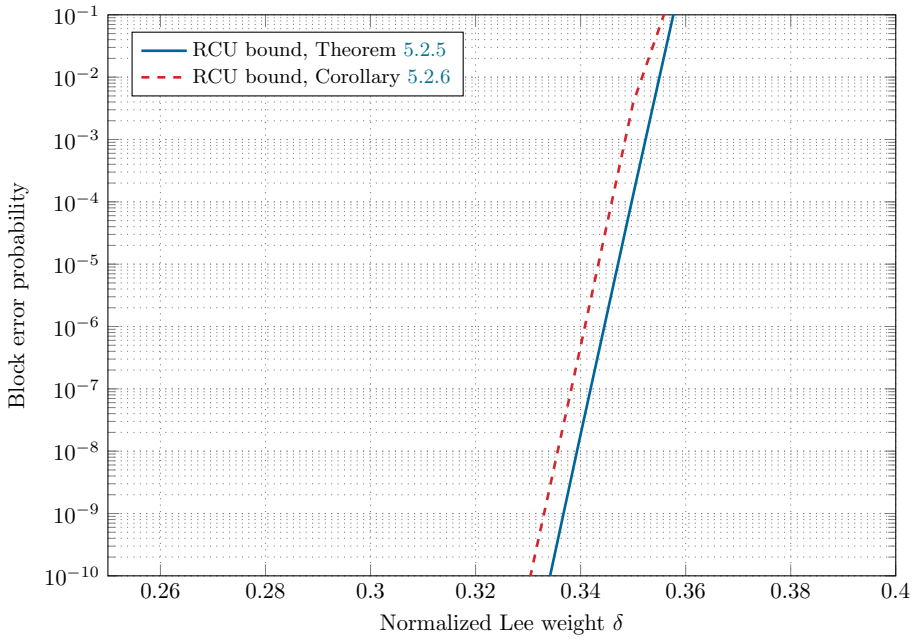

FIGURE 5.4: Random coding union bounds under minimum distance decoding based on Theorem 5.2.5 and Corollary 5.2.6 for the parameters $n = 500$ and $k = 250$ over $\mathbb{Z}/7\mathbb{Z}$.

5.2.3 Error Probability Bounds for the Memoryless Lee Channel

We consider next a memoryless Lee channel with expected normalized Lee weight of the error pattern δ. We restrict the attention to the case $\delta \leq \delta_q$. In this regime, the maximum likelihood and the minimum distance decoding rules coincide.

Theorem 5.2.7 (Random Coding Union Bound)**.** *Let $\mathcal{C} \subset (\mathbb{Z}/q\mathbb{Z})^n$ be a random code of rate R_2. The average maximum likelihood/minimum distance decoding error probability of $\mathcal{C}$ used to transmit over a memoryless Lee channel with expected normalized Lee weight of the error*

pattern δ satisfies

$$\mathbb{E}\left(P_B(\mathcal{C})\right) < \mathbb{E}\left(2^{-n\left[\log_2(q)-\nu_L^{(n)}-R_2\right]^+}\right),$$

where the expectation is taken over the distribution of the Lee weight $L = \mathrm{wt}_{\mathsf{L}}(E)$.

A direct consequence using Lemma 5.2.2 is captured in Corollary 5.2.8. Its proof follows similar to the constant Lee-weight case.

Corollary 5.2.8. *The average maximum likelihood/minimum distance decoding error probability of a random code $\mathcal{C} \subset (\mathbb{Z}/q\mathbb{Z})^n$ of rate R_2 used to transmit over a memoryless Lee channel satisfies*

$$\mathbb{E}(P_B(\mathcal{C})) < \mathbb{E}\left(2^{-n\left[\log_2(q)-H_{L/n}^{+}-R_2\right]^+}\right),$$

where the expectation is taken over the distribution of the Lee weight $L = \mathrm{wt}_{\mathsf{L}}(E)$.

Following the idea of [59, Section 5.8], we provide now a lower bound on the block error probability achievable by *any* $[n, k]$ code over the memoryless Lee channel.

Theorem 5.2.9 (Sphere Packing Bound)**.** *The block error probability of any code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ of rate R_2 over a memoryless Lee channel is lower bounded as*

$$P_B(\mathcal{C}) > \frac{1}{Z(\beta)^n} \sum_{d=d_0+1}^{rn} \left|\mathcal{S}_{d,q}^{(n)}\right| \exp\left(-\beta d\right) + \frac{1}{Z(\beta)^n}\left(\left|\mathcal{S}_{d_0,q}^{(n)}\right| - \xi\right)\exp\left(-\beta d_0\right),$$

where d_0 and ξ are chosen so that

$$\sum_{d=0}^{d_0-1}\left|\mathcal{S}_{d,q}^{(n)}\right| + \xi = 2^{n(\log_2(q)-R_2)} \qquad \text{and} \qquad 0 < \xi \leq \left|\mathcal{S}_{d_0,q}^{(n)}\right|.$$

Proof. The proof follows closely the analogous proof for the binary symmetric channel provided in [59, Section 5.8]. □

Figure 5.5 depicts the random coding union bound of Corollary 5.2.8 and the sphere-packing bound of Theorem 5.2.9, over a memoryless Lee channel, for $[1024, 512]$ codes over $\mathbb{Z}/7\mathbb{Z}$. The two bounds are close to each other. Hence, they provide an accurate benchmark to assess the performance achievable over the memoryless Lee channel.

Notice that the Boltzmann-like distribution given in (5.6) and its entropy can be used in the asymptotic version of the sphere-packing bound and the Gilbert-Varshamov bound which we state in Theorem 5.2.10 and Theorem 5.2.11, respectively. In [10] some of these bounds have already been stated using the entropy but without using the Boltzmann-like distribution.

Theorem 5.2.10 (Sphere-Packing Bound, asymptotic)**.** *Consider a code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ of minimum Lee distance $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) = d_{\min}$. In the limit of n, denote by $\delta_{\min} = d_{\min}/n$. We have*

$$R \leq 1 - H_{\delta}^{+}\log_q(2),$$

with $\delta = \delta_{\min}/2$.

Theorem 5.2.11 (Gilbert-Varshamov Bound, asymptotic)**.** *Consider a code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ of minimum Lee distance $\mathrm{d}_{\mathsf{L}}(\mathcal{C}) = d_{\min}$. In the limit of large n, denote by $\delta_{\min} = d_{\min}/n$. We have that the largest rate achievable satisfies*

$$R \geq 1 - H_{\delta}^{+}\log_q(2), \tag{5.8}$$

with $\delta = \delta_{\min}$.

Note that the expected Lee weight distribution for a Lee weight $d \in \mathbb{N}$ of a code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ is

$$\bar{A}_d = |\,\mathcal{C}\,|\,\frac{\left|\mathcal{S}_{d,q}^{(n)}\right|}{q^n}.$$

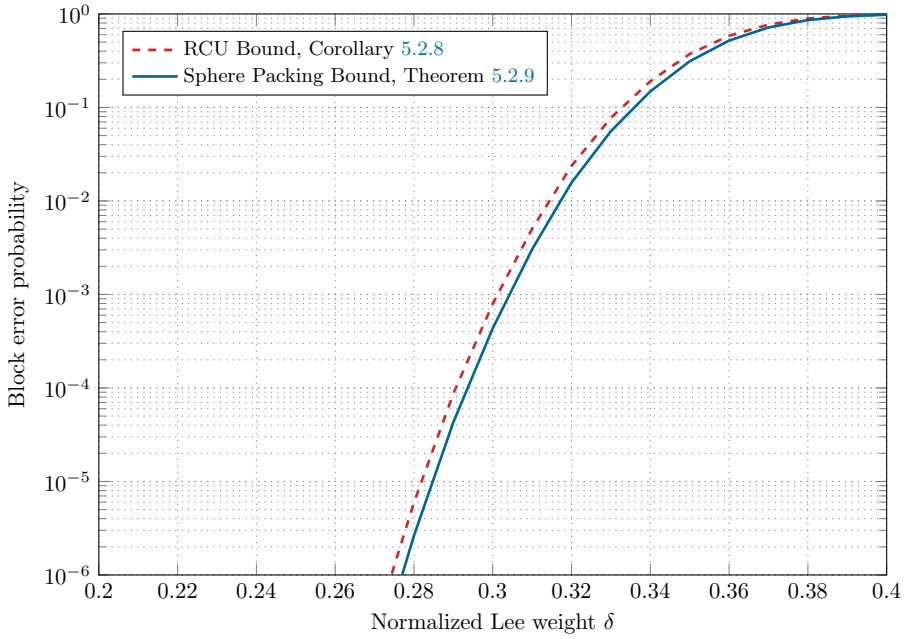

FIGURE 5.5: Random coding union (Corollary 5.2.8) and sphere-packing bounds (Theorem 5.2.9) for the parameters $n = 1024$ and $k = 512$ over $\mathbb{Z}/7\mathbb{Z}$.

The result of Theorem 5.2.11 can be recovered by showing that a random (n, M) code $\mathcal{C}$ possesses a minimum distance that is at least $n\delta_{\min}$ w.h.p. as n grows large, where $\delta_{\min}$ is the solution in δ of (5.8). In fact,

$$\mathbb{P}\left(\mathrm{d_L}(\mathcal{C}) \leq \delta n\right) \leq \sum_{i=0}^{\delta n} \bar{A}_i = |\,\mathcal{C}\,| \frac{\left|\mathcal{B}_{\delta n,q}^{(n)}\right|}{q^n} \leq 2^{-n\left[\log_2(q) - H_\delta^+ - R_2\right]},$$

where the inequality follows from Markov's inequality. Note that the exponent of the right-hand side is negative whenever

$$R_2 < \log_2(q) - H_\delta^+.$$

This means, for large n, we have that $\mathbb{P}\left(\mathrm{d_L}(\mathcal{C}) \leq \delta n\right)$ tends to zero, recovering the result of Theorem 5.2.11. Figure 5.6 shows the asymptotic sphere-packing bound and Gilbert-Varshamov bound in the Lee metric over the ring $\mathbb{Z}/7\mathbb{Z}$.

5.3 Fixed Lee Weight Vectors

In this section, we turn our focus to the application to code-based cryptography. As mentioned in Section 5.1.2, the motivation for the constant Lee-weight channel comes inter alia from code-based cryptography where an error vector of given weight is intentionally added to a codeword sent to obtain a received message. The task (either as a receiver or an attacker), when receiving this erroneous codeword, is to reconstruct either the original message or the error vector. This problem is known as the *syndrome decoding problem* (see Problem 1.0.1). From an adversarial point of view, the goal is to reduce the weight of the introduced error vector in order to reduce the complexity of solving the generic (syndrome) decoding problem. Thus, sampling the error vector uniformly at random among all vectors of the same Lee weight is crucial not to leak information. In Section 5.3.1, we present two algorithms that draw a vector of length n and fixed Lee weight t over the ring of integers $\mathbb{Z}/q\mathbb{Z}$ uniformly at random from the set of vectors with the same parameters. The first algorithm is based only on integer partitions whereas the second algorithm involves also the generating function of the sphere $\mathcal{S}_{t,q}^{(n)}$. Introducing errors uniformly at random is important from a cryptographic point of view in order to hide the structure of the error pattern.

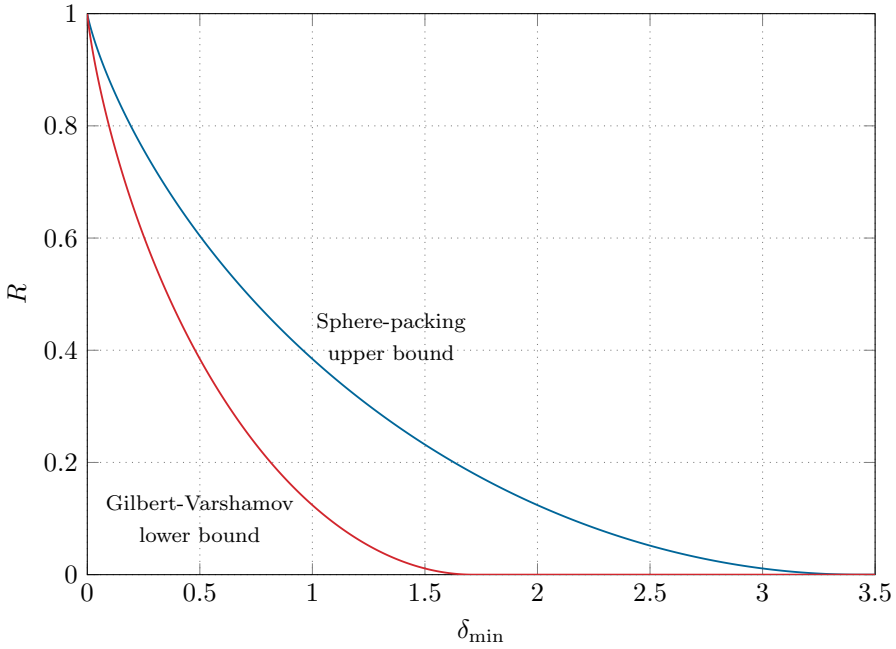

FIGURE 5.6: Asymptotic Lee sphere-packing and Gilbert-Varshamov bounds over $\mathbb{Z}/7\mathbb{Z}$.

While the Hamming weight of a vector with entries from a finite field is invariant under multiplication with a nonzero scalar, the Lee weight of a vector can be increased or decreased by the product with a scalar. Understanding under which conditions (and with what probability) the Lee weight on the error vector e is reduced represents a key preliminary step in the design of Lee-metric code-based cryptosystems. We will refer to this problem as *scalar multiplication problem*. The marginal distribution derived in Section 5.1.2 enables to analyze how the Lee weight of a given error vector changes when multiplied by a random nonzero scalar, in the asymptotic regime. We show in Section 5.3.2 that, under certain conditions, the Lee weight of such an error vector will not decrease after scalar multiplication with high probability.

5.3.1 Construction of Random Error Vectors

Consider the following task: Given two positive integers t and n, construct a vector $x \in (\mathbb{Z}/q\mathbb{Z})^n$ of Lee weight t such that it follows a uniform distribution over the n-dimensional Lee-sphere $\mathcal{S}_{t,q}^{(n)}$ of radius t.

Algorithm Based on Weight Partition

In the following we present an algorithm that draws a vector uniformly at random from $\mathcal{S}_{t,q}^{(n)}$ for given parameters n, t and q. The idea is inspired by the algorithm presented in [114]. We start from partitioning the desired Lee weight t into integer parts of size at most $\lfloor q/2 \rfloor$, since the maximum possible Lee weight is $\lfloor q/2 \rfloor$. Hence, let us formally introduce integer partitions.

Definition 5.3.1. Let t and s be positive integers. An *integer partition* of t into s parts is an s-tuple $\lambda := (\lambda_1, \ldots, \lambda_s)$ of positive integers satisfying the following two properties:

i. $\lambda_1 + \ldots + \lambda_s = t$,

ii. $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_s > 0$.

The elements λ_i are called *parts*.

Note that the order of the parts does not matter. This means that, for instance, the tuples $(1, 1, 2)$, $(1, 2, 1)$ and $(2, 1, 1)$ are all identical and represented only by $(2, 1, 1)$. We will denote

by Π_λ the set of all permutations of an integer partition λ. Let n_i denote the number of occurrences of a positive integer i in an integer partition λ of t, where $i \in \{1, \ldots, t\}$ then

$$|\,\Pi_\lambda\,| = \binom{t}{n_1, \ldots, n_t} = \frac{t!}{n_1! \ldots n_t!}.$$

In the following, we use $\mathcal{P}(t)$ to denote the set of integer partitions of t. We write $\mathcal{P}_k(t)$ instead, if we restrict $\mathcal{P}(t)$ to those partitions with part sizes not exceeding some fixed nonnegative integer value k. If we further restrict $\mathcal{P}_k(t)$ to partitions of a fixed length ℓ, we use the notation $\mathcal{P}_{k,\ell}(t)$. We denote by ℓ_λ the length of a given partition λ. Note that for any $\lambda \in \mathcal{P}_k(t)$ its length ℓ_λ is bounded by $\lceil \frac{t}{k} \rceil \leq \ell_\lambda \leq t$.

The main difference to the algorithm presented in [114, Lemmas 2 and 3], and crucial to design the vector uniformly at random from $\mathcal{S}_{t,q}^{(n)}$, is that the integer partition of t is not chosen uniformly at random from the set of all integer partitions $\mathcal{P}_{\lfloor q/2 \rfloor}(t)$ of t. In fact, picking a partition uniformly at random from $\mathcal{P}_{\lfloor q/2 \rfloor}(t)$ yields that some vectors in $\mathcal{S}_{t,q}^{(n)}$ are more probable than others. Therefore, we need to understand the number of vectors with weight decomposition λ, for a fixed partition $\lambda \in \mathcal{P}_{\lfloor q/2 \rfloor}(t)$.

We now introduce a definition describing vectors whose Lee weight decomposition is based on a given integer partition.

Definition 5.3.2. For a positive integer n and a given partition $\lambda \in \mathcal{P}_{\lfloor q/2 \rfloor}(t)$ of a positive integer t, we say that a length-n vector x has *weight decomposition* λ *over* $\mathbb{Z}/q\mathbb{Z}$ if there is a one-to-one correspondence between the Lee weight of the nonzero entries of x and the parts of λ.

Example 5.3.3. Let $n = 5$ and let $\lambda = (2, 1, 1)$ be an integer partition of $t = 4$ over $\mathbb{Z}/7\mathbb{Z}$. All vectors of length n over $\mathbb{Z}/7\mathbb{Z}$ consisting of one element of Lee weight 2 and two elements of Lee weight 1 have weight decomposition λ.

We denote the set of all vectors of length n of the same weight decomposition $\lambda \in \mathcal{P}(t)$ by $\mathcal{V}_{t,\lambda}^{(n)}$. The following result gives an answer to the number of vectors with weight decomposition $\lambda \in \mathcal{P}_{\lfloor q/2 \rfloor}(t)$.

Lemma 5.3.4. *Let n, q and t be positive integers with $t \leq n$ and consider the set of partitions $\mathcal{P}_{\lfloor q/2 \rfloor}(t)$ of t with part sizes not exceeding $\lfloor q/2 \rfloor$. For any $\lambda \in \mathcal{P}_{\lfloor q/2 \rfloor}(t)$ the number of vectors of length n over $\mathbb{Z}/q\mathbb{Z}$ with weight decomposition λ is given by*

$$\left| \mathcal{V}_{t,\lambda}^{(n)} \right| = \begin{cases} 2^{\ell_\lambda} \,|\,\Pi_\lambda\,| \binom{n}{\ell_\lambda} & \text{if } q \text{ is odd} \\ 2^{\ell_\lambda - c_{r,\lambda}} \,|\,\Pi_\lambda\,| \binom{n}{\ell_\lambda} & \text{else} \end{cases},$$

where $c_{\lfloor q/2 \rfloor, \lambda} = |\,\{i \in \{1, \ldots, \ell_\lambda\} \mid \lambda_i = \lfloor q/2 \rfloor\}\,|$.

Proof. Recall from Definition 5.3.2 that $\mathcal{V}_{t,\lambda}^{(n)}$ consists of all length n vectors x whose nonzero entries are in one-to-one correspondence with the parts of λ. Let $x_{i_1}, \ldots, x_{i_{\ell_\lambda}}$ denote the nonzero positions of x and let us first consider the case where

$$\mathrm{wt_L}(x_{i_1}) = \lambda_1, \ \ldots, \ \mathrm{wt_L}(x_{i_{\ell_\lambda}}) = \lambda_{\ell_\lambda}. \tag{5.9}$$

Finding the number of such vectors relies on the "selection with repetition" problem [79, Section 1.2], which implies that this number is exactly $\binom{\text{number of zeros} + \text{free spaces} - 1}{\text{free spaces} - 1}$, i.e.,

$$\binom{(n - \ell_\lambda) \ + \ (\ell_\lambda + 1) \ - \ 1}{(\ell_\lambda + 1) \ - \ 1} = \binom{n}{\ell_\lambda},$$

where with "free spaces" we mean all the possible gaps in front, between and at the end of the parts of λ. If q is odd, the number n_i of elements in $\mathbb{Z}/q\mathbb{Z}$ having a nonzero Lee weight i is always 2 for every possible Lee weight $i \in \{1, \ldots, \lfloor q/2 \rfloor\}$. Hence, there are $2^{\ell_\lambda}\binom{n}{\ell_\lambda}$ vectors satisfying (5.9). On the other hand, if q is even, then $n_i = 2$ for $i \in \{1, \ldots, \lfloor q/2 \rfloor - 1\}$ and

$n_{\lfloor q/2 \rfloor} = 1$. Let us define the number of parts of λ equal to $\lfloor q/2 \rfloor$ by

$$c_{\lfloor q/2 \rfloor, \lambda} = |\{i \in \{1, \dots, \ell_\lambda\} \mid \lambda_i = \lfloor q/2 \rfloor\}|.$$

Then the number of parts of λ that can be flipped is $2^{\ell_\lambda - c_{\lfloor q/2 \rfloor, \lambda}}$. Hence, the number of vectors satisfying (5.9) is $2^{\ell_\lambda - c_{\lfloor q/2 \rfloor, \lambda}} \binom{n}{\ell_\lambda}$.

Finally, since the ordering of the nonzero elements of x is not necessarily the same as the order of the parts of λ, we multiply $\binom{n}{\ell_\lambda}$ by the number of permutations $|\Pi_\lambda|$ of λ and obtain the desired result. □

Finally, the actual vector construction over $\mathbb{Z}/q\mathbb{Z}$, described in Algorithm 1, mainly consists of picking a partition $\lambda \in \mathcal{P}_{\lfloor q/2 \rfloor}(t)$ of the Lee weight t with part sizes not exceeding $\lfloor q/2 \rfloor$. The probability of $x \in \mathcal{S}_{t,q}^{(n)}$ with weight decomposition $\lambda \in \mathcal{P}_{\lfloor q/2 \rfloor}(t)$ is given by

$$p_\lambda := \frac{\left| \mathcal{V}_{t,\lambda}^{(n)} \right|}{\sum_{\tilde{\lambda} \in \mathcal{P}_{r(t)}} \left| \mathcal{V}_{t,\tilde{\lambda}}^{(n)} \right|}.$$

The idea is to choose the integer partition according to the probability mass function $\mathcal{X}_{t,q}^{(n)}$ defined by the probabilities p_λ, for $\lambda \in \mathcal{P}_{\lfloor q/2 \rfloor}(t)$. We will denote this procedure by

$$\lambda \xleftarrow{\mathcal{X}_{t,q}^{(n)}} \mathcal{P}_{\lfloor q/2 \rfloor}(t).$$

We then randomly flip the elements of the partition modulo q and assign these values to randomly chosen positions of the error vector. Choosing an element a uniformly at random from a given set $\mathcal{A}$ will be denoted by $a \xleftarrow{\$} \mathcal{A}$. Additionally, let $\Pi(x)$ be a random permutation of a vector $x \in \mathcal{A}^n$. We want to emphasize at this point that for fixed parameters n, t and q the computation of $\mathcal{X}_{t,q}^{(n)}$ needs to be done only once at the beginning, since the distribution is only dependent on these parameters and does not change anymore.

Algorithm 1 Drawing a vector uniformly at random from $\mathcal{S}_{t,q}^{(n)}$

Require: $n, q, t \in \mathbb{N}_{>0}$, distribution $\mathcal{X}_{t,q}^{(n)}$
Ensure: $e \xleftarrow{\$} \mathcal{S}_{t,q}^{(n)}$

$\lambda \xleftarrow{\mathcal{X}_{t,q}^{(n)}} \mathcal{P}_{\lfloor q/2 \rfloor}(t)$
$F = \{f_1, \dots, f_{\ell_\lambda}\} \xleftarrow{\$} \{\pm 1\}^{\ell_\lambda}$
$\mathrm{supp}(e) \xleftarrow{\$} \{S \subset \{1, \dots, n\} : |S| = \ell_\lambda\}$
for $i = 1, \dots, n$ **do**
    if $i \in \mathrm{supp}(e)$ **then**
        $e_i \leftarrow f_i \cdot \lambda_i$
    else
        $e_i = 0$
    end if
end for
return $\Pi(e)$

Theorem 5.3.5. *Let n, q and t be positive integers. Algorithm 1 draws a vector uniformly at random among $\mathcal{S}_{t,q}^{(n)}$.*

Proof. First note that we can describe the n-dimensional Lee-sphere of radius t over $\mathbb{Z}/q\mathbb{Z}$ as the disjoint union over all sets $\mathcal{V}_{t,\lambda}^{(n)}$ where $\lambda \in \mathcal{P}_{\lfloor q/2 \rfloor}(t)$, i.e.,

$$\mathcal{S}_{t,q}^{(n)} = \bigsqcup_{\lambda \in \mathcal{P}_{\lfloor q/2 \rfloor}(t)} \mathcal{V}_{t,\lambda}^{(n)}.$$

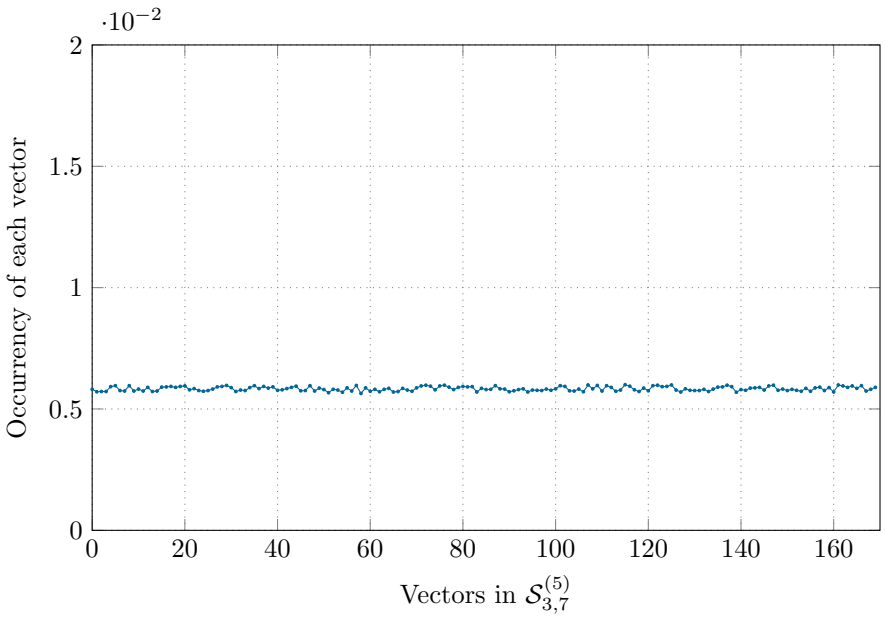

FIGURE 5.7: Distribution of 10^6 randomly constructed Lee error vectors using Algorithm 1 for vectors $x \in \mathcal{S}_{3,7}^{(5)}$.

Hence, we want to pick $\lambda \in \mathcal{P}_{\lfloor q/2 \rfloor}(t)$ such that all the vectors in $\mathcal{S}_{t,q}^{(n)}$ are equally probable to be drawn. The choice of λ is decisive for the set $\mathcal{V}_{t,\lambda}^{(n)}$. Since $\left| \mathcal{V}_{t,\lambda}^{(n)} \right|$ changes with λ, we pick λ according to distribution p_λ from $\mathcal{X}_{t,q}^{(n)}$ using Lemma 5.3.4 and the result follows. □

Figure 5.7 supports the result in Theorem 5.3.5. It shows the distribution for each vector $x \in \mathcal{S}_{3,7}^{(5)}$ when sampling 10^6 among them using Algorithm 1. Notice, that $\left| \mathcal{S}_{3,7}^{(5)} \right| = 170$ and the average uniform distribution should be close to $\frac{1}{170} \approx 0.00588$.

Algorithm Based on Enumerative Coding

Alternatively, we may draw vectors of length n and a fixed Lee weight t over $\mathbb{Z}/q\mathbb{Z}$ uniformly at random using ideas from enumerative coding [44]. This idea is also used to draw errors in the sum-rank and sum-subspace metric [105, 20]. Recall that $\left| \mathcal{S}_{i,q}^{(n)} \right|$ denotes the number of vectors of length n that have Lee weight exactly i. The main idea of this approach is to draw an integer within the interval $\left[0, \left| \mathcal{S}_{i,q}^{(n)} \right| - 1\right]$ uniformly at random and consider this as the index, from which the vector of fixed Lee weight can be constructed uniquely.

Denote the number of elements in $\mathbb{Z}/q\mathbb{Z}$ that have Lee weight i as N_i. Note, that each N_i corresponds to a coefficient in the generating function of the one-dimensional Lee-sphere of radius i over $\mathbb{Z}/q\mathbb{Z}$ presented in (3.10). Recall that the set of integer partitions of i into part sizes not exceeding $\lfloor q/2 \rfloor$ was denoted by $\mathcal{P}_{\lfloor q/2 \rfloor}(i)$. As integer partitions are not ordered, and they do not consider zeros, let us define the set of Lee weight decompositions of the Lee weight i for vectors of length n as

$$\mathcal{W}_{\lfloor q/2 \rfloor, i}^{(n)} := \left\{ \omega \in (\mathbb{Z}/\lfloor q/2 \rfloor \mathbb{Z})^n \;\middle|\; \sum_{j=1}^{n} \omega_j = i \right\}.$$

Thus, we can rewrite

$$\left| \mathcal{S}_{i,q}^{(n)} \right| = \sum_{\omega \in \mathcal{W}_{\lfloor q/2 \rfloor, i}^{(n)}} \prod_i N_{\lambda_i}.$$

This allows us to compute $\left|\mathcal{S}_{i,q}^{(n)}\right|$ recursively as

$$\left|\mathcal{S}_{i,q}^{(n)}\right| = \begin{cases} 0 & \text{if } i > \lfloor \frac{q}{2} \rfloor n, \\ N_i & \text{if } n = 1 \text{ and } i \leq \lfloor \frac{q}{2} \rfloor, \\ \sum\limits_{i'=\max\{0,i-(n-1)\lfloor q/2 \rfloor\}}^{\min\{i,\lfloor q/2 \rfloor\}} N_{i'} \left|\mathcal{S}_{i-i',q}^{(n-1)}\right| & \text{if } n > 1 \text{ and } i \leq \lfloor \frac{q}{2} \rfloor n, \end{cases}$$

where the lower limit in the sum comes from the restriction that

$$i - i' \leq (n-1)\left\lfloor \frac{q}{2} \right\rfloor \quad \text{if and only if} \quad i' \geq i - (n-1)\left\lfloor \frac{q}{2} \right\rfloor.$$

This leads to an efficient algorithm that draws vectors of length n and Lee weight i over $\mathbb{Z}/q\mathbb{Z}$ uniformly at random. In particular, it constructs a Lee weight decomposition $\omega \in \mathcal{W}_{\lfloor q/2 \rfloor, i}^{(n)}$ which is then transformed into the error vector e by randomly choosing the signs of the nonzero entries.

1. Draw an integer D uniformly at random from $\left[0, \left|\mathcal{S}_{i,q}^{(n)}\right| - 1\right]$. This is the index of the chosen vector.
2. Choose the first entry of the weight decomposition ω_1 as the largest integer such that

$$D'(\omega_1) = \sum_{i'=\max\{0,i-(n-1)\lfloor q/2 \rfloor\}}^{\min\{\omega_1, q/2\}} N_{i'} \left|\mathcal{S}_{i-i',q}^{(n-1)}\right| \leq D$$

   holds.
3. Update the number of sequences having a prefix with weight decomposition t_1 accordingly. This is necessary since we compute the weight distribution rather than the vector itself.
4. Repeat the steps with the updated values for D, $i - \omega_1$ and $n - 1$.
5. Once the weight decomposition ω is constructed, choose z signs $\{+, -\}^z$ uniformly at random and apply them to the z nonzero entries in i to obtain the resulting vector e.

We have sampled 10^6 Lee-error vectors of length $n = 5$ and Lee weight $i = 3$ over $\mathbb{Z}/7\mathbb{Z}$. The results in Figure 5.8 show that this construction also yields a uniform distribution.

5.3.2 The Scalar Multiplication Problem

While we know that the Hamming weight of a vector over a finite field is invariant under multiplication with a nonzero scalar, the Lee weight can possibly change. We want to emphasize that over a finite integer ring, that a nonzero element $a \in \mathbb{Z}/q\mathbb{Z}$ can turn into zero under multiplication with a nonzero scalar $b \in \mathbb{Z}/q\mathbb{Z}$ if and only if both a and b are nonunits. In this case, both the Hamming weight and the Lee weight of a become zero when multiplying a by b. However, if one of a and b is a unit, the Hamming weight of a remains invariant under multiplication by b which is not true for the Lee metric.

In this section, we analyze the behavior of the Lee weight of a vector when multiplied by a scalar. Let us give a quick example to give an intuition to the problem.

Example 5.3.6. Consider the vector $x = (1, 0, 0, 1, 0, 2, 0, 0)$ over $\mathbb{Z}/7\mathbb{Z}$, which has Lee weight $\mathrm{wt}_{\mathsf{L}}(x) = 4$ and Hamming weight $\mathrm{wt}_{\mathsf{H}}(x) = 3$. Let us chose a nonzero scalar $a = 2 \in \mathbb{Z}/7\mathbb{Z}$ and let us stretch x by a, i.e., $a \cdot x = (2, 0, 0, 2, 0, 4, 0, 0)$. Still, the Hamming weight is equal to 3 but the Lee weight now is given by 7.

Recalling that the Lee metric coincides with the Hamming metric over $\mathbb{Z}/2\mathbb{Z}$ and $\mathbb{Z}/3\mathbb{Z}$, in the following we focus only on the case where the Lee weight is different from the Hamming weight, i.e., we focus on $\mathbb{Z}/q\mathbb{Z}$ with $q > 3$.

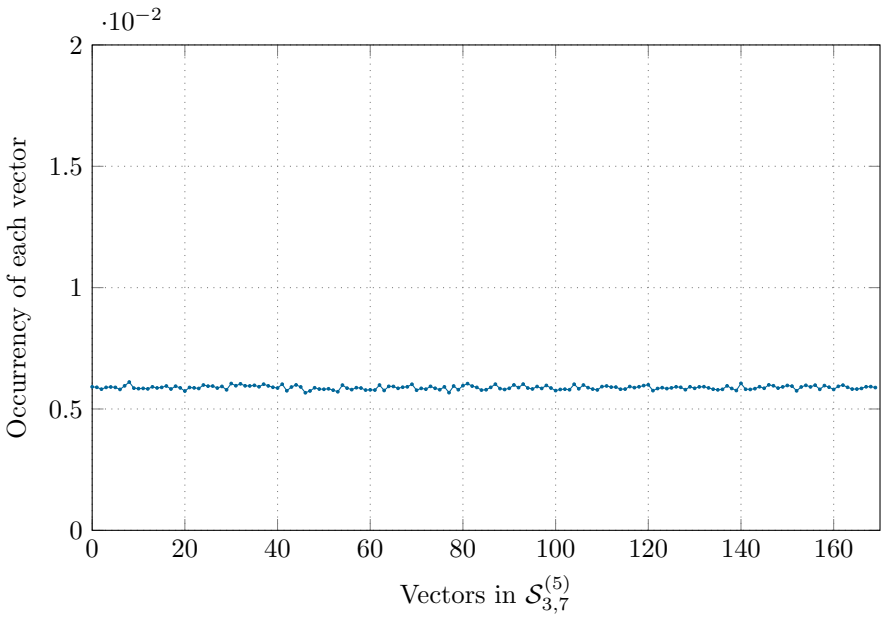

FIGURE 5.8: Distribution of 10^6 randomly constructed Lee error vectors using the enumerative coding-based method for vectors $x \in \mathcal{S}_{3,7}^{(5)}$.

Remark 5.3.7. Even though we will not discuss the following in detail, we want to emphasize at this point that the Hamming weight is *not* invariant under multiplication with a nonzero scalar when working over a finite integer ring that is not a field.

We now formalize the problem we are interested in.

Problem 5.3.8. Consider the ring of integers $\mathbb{Z}/q\mathbb{Z}$, with $q > 3$, and a random vector $x \in (\mathbb{Z}/q\mathbb{Z})^n$ with Lee weight $\mathrm{wt}_{\mathsf{L}}(x) = t$ uniformly distributed in $\mathcal{S}_{t,q}^{(n)}$. Let $a \in (\mathbb{Z}/q\mathbb{Z}) \setminus \{0\}$ be chosen uniformly at random. Find the probability that the Lee weight of $a \cdot x$ is less than the Lee weight t of x, i.e.,

$$\mathbb{P}\left(\mathrm{wt}_{\mathsf{L}}(a \cdot x) < t\right).$$

For simplicity, let us define the following event

$$F := \{\mathrm{wt}_{\mathsf{L}}(a \cdot x) < \mathrm{wt}_{\mathsf{L}}(x)\}.$$

We denote by Q_x the empirical distribution of the elements of x. Recall the distribution $P^\star$ defined in (5.6). We rewrite $\mathbb{P}(F)$ by distinguishing between vectors x with Q_x close to $P^\star$ and all others, where by "close" we mean with respect to the Kullback-Leibler divergence (see Definition 2.1.5), i.e., Q_x satisfies $\mathsf{D}(Q_x \,||\, P^\star) < \varepsilon$ for some $\varepsilon > 0$ small. We have that

$$\begin{aligned}\mathbb{P}(F) =& \,\mathbb{P}\left(\mathrm{wt}_{\mathsf{L}}(a \cdot x) < t \mid \mathsf{D}(Q_x \,||\, P^\star) < \varepsilon\right) \mathbb{P}\left(\mathsf{D}(Q_x \,||\, P^\star) < \varepsilon\right) \\ &+ \mathbb{P}\left(\mathrm{wt}_{\mathsf{L}}(a \cdot x) < t \mid \mathsf{D}(Q_x \,||\, P^\star) \geq \varepsilon\right) \mathbb{P}\left(\mathsf{D}(Q_x \,||\, P^\star) \geq \varepsilon\right) \\ \leq& \,\mathbb{P}\left(\mathrm{wt}_{\mathsf{L}}(a \cdot x) < t \,|\, \mathsf{D}(Q_x \,||\, P^\star) < \varepsilon\right) + \mathbb{P}\left(\mathsf{D}(Q_x \,||\, P^\star) \geq \varepsilon\right), \qquad (5.10)\end{aligned}$$

where the inequality (5.10) holds, since every probability can be upper bounded by one. In the next subesection, we see that the upper bound is well-defined, i.e., that indeed

$$\mathbb{P}\left(\mathrm{wt}_{\mathsf{L}}(a \cdot x) < t \,|\, \mathsf{D}(Q_x \,||\, P^\star) < \varepsilon\right) + \mathbb{P}\left(\mathsf{D}(Q_x \,||\, P^\star) \geq \varepsilon\right) \leq 1.$$

Note that the probability $\mathbb{P}(F)$ depends on three parameters: the length n of the constructed vector x, the size q of the integer ring and the given Lee weight t of x. The evaluation of the bound (5.10) is challenging for $q > 3$, finite n and generic t. In the following we describe how to attack the problem for large n.

We are going to give an answer to Problem 5.3.8 in the asymptotic regime here, i.e., we consider the case, where the block length n tends to infinity. In fact, Corollary 5.1.6 allows us to assume that the entries of a sequence x drawn uniformly in $\mathcal{S}^{(n)}_{n\delta,q}$ follow the distribution $P^\star$ as n grows. Hence, in the asymptotic regime, Problem 5.3.8 reduces to estimating the probability $\mathbb{P}\left(\mathrm{wt_L}(a \cdot x) \leq \mathrm{wt_L}(x) \mid \mathsf{D}(Q_x \,||\, P^\star) < \varepsilon\right)$. In that case, we apply the definition of the Lee weight of a vector x, the assumption that the entries of x are distributed as in (5.6) yields, in the limit of n, the following equivalent description of the desired probability

$$\lim_{n \longrightarrow \infty} \mathbb{P}(F) = \mathbb{P}\left(\sum_{i=1}^{q-1} \mathrm{e}^{-\beta\, \mathrm{wt_L}(i)}\, \mathrm{wt_L}([a \cdot i]_q) < \sum_{i=1}^{q-1} \mathrm{e}^{-\beta\, \mathrm{wt_L}(i)}\, \mathrm{wt_L}(i) \right). \tag{5.11}$$

By the symmetric property of the Lee weight, Property (3.6), we can run the sum only up to r. Nevertheless, we need to distinguish between even or odd ring order q. In particular, for q odd we rewrite (5.11) as

$$\lim_{n \longrightarrow \infty} \mathbb{P}(F) = \mathbb{P}\left(0 < \sum_{i=1}^{r} \mathrm{e}^{-\beta i} (i - \mathrm{wt_L}([a \cdot i]_q)) \right), \tag{5.12}$$

whereas for q even (5.11) is equivalent to

$$\lim_{n \longrightarrow \infty} \mathbb{P}(F) = \mathbb{P}\left(0 < \sum_{i=1}^{r-1} 2\mathrm{e}^{-\beta i} (i - \mathrm{wt_L}([a \cdot i]_q)) + \mathrm{e}^{-\beta r} (r - \mathrm{wt_L}([a \cdot r]_q)) \right), \tag{5.13}$$

where $[a \cdot i]_q$ denotes the reduction of $a \cdot i \mod q$.

Since we want $\mathbb{P}(F)$ to be small (or equal to zero), we need to understand under what circumstances the sums in (5.12) and (5.13) are non-positive. Notice that both the sums $\sum_{i=1}^{r} \mathrm{e}^{-\beta i}(i - \mathrm{wt_L}([a \cdot i]_q))$ and $\sum_{i=1}^{r-1} 2\mathrm{e}^{-\beta i}(i - \mathrm{wt_L}([a \cdot i]_q)) + \mathrm{e}^{-\beta r}(r - \mathrm{wt_L}([a \cdot r]_q))$ are dependent on q and β, where β depends on δ. If we fix these parameters, we are able to compute the sum and hence (5.11). We therefore fixed q and evaluated the two expressions for different values of δ. Let us denote by $\delta_{\max}$ the maximum δ for which (5.12) or rather (5.13) are equal to zero. The table shows the values of the threshold $\delta_{\max}$ for different ring orders q.

q	5	7	8	9	11	15	16	31	32	33	53
$\lfloor q/2 \rfloor$	2	3	4	4	5	7	8	15	16	16	26
$\delta_{\max}$	1.2	1.714	2	1.962	2.727	3.310	4	7.741	8	8.242	13.245

TABLE 5.1: Maximal normalized Lee weight $\delta_{\max}$ over $\mathbb{Z}/q\mathbb{Z}$ such that $\mathbb{P}(F) = 0$ as $n \longrightarrow \infty$, for some values of q compared to the maximal possible normalized Lee weight $\lfloor q/2 \rfloor$.

Table 5.1 already suggests some bounds on the threshold $\delta_{\max}$. For instance, we observe that if $q = 2^m$, for some $m > 0$, $\delta_{\max}$ seems to be 2^{m-2}. In the following let us consider on different cases.

Field Case

For the case where q is a prime we are able to give an asymptotic answer to Problem 5.3.8. Indeed, we show that it is never possible to reduce the Lee weight of a vector by a nontrivial scalar multiplication as the length of the vector grows large.

Theorem 5.3.9. *Let q be an odd prime number and let $x \in (\mathbb{Z}/q\mathbb{Z})^n$ where each x_i was drawn following the Boltzmann-like distribution, i.e., for every $i = 1, \ldots, n$ it holds $\mathbb{P}(x_i = k) = \kappa \exp(-\lambda\, \mathrm{wt_L}(k))$ for every $k \in \mathbb{Z}/q\mathbb{Z}$. Let δ denote the average Lee weight per entry x_i*

of x and choose $a \in (\mathbb{Z}/q\mathbb{Z})^\times$. If $\delta < \frac{q^2-1}{4q}$, then

$$\lim_{n \longrightarrow \infty} \mathbb{P}(F) = 0.$$

It has been shown in [137, Theorem 2] that if we chose an element randomly from $\mathbb{Z}/q\mathbb{Z}$ its expected Lee weight is exactly the threshold $\frac{q^2-1}{4q}$ mentioned in Theorem 5.3.9. In order to prove the result, let us first prove the following lemma.

Lemma 5.3.10. *Let $\pi = (\pi_1, \ldots, \pi_n)$ a sequence of distinct positive integers satisfying $\pi_1 < \pi_2 < \cdots < \pi_n$. Define the following swap operation:*

"Two elements π_i and π_j with $i < j$ can only be swapped if $\pi_i < \pi_j$."

Applying an arbitrary finite sequence of swap operations we are able to generate every permutation of π.

Let us give a short example to illustrate the above defined swap operation.

Example 5.3.11. Consider the sequence $\pi = (2, 6, 4, 8)$. We can only swap two elements π_i and π_j if the left most of the two is smaller than the right most element of the two. For instance the elements 2 and 8 can be swapped with any other element of π. On the other hand the entry 4 can only be swapped with either the element 2 or the element 8.

Proof. We will use induction on the length n of the vector π.
For the base case, we assume that $n = 1$. Hence, $\pi = (\pi_1)$ consists only of one positive integer. Then clearly there is nothing to show.
Assume then that the result of Lemma 5.3.10 applies to a vector $\pi = (\pi_1, \ldots, \pi_n)$ with $\pi_1 < \pi_2 < \cdots < \pi_n$. We refer to this assumption as the *induction hypothesis*.
Finally, assume we are given $\pi = (\pi_1, \ldots, \pi_{n+1})$ such that $\pi_1 < \pi_2 < \cdots < \pi_{n+1}$. By the induction hypothesis, if we focus only on $(\pi_2, \ldots, \pi_{n+1})$ by applying a sequence of swap operations we are able to generate every permutation of $(\pi_2, \ldots, \pi_{n+1})$. Let σ be an arbitrary permutation of it and consider then $(\pi_1, \sigma(\pi_2), \ldots \sigma(\pi_{n+1}))$. Since for every $k \in \{2, 3, \ldots, n+1\}$ we have $\pi_1 < \sigma(\pi_k)$, we can apply the swap operation up to n times to the element π_1 and its right-neighboring element and obtain any permutation of $(\pi_1, \sigma(\pi_2), \ldots \sigma(\pi_{n+1}))$. □

Lemma 5.3.12. *Let $\pi = (\pi_1, \ldots, \pi_n)$ be a sequence of n positive integers such that $\pi_1 < \pi_2 < \cdots < \pi_n$. We denote by $P[n]$ the set of permutations of π. Assume that $f : \mathbb{N} \longrightarrow \mathbb{R}$ is a strictly monotone decreasing function, i.e., $f(i) > f(j)$ whenever $i < j$. Then*

$$\pi = \arg \min_{x \in P[n]} \sum_{i=1}^{n} x_i f(i).$$

Proof. By Lemma 5.3.10 we know that starting from π and applying a sequence of swap operations we can generate any other permutation of π. Let x be a permutation of π obtained after one arbitrary swap operation, say $x_i = \pi_j$ and $x_j = \pi_i$ with $i < j$. The swap rule implies then that $\pi_i < \pi_j$. We would like to have $\sum_{i=1}^{n} \pi_i f(i) < \sum_{i=1}^{n} x_i f(i)$. Since we only switched two elements this is equivalent to

$$\pi_i f(i) + \pi_j f(j) < \pi_j f(i) + \pi_i f(j).$$

Which, by simultaneous operations on the left and right, yields

$$(\pi_j - \pi_i) f(j) < (\pi_j - \pi_i) f(i).$$

This inequality is clearly true, since $(\pi_j - \pi_i) > 0$, $i < j$ and since f is monotone decreasing. Since we can apply the same procedure iteratively and by Lemma 5.3.10 the desired result follows. □

Proof of Theorem 5.3.9. Firstly, δ denotes the average Lee weight per entry of a vector $x \in (\mathbb{Z}/q\mathbb{Z})^n$. Hence, we can write

$$\delta = \kappa \sum_{i=0}^{q-1} \mathrm{wt}_{\mathsf{L}}(i) \exp(-\lambda \, \mathrm{wt}_{\mathsf{L}}(i)).$$

Note that if $\lambda = 0$ we have $\delta = \kappa \sum_{i=0}^{q-1} \mathrm{wt}_{\mathsf{L}}(i)$, implying that every Lee weight is equally likely. More explicitly that means, $\lambda = 0$ yields

$$\delta = \kappa \sum_{i=0}^{q-1} \mathrm{wt}_{\mathsf{L}}(i) = \frac{1}{q} 2 \sum_{i=1}^{(q-1)/2} i = \frac{(q-1)(q+1)}{4q} = \frac{q^2-1}{4q},$$

where $(\star)$ is the closed form of the sum of the first $(q-1)/2$ integers. Furthermore, if $\lambda > 0$ then $\delta < \frac{q^2-1}{4q}$ as well as if $\lambda < 0$ then $\delta > \frac{q^2-1}{4q}$.
Now let us come back to the event E we want to estimate and assume at this point that $\delta < \frac{q^2-1}{4q}$. Note that due to symmetry of the Lee weight, we have for any two nonzero integers $a, b \in \mathbb{Z}/q\mathbb{Z}$ that

$$\mathrm{wt}_{\mathsf{L}}(ab) = \mathrm{wt}_{\mathsf{L}}(a(q-b)) = \mathrm{wt}_{\mathsf{L}}(aq - ab) = \mathrm{wt}_{\mathsf{L}}(q - ab) = \mathrm{wt}_{\mathsf{L}}(-ab).$$

Since q is a prime $\mathrm{wt}_{\mathsf{L}}([ai]_q)$ represents a permutation of the Lee weights $\mathrm{wt}_{\mathsf{L}}(i)$ for every $i \in \mathbb{Z}/q\mathbb{Z}$, where $a \in (\mathbb{Z}/q\mathbb{Z})^\times$. Let us denote by $\pi_a(i) := \mathrm{wt}_{\mathsf{L}}([ai]_q)$ the permutation caused by a. Since the Lee weight should not be able to be reduced by any multiplication with a nonzero scalar, we want to show that

$$\mathbb{P}\left(0 < \sum_{i=1}^{q-1} \mathrm{e}^{-\lambda i} (\mathrm{wt}_{\mathsf{L}}(i) - \mathrm{wt}_{\mathsf{L}}([a \cdot i]_q))\right) = 0.$$

Equivalently, we show $\mathbb{P}\left(\sum_{i=1}^{q-1} \mathrm{e}^{-\lambda i} \, \mathrm{wt}_{\mathsf{L}}([a \cdot i]_q) < \sum_{i=1}^{q-1} \mathrm{e}^{-\lambda i} \, \mathrm{wt}_{\mathsf{L}}(i)\right) = 0$.

Using the permutation representation π_a from above, we can rewrite

$$\sum_{i=1}^{q-1} \mathrm{e}^{-\lambda \, \mathrm{wt}_{\mathsf{L}}(i)} \, \mathrm{wt}_{\mathsf{L}}([a \cdot i]_q) = \sum_{i=1}^{q-1} \pi_a(\mathrm{wt}_{\mathsf{L}}(i)) \mathrm{e}^{-\lambda i} = 2 \sum_{i=1}^{(q-1)/2} \pi_a(i) \mathrm{e}^{-\lambda i}.$$

Since $\lambda > 0$, the function $\exp(-\lambda i)$ is strictly monotone decreasing and convex. Furthermore, π_a is a permutation of the sequence of integers from 1 to $(q-1)/2$, hence by Lemma 5.3.12, for any nonzero $a \in \mathbb{Z}/q\mathbb{Z}$,

$$2 \sum_{i=1}^{(q-1)/2} \pi_a(i) \mathrm{e}^{-\lambda i} \geq 2 \sum_{i=1}^{(q-1)/2} i \mathrm{e}^{-\lambda i} = \sum_{i=1}^{q-1} \mathrm{e}^{-\lambda \, \mathrm{wt}_{\mathsf{L}}(i)} \, \mathrm{wt}_{\mathsf{L}}(i)$$

and thus, since $\lambda > 0$, $\lim_{n \longrightarrow \infty} \mathbb{P}(F) = 0$. □

Remark 5.3.13. We want to stress at this point that the exact same result applies in the case where q is non-prime and $a \in (\mathbb{Z}/q\mathbb{Z})^\times$ be a unit modulo q.

Special Case: Power of 2

Let us focus on the case where the ring order q is a power of 2, i.e., there exists a positive integer $h \in \mathbb{Z}$ such that $q = 2^h$. Let us denote by

$$\mathcal{Z}_q := (\mathbb{Z}/q\mathbb{Z}) \setminus \left(\{0\} \cup (\mathbb{Z}/q\mathbb{Z})^\times\right)$$

the set of nonzero non-units modulo q. Since $q = 2^h$, every nonzero non-unit is a multiple of 2, hence we can rewrite

$$\mathcal{Z}_q = \left\{2m \mid m \in \left\{1, \ldots, 2^{h-1}\right\}\right\}. \tag{5.14}$$

Lemma 5.3.14. *Let $q = 2^h$ for some positive integer $h \in \mathbb{Z}$. For any $a \in \mathcal{Z}_q$ we have*

$$\mathrm{wt}_{\mathsf{L}}([a(q/2)]_q) = 0.$$

Proof. Let $a \in \mathcal{Z}_q$ be chosen arbitrarily. By (5.14) there exists an integer $k \in \{1, \dots, 2^{h-1}\}$ such that $a = 2k$. Hence, we obtain

$$\mathrm{wt}_{\mathsf{L}}\left([a(q/2)]_q\right) = \mathrm{wt}_{\mathsf{L}}\left([2k2^{h-1}]_q\right) = \mathrm{wt}_{\mathsf{L}}\left([k2^h]_q\right) = \mathrm{wt}_{\mathsf{L}}\left(0\right) = 0.$$

□

Similar to Subsection 5.3.2 above, we will show that the limit described in Equation (5.13) is zero. Explicitly, we show that since $\beta > 0$, for any $a \in (\mathbb{Z}/q\mathbb{Z}) \setminus \{0\}$ it holds

$$\sum_{i=1}^{q/2-1} 2\mathrm{e}^{-\beta i} i + \mathrm{e}^{-\beta q/2} q/2 \leq \sum_{i=1}^{q/2-1} 2\mathrm{e}^{-\beta i} \,\mathrm{wt}_{\mathsf{L}}([a \cdot i]_q) + \mathrm{e}^{-\beta q/2} \,\mathrm{wt}_{\mathsf{L}}([a \cdot q/2]_q). \tag{5.15}$$

By Remark 5.3.13 we know that this is fulfilled, whenever a is a unit. Therefore, our goal is to show Inequality (5.15) for $a \in \mathcal{Z}_q$ a nonzero nonunit.

In the following let $a \in \mathcal{Z}_q$ be arbitrary. Let us define the *minimal zero-multiplier* of a as

$$\mu_a := \min_{j>0}(aj \equiv 0 \mod q).$$

Since μ_a is minimal it holds that $a\mu_a = q$.

Lemma 5.3.15. *The sum of all Lee weights over $\mathbb{Z}/q\mathbb{Z}$ with $q = 2^h$ coincides with the sum of the scaled Lee weights mod q, i.e.,*

$$\sum_{i=1}^{q-1} \mathrm{wt}_{\mathsf{L}}([ai]_q) = \frac{q^2}{4} = \sum_{i=1}^{q-1} \mathrm{wt}_{\mathsf{L}}(i).$$

Proof. By the symmetry property of the Lee weight, we have

$$\sum_{i=1}^{q-1} \mathrm{wt}_{\mathsf{L}}([ai]_q) = 2 \sum_{i=1}^{q/2-1} \mathrm{wt}_{\mathsf{L}}([ai]_q) + \mathrm{wt}_{\mathsf{L}}([aq/2]_q),$$

which by Lemma 5.3.14 is just equal to

$$2 \sum_{i=1}^{q/2-1} \mathrm{wt}_{\mathsf{L}}([ai]_q).$$

By Remark 5.3.13 again, we can exclude the cases, where a is not a power of 2, since in that case it must be a product $a = 2^\ell u$, where $u \in (\mathbb{Z}/q\mathbb{Z})^\times$ and hence is a permuted version of the case where $a = 2^\ell$. Since a is even, it divides $q = 2^h$. By the symmetry property of the Lee weight and by Lemma 5.3.14, $(\mathrm{wt}_{\mathsf{L}}([ai]_q))_{i=1,\dots,q-1}$ is repeated (permuted) sequence of

$$a, 2a, \dots, \frac{\mu_a a}{2}, \dots, 2a, a, 0, \tag{5.16}$$

where this part is repeated $a/2$ times. Observing the symmetry in Equation (5.16), we get

$$\begin{aligned}
2 \sum_{i=1}^{q/2-1} \mathrm{wt}_{\mathsf{L}}([ai]_q) &= 2\left(a^2 \sum_{i=1}^{\frac{\mu_a}{2}} i + a2^{h-2}\right) \\
&= 2\left(a^2 \frac{(\frac{\mu_a}{2} - 1)(\frac{\mu_a}{2})}{2} + a2^{h-2}\right) \\
&= 2\left(2^{h-3} a\mu_a\right) \\
&= 2^{2h-2}.
\end{aligned}$$

□

Lemma 5.3.15 implies hence, if the distribution of the elements in the vector $x \in \mathcal{S}^{(n)}_{\delta n, 2^h}$ is uniform (i.e., for $\beta = 0$), asymptotically we are not able to reduce the Lee weight of x by multiplying it with a nonzero scalar $a \in \mathbb{Z}/q\mathbb{Z}$.

5.4 Summary and Outlook

This chapter studied two channel models in the Lee metric, a memoryless channel model and a channel introducing an error of given Lee weight. The discrete memoryless Lee channel matches to the Lee metric under the decoding rule to decode to the nearest codeword. For the second channel model, the constant Lee-weight channel, the error vector is drawn uniformly at random among the set of vectors of the same Lee weight. We studied the typical sequences under the constraint that the types considered have the same expected value. This in fact reflects the choice of the error vector of fixed Lee weight. With the help of Sanov's Theorem (Theorem 2.2.8) and the conditional limit theorem (Theorem 2.2.9), a main result of this chapter consisted in the derivation of the marginal distribution of the constant Lee-weight channel. This distribution, strongly related to the Boltzmann distribution introduced in statistical mechanics, gave rise to asymptotically tight bounds for the size of n-dimensional spheres and balls in the Lee metric and bounds on the block error probability for both channel models which were discussed in Section 5.2.

The chapter was concluded with Section 5.3 with the study on vectors of a fixed Lee weight. In Section 5.3.1, we derived two algorithms to construct randomly vectors of given Lee weight. The algorithm requires the precomputation of the set of integer partitions which, for large parameters, is a complex task. An interesting problem is to come up with different algorithms to construct such vectors and to analyze their complexities. Additionally, by the central limit theorem and using the marginal distribution of the constant Lee-weight channel, a vector of growing length n has entries that follow a Gaussian distribution whose mean and variance can be determined by the marginal distribution. With this we would be able to construct a vector that has the desired weight with high probability. It would be interesting to formally introduce an algorithm in this way with a decision step, discarding vectors of a Lee weight that does not lie in some ε-neighborhood of the desired Lee weight, and to compare it with Algorithm 1. Furthermore, in Section 5.3.2 we formalized the *scalar multiplication problem* consisting in the problem of reducing or increasing the Lee weight of a given vector when multiplying it with a nontrivial scalar. Applying the marginal distribution of the constant Lee-weight channel, we showed that, in the limit of large block lengths, the Lee weight of a vector can never be reduced. The scalar multiplication problem has only partly been answered in the finite length regime. The open problem of characterizing the probability in the finite sequence length remains which would answer also the problem for any $\mathbb{Z}/q\mathbb{Z}$.

Chapter 6

Regular Lee-LDPC Codes

Low-density parity-check (LDPC) codes were introduced by Gallager in the early 1960's in [58]. They are binary linear error-correcting codes characterized by a sparse parity-check matrix, meaning that the number of nonzero entries in a parity-check matrix is small. After their invention they seem to have almost been forgotten for about thirty years. During these years only few researchers investigated LDPC codes. Among these results, Tanner gave a representation in terms of a bipartite graph [126]. This representation allows analyzing LDPC codes from a combinatorial point of view. In the late 1990's LDPC were then rediscovered simultaneously introducing different successful LDPC code designs, such as MacKay and Neal's near-Shannon capacity LDPC codes [88, 89] or designs based on graphical representations [55, 122, 132, 133]. Ever since, LDPC codes were widely studied. Generating random LDPC codes has the advantage that there is no underlying algebraic structure. Thus, especially for code-based cryptography LDPC codes seemed to be good candidates for some McEliece variant and were proposed in various contexts [12, 11]. Nevertheless, LDPC codes have shown one major drawback for cryptography: the sparsity of their parity-check matrices which leads to statistical attacks.

As mentioned, we can describe a parity-check matrix H by a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$ connecting the vertices. The set of vertices consists of two disjoint sets; the set of variable nodes $\{\mathsf{v}_1, \dots, \mathsf{v}_n\}$, representing the columns of H, and the set of check nodes $\{\mathsf{c}_1, \dots, \mathsf{c}_m\}$, representing the rows of H. A variable node v_i is connected to a check node c_j by an edge if and only if the corresponding entry h_{ij} in the parity-check matrix is nonzero. The *degree* d_{v} of a variable node v is the number of edges connected to v. The *neighbors* $\mathcal{N}(\mathsf{v})$ of a variable node v is the set of check nodes connected to v. Similarly, we define the degree d_{c} and the neighbors $\mathcal{N}(\mathsf{c})$ of a check node c. Let us give an example of the graphical representation of a parity-check matrix.

Example 6.0.1. For instance, consider the following (non-sparse) 4×8 parity-check matrix over $\mathbb{F}_2$ as a toy example

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{6.1}$$

Then the bipartite graph $\mathcal{G}$ describing H is shown in Figure 6.1.

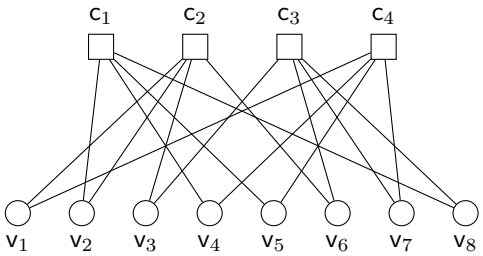

FIGURE 6.1: Graph representation via check nodes and variable nodes of the parity-check matrix (6.1).

There are several classes of LDPC codes depending on number of neighbors per variable node and check node, or other structural properties. Of particular interest in this thesis are *regular* LDPC codes which have a constant variable node degree $d_{\mathsf{v}} = v$ and a constant check node degree $d_{\mathsf{c}} = c$. We refer to these LDPC codes as (v, c)-regular LDPC codes. Furthermore, we denote by $\mathscr{C}_{v,c}^{n}$ the unstructured regular LDPC code ensemble of length n, i.e., the set of all LDPC codes defined by an $(m \times n)$ parity-check matrix, whose associated bipartite graph has constant variable node degree v and constant check node degree c. This ensemble has then the designed rate $R_0 = 1 - m/n$. Lately, LDPC codes are also studied over finite rings. In [124], the authors analyzed LDPC codes over in integer residue ring $\mathbb{Z}/q\mathbb{Z}$ defining the nonzero entries of the parity-check matrix over the units $(\mathbb{Z}/q\mathbb{Z})^{\times}$. As proposed in [124], when sampling an LDPC code from $\mathscr{C}_{v,c}^{n}$, we assume that the nonzero entries are drawn independently and uniformly at random from the set of units $(\mathbb{Z}/q\mathbb{Z})^{\times}$. In the following, let $\mathcal{C}$ always denote an $[n, k]$ linear block code over $\mathbb{Z}/q\mathbb{Z}$ and let $H \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ be a parity-check matrix of $\mathcal{C}$, where $m \geq n - k$.

We start the chapter by introducing the belief propagation algorithm and the message-passing decoding algorithm, and its adaption to the Lee metric. In a next step, we focus on the algebraic structure of the regular Lee-LDPC code ensembles discussing its average weight enumerator in Section 6.2. Lee-LDPC codes have been introduced in [114] together with a bit-flipping decoding variant, that they introduced as Lee-symbol flipping decoder. This code family defines the first Lee-metric code which is efficiently decodable. The average weight enumerator together with the bounds on the block error probabilities of the Lee channels derived in Section 5.2 yields bounds on the error-correction performance of the regular LDPC ensembles. Furthermore, the average weight enumerator provides important information about the codes performance in the error floor region. Finally, we analyze and compare the performance of LDPC codes over both channel models introduced in Chapter 5 under belief propagation and symbol message-passing decoding in Section 6.3. We discuss the main ingredients to adapt the symbol message-passing decoder to the Lee metric in both of the channel models. The results presented in this chapter are studied in [15, 17] in collaboration with Hannes Bartz, Gianluigi Liva and Joachim Rosenthal.

6.1 Message Passing Decoders

We briefly recall two message-passing algorithms for nonbinary LDPC codes. The first algorithm is the well-known (nonbinary) belief propagation algorithm [99]. The second algorithm is a message-passing algorithm where the messages exchanged between variable and check nodes are hard symbol estimates. The latter algorithm, dubbed symbol message-passing, generalizes the Gallager-B algorithm [58] and binary message-passing algorithm [82] to nonbinary alphabets.

Let us fix some notation used in the description of the two decoders. We consider a nonbinary alphabet $\mathbb{Z}/q\mathbb{Z}$, and we denote by $m_{\mathsf{v}\to\mathsf{c}}$ the message sent from variable node v to a neighboring check node c and vice versa $m_{\mathsf{c}\to\mathsf{v}}$ is the message sent from c to v. Furthermore, we will denote the likelihood at the variable node v input (associated with the corresponding channel observation) by

$$m_{\mathsf{v}} := \left(P_{Y|X}(y \mid 0), \ldots, P_{Y|X}(y \mid q-1)\right),$$

i.e., the vector of probabilities of the channel output y, conditioned on the q possible channel input values. For every connected variable node v and check node c we denote by h_{cv} the corresponding entry in the parity-check matrix H. Following [124] the nonzero entries of H are taken from the set of units. In this way, the inverse h_{cv}^{-1} is guaranteed to exist.

6.1.1 Belief Propagation Decoding

We consider now the belief propagation algorithm for nonbinary LDPC codes over finite rings. The decoder consists of four main steps that are outlined below, where Step 2 and 3 are repeated at most $\ell_{\max}$ times. For every connected variable node v and check node c we let Π_{cv} be the $q \times q$ permutation matrix induced by h_{cv}.

1. **Initialization.** Each variable node v receives the channel observation in the form of m_{v}. Then, the variable node v sends to each $\mathsf{c} \in \mathcal{N}(\mathsf{v})$ the permuted channel observation, i.e.,

$$m_{\mathsf{v}\to\mathsf{c}} = m_{\mathsf{v}} \cdot \Pi_{\mathsf{cv}}.$$

2. **Check node-to-variable node update.** Consider a given check node c and a neighboring variable node $\mathsf{v} \in \mathcal{N}(\mathsf{c})$. For the message $m_{\mathsf{c}\to\mathsf{v}}$, the check node computes the circular convolution of the incoming messages $m_{\mathsf{v}'\to\mathsf{c}}$ from all neighboring variable nodes $\mathsf{v}' \in \mathcal{N}(\mathsf{c}) \setminus \{\mathsf{v}\}$ as

$$u = \bigoplus_{\mathsf{v}'\in\mathcal{N}(\mathsf{c})\setminus\{\mathsf{v}\}} m_{\mathsf{v}'\to\mathsf{c}}$$

and sends to every neighboring variable node $\mathsf{v} \in \mathcal{N}(\mathsf{c})$ a permuted version of u according to the permutation Π_{cv}^{-1}, i.e., the check node-to-variable node message is

$$m_{\mathsf{c}\to\mathsf{v}} = u \cdot \Pi_{\mathsf{cv}}^{-1}.$$

3. **Variable node-to-check node update.** The variable node v computes the Schur product (i.e., a component-wise product) $\odot$ of all incoming messages but the one from check node c and normalizes the result by a constant K (to obtain a proper probability vector)

$$v = K \bigodot_{\mathsf{c}'\in\mathcal{N}(\mathsf{v})\setminus\{\mathsf{c}\}} m_{\mathsf{c}'\to\mathsf{v}}.$$

Finally, it applies the permutation matrix Π_{cv} to the vector v and sends the following message to the check node c

$$m_{\mathsf{v}\to\mathsf{c}} = v \cdot \Pi_{\mathsf{cv}}.$$

4. **Final decision.** The final decision happens at the variable node side. After at most $\ell_{\max}$ iterations of steps 2 and 3 each variable node computes the Schur product of all incoming messages, yielding the a posteriori probability estimate

$$m_{\mathsf{v}}^{\mathsf{APP}} = \bigodot_{\mathsf{c}\in\mathcal{N}(\mathsf{v})} m_{\mathsf{c}\to\mathsf{v}}.$$

The decision $\hat{x}$ is the index of the maximal entry of $m_{\mathsf{v}}^{\mathsf{APP}}$

$$\hat{x} = \underset{i\in\mathbb{Z}/q\mathbb{Z}}{\arg\max}\, m_{\mathsf{v},i}^{\mathsf{APP}}.$$

6.1.2 Symbol Message-Passing Decoding

The symbol message-passing algorithm is a message-passing algorithm for nonbinary LDPC codes, where each message exchanged by a variable node/check node pair is a symbol, i.e., a hard estimate of the codeword symbol associated with the variable node. Following the principle outlined in [82], the messages sent by check nodes to variable nodes are modeled as observations at the output a q-ary input, q-ary output discrete memoryless channel. By doing so, the messages at the input of each variable node can be combined by multiplying the respective likelihoods (or by summing the respective log-likelihoods), providing a simple update rule at the variable nodes.

Assume that we have a discrete memoryless channel over $\mathbb{Z}/q\mathbb{Z}$ with input $x \in \mathbb{Z}/q\mathbb{Z}$, output $w \in \mathbb{Z}/q\mathbb{Z}$ and channel law $P_{W\mid X}(w \mid x)$. We define the log-likelihood of w given x by $L_x(w) := \log\big(P_{W\mid X}(w \mid x)\big)$ and the log-likelihood vector by

$$L(w) := (L_0(w), L_1(w), \ldots, L_{q-1}(w)).$$

With a slight abuse of notation, we will use the $L(\cdot)$ for different channels, where the channel law to be applied is made clear by the argument.

1. **Initialization.** The decoder is initialized by forwarding the channel observation y to every variable node v. Then, the variable node v sends to each $\mathsf{c} \in \mathcal{N}(\mathsf{v})$

$$m_{\mathsf{v}\to\mathsf{c}} = y.$$

2. **Check node-to-variable node update.** Consider a given check node c and a neighboring variable node $\mathsf{v} \in \mathcal{N}(\mathsf{c})$. For the message $m_{\mathsf{c}\to\mathsf{v}}$, the check node computes

$$m_{\mathsf{c}\to\mathsf{v}} = h_{\mathsf{c},\mathsf{v}}^{-1} \sum_{\mathsf{v}'\in\mathcal{N}(\mathsf{c})\setminus\{\mathsf{v}\}} h_{\mathsf{c},\mathsf{v}'} m_{\mathsf{v}'\to\mathsf{c}}.$$

3. **Variable node-to-check node update.** At each variable node v, incoming messages are treated as observations of the codeword symbol at the output of an "extrinsic channel" ([8, 82]) with conditional probability

$$P_{M\mid X}(m \mid x) = \begin{cases} 1-\xi & \text{if } m = x \\ \xi/(q-1) & \text{otherwise} \end{cases}, \tag{6.2}$$

for a given error probability $\xi \in [0,1]$. For the calculation of the message to be sent of each check node $\mathsf{c} \in \mathcal{N}(\mathsf{v})$, (6.2) is used to compute the log-likelihood vector

$$E = L(y) + \sum_{\mathsf{c}'\in\mathcal{N}(\mathsf{v})\setminus\{\mathsf{c}\}} L\left(m_{\mathsf{c}'\to\mathsf{v}}\right). \tag{6.3}$$

For each $\mathsf{c} \in \mathcal{N}(\mathsf{v})$, the message sent by the variable node v is then

$$m_{\mathsf{v}\to\mathsf{c}} = \arg\max_{i\in\mathbb{Z}/q\mathbb{Z}} E_i.$$

4. **Final decision.** After at most $\ell_{\max}$ iterations for each variable node v we compute

$$L^{\mathsf{FIN}} = L(y) + \sum_{\mathsf{c}\in\mathcal{N}(\mathsf{v})} L\left(m_{\mathsf{c}\to\mathsf{v}}\right).$$

Then the final decision, $\hat{x}$, is the index of the maximal entry of L^{FIN}, i.e.,

$$\hat{x} = \arg\max_{i\in\mathbb{Z}/q\mathbb{Z}} L_i^{\mathsf{FIN}}.$$

Note that the extrinsic channel of (6.2) is modelled as a q-ary symmetric channel (q-SC) with error probability ξ. As it will be shown in Section 6.3.2, this choice yields an accurate description of the extrinsic channel conditional probability, despite its simplicity. The extrinsic channel parameter ξ is iteration-dependent. Its evaluation can be performed via Monte Carlo simulations, or by using estimates that follow from density evolution analysis [81, 82].

6.2 Average Weight Enumerator

The near-capacity performance of LDPC codes under iterative, low-complexity decoders is analyzed in terms of the distance spectrum of the code. In fact, knowing the distance distribution yields knowledge about the minimum distance of a code which, as discussed in Section 2.3.1, is in direct relation with the error-correction capability of a code. However, Vardy showed in [128] that for a given LDPC code characterizing its weight distribution (and hence the minimum distance) is a hard task. In his Ph.D. thesis [58], Gallager therefore considered an ensemble of regular LDPC codes and studied the average weight spectrum of the ensemble. Ever since, it is common practice to consider a code ensemble rather than focusing on a specific LDPC code. The weight spectrum of a code is used to derive bounds on the block error

probability over a given channel model. In the case of random LDPC code ensembles it allows to understand the error floor of the ensemble, and reasonable bounds on the error-correction performance of a code with given weight distribution can be derived [31, 59, 102].

In this section we provide an expression for the average Lee weight spectrum of a random regular LDPC code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ of dimension k. This task has been studied for the Hamming metric by [43]. However, we derive the average weight spectrum using a different approach using the type of the Lee weights of a vector together with the group action defined by the units $(\mathbb{Z}/q\mathbb{Z})^\times$ on $\mathbb{Z}/q\mathbb{Z}$.

For each possible Lee weight $\ell \in \{0, \dots, n\lfloor q/2 \rfloor\}$ we define the weight enumerator, i.e., the number of codewords of Lee weight ℓ as

$$W_\ell^{(n)} := |\{c \in \mathcal{C} \mid \mathrm{wt}_\mathsf{L}(c) = \ell\}|.$$

As the Lee weight of a vector is not identical to the number of nonzero positions but gives the nonzero entries a specific value, we are interested in the number of entries of a certain Lee weight in a codeword, i.e., we are interested in the type in terms of the Lee weight of the codeword (see Section 2.2 for a recap on types).

Definition 6.2.1. For every codeword $c \in \mathcal{C}$ we define its *Lee type* to be the $(\lfloor q/2 \rfloor + 1)$-tuple $\theta_c = (\theta_c(0), \dots, \theta_c(\lfloor q/2 \rfloor))$ consisting of the relative fraction of occurrences of each possible Lee weight $\ell \in \{0, \dots, \lfloor q/2 \rfloor\}$, i.e.,

$$\theta_c(\ell) = \frac{1}{n} |\{k = \{1, \dots, n\} \mid \mathrm{wt}_\mathsf{L}(c_k) = \ell\}|.$$

We denote the set of all Lee types over $(\mathbb{Z}/q\mathbb{Z})^n$ by $\mathcal{T}((\mathbb{Z}/q\mathbb{Z})^n)$. Then, we define the number of codewords in a code $\mathcal{C} \subseteq (\mathbb{Z}/q\mathbb{Z})^n$ of Lee type $\theta \in \mathcal{T}((\mathbb{Z}/q\mathbb{Z})^n)$ as

$$A_\theta^{(n)} := |\{c \in \mathcal{C} \mid \theta_c = \theta\}|.$$

Note that we can describe the Lee weight of a codeword $c \in \mathcal{C}$ in terms of its Lee type as

$$\mathrm{wt}_\mathsf{L}(c) = n \sum_{\ell=1}^{\lfloor q/2 \rfloor} \ell \theta_c(\ell).$$

Given a codeword c and its Lee type θ_c, by abuse of notation, we will use the notation $\mathrm{wt}_\mathsf{L}(\theta_c)$ to indicate the Lee weight of c. Thus, there is a natural relation between $W_\ell^{(n)}(\mathcal{C})$ and $A_\theta^{(n)}(\mathcal{C})$. In fact, we have

$$W_\ell^{(n)}(\mathcal{C}) = \sum_{\substack{\theta \in \mathcal{T}((\mathbb{Z}/q\mathbb{Z})^n) \\ \mathrm{wt}_\mathsf{L}(\theta) = \ell}} A_\theta^{(n)}(\mathcal{C}).$$

In the following, we consider a $(d_\mathsf{v}, d_\mathsf{c})$-regular LDPC code $\mathcal{C}$ of length n and $\mathbb{Z}/q\mathbb{Z}$-dimension k. taken uniformly at random from an ensemble of $(d_\mathsf{v}, d_\mathsf{c})$-regular LDPC codes over $\mathbb{Z}/q\mathbb{Z}$. Let H be a parity-check matrix of $\mathcal{C}$ where the nonzero entries of H lie in the set of units $(\mathbb{Z}/q\mathbb{Z})^\times$. As $\mathcal{C}$ is a random regular LDPC code, the parity-check matrix H is a random matrix where each row has d_c nonzero entries taken randomly among the unit elements and each column has d_v of them. In the following, we always consider a randomly chosen $c \in (\mathbb{Z}/q\mathbb{Z})^n$ and denote its Lee type by θ_c. Recall that c is a codeword if and only if $cH^\top = 0$.

We now briefly discuss what it means for a codeword c of a random LDPC code to satisfy the check equations of a parity-check matrix H. Considering the Tanner graph of a code $\mathcal{C}$, given a codeword c we start by repeating each position c_i exactly d_v times over the edges connected to the i-th variable node. We denote the resulting vector by $z' := (c_1, \dots, c_1, \dots, c_n, \dots, c_n)$. Note that z' is of length nd_v and is of Lee type $\theta_{z'} = \theta_c$. Let then $u \in ((\mathbb{Z}/q\mathbb{Z})^\times)^{nd_\mathsf{v}}$ be chosen uniformly at random, i.e., every entry u_i is chosen uniformly at random among the units $(\mathbb{Z}/q\mathbb{Z})^\times$. Finally, choosing a random permutation Π we compute

$z := \Pi(z' \odot u)$. Now, c satisfies $cH^\top = 0$ if and only if z satisfies the m check equations. Figure 6.2 below visualizes this procedure for a random $(d_\mathsf{v}, d_\mathsf{c})$-regular LDPC code.

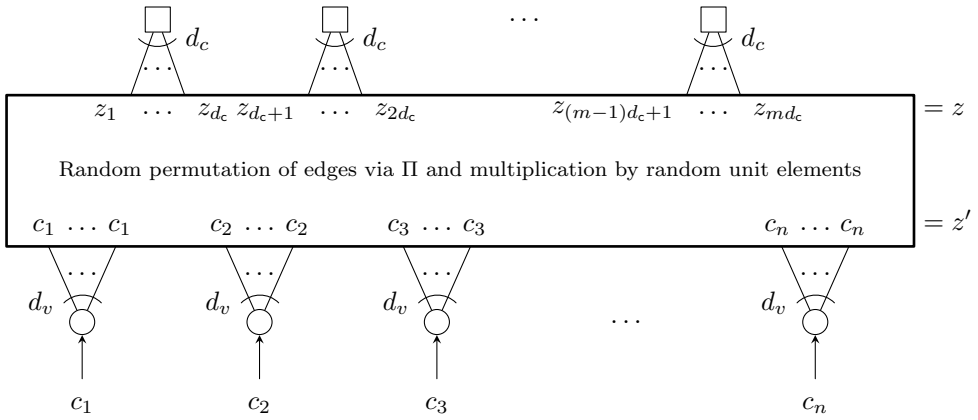

FIGURE 6.2: Graphical representation of a random $(d_\mathsf{v}, d_\mathsf{c})$-LDPC code of length n.

Note that $z_i = u_i z'_{\Pi(j)}$, where $u_i \in (\mathbb{Z}/q\mathbb{Z})^\times$ is chosen uniformly at random from the set of units modulo q. Having Figure 6.2 in mind, we can say that the average number of codewords of type $\theta \in \mathcal{T}\left((\mathbb{Z}/q\mathbb{Z})^n\right)$ of a random LDPC code is given by

$$\overline{A}_\theta^{(n)}(\mathcal{C}) = \binom{n}{n\theta} \mathbb{P}\left(z \text{ satisfies the check equations} \mid \theta_c = \theta\right).$$

We denote the Lee type of z by ω_z in order not to confuse it with the Lee type θ_c. Note that ω_z highly depends on θ_c. Further discussions and observations follow in Theorem 6.2.4. For now, let $\mathcal{T}_{\theta_c}\left((\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}\right)$ denote the set of all possible Lee types for a vector z resulting from the Lee type θ_c. Hence, we can further break down the conditional probability as

$$\overline{A}_\theta^{(n)}(\mathcal{C}) = \binom{n}{n\theta} \sum_{\omega \in \mathcal{T}_\theta\left((\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}\right)} \mathbb{P}\left(\omega_z = \omega \mid \theta_c = \theta\right) \mathbb{P}\left(z \text{ satisfies the check equations} \mid \omega_z = \omega\right). \tag{6.4}$$

In the following we elaborate more on the two probabilities, which we will denote by

$$f^{(n)}(\omega \mid \theta) := \mathbb{P}\left(\omega_z = \omega \mid \theta_c = \theta\right) \quad \text{and} \tag{6.5}$$

$$a^{(n)}(\omega) := \mathbb{P}\left(z \text{ satisfies the check equations} \mid \omega_z = \omega\right). \tag{6.6}$$

6.2.1 Transformation of the Lee Type

We start by analyzing how the Lee type of c changes to the Lee type of the vector z. More precisely, we now study the probability $f^{(n)}(\omega \mid \theta)$ that the vector z has a Lee type $\omega_z = \omega$ given that the Lee type of the codeword c is $\theta_c = \theta$. Recall that $z \in (\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}$ is formed from c by repeating the entries c_i each d_v times and then multiplying each copy by a randomly chosen unit. This already implies that the fraction of zeros in z must be equal to the fraction of zeros in c. Focusing on the nonzero entries of c we have to treat several cases separately, as the multiplication of a random nonzero element $x \in \mathbb{Z}/q\mathbb{Z}$ by a random unit $u \in (\mathbb{Z}/q\mathbb{Z})^\times$ lies in different orbits.

Note that the group of units $(\mathbb{Z}/q\mathbb{Z})^\times$ acts under multiplication on $\mathbb{Z}/q\mathbb{Z}$. For an element $a \in \mathbb{Z}/q\mathbb{Z}$ we define its orbit $\mathcal{O}_a$ as

$$\mathcal{O}_a := \left\{a \cdot u \mid u \in (\mathbb{Z}/q\mathbb{Z})^\times\right\}. \tag{6.7}$$

Orbits induce an equivalence relation, i.e., two elements are equivalent if and only if they lie within the same orbit. Each orbit can be represented by the smallest element in it which

corresponds exactly to a divisor of q. Let $\mathbb{D}_q$ denote the set of divisors of q, i.e.,

$$\mathbb{D}_q := \{\ell \in \mathbb{N} \mid \exists s \in \mathbb{N} \text{ with } \ell s = q\}.$$

Then the distinct orbits are given by $\mathcal{O}_d$ for $d \in \mathbb{D}_q$.

Example 6.2.2. We consider the integer residue ring $\mathbb{Z}/10\mathbb{Z}$. The set of divisors is given by

$$\mathbb{D}_{10} = \{1, 2, 5, 10\}.$$

Hence, there are four orbits defined by the divisors of ten, namely,

$$\mathcal{O}_1 = (\mathbb{Z}/10\mathbb{Z})^\times = \{1, 3, 7, 9\},\ \mathcal{O}_2 = \{2, 4, 6, 8\},\ \mathcal{O}_5 = \{5\} \text{ and } \mathcal{O}_0 := \mathcal{O}_{10} = \{0\}.$$

By the definition of an orbit in (6.7), we observe that if an element a lies in a given orbit $\mathcal{O}_d$ then every multiple of a by a unit element is in the same orbit. Hence, a codeword c and a vector z resulting from c have the same fraction of elements in an orbit $\mathcal{O}_d$ for every divisor $d \in \mathbb{D}_q$. For a codeword c with Lee type θ_c and for every $d \in \mathbb{D}_q$ the fraction of elements in orbit $\mathcal{O}_d$ is denoted as

$$\theta_c(\mathcal{O}_d) := \sum_{\substack{a \in \mathcal{O}_d \\ a \leq \lfloor q/2 \rfloor}} \theta_c(a). \tag{6.8}$$

The tuple of all such fractions is denoted by

$$\theta_{c,\mathcal{O}} := \left(\theta_c(\mathcal{O}_{d_1}), \ldots, \theta_c(\mathcal{O}_{d_{|\mathbb{D}_q|}})\right).$$

Regarding the Lee metric, we can prove that two elements of the same Lee weight are equivalent.

Lemma 6.2.3. *Elements of the same Lee weight in $\mathbb{Z}/q\mathbb{Z}$ lie in the same orbit, i.e., for every $a \in \mathbb{Z}/q\mathbb{Z}$ we have $\mathcal{O}_a = \mathcal{O}_{q-a}$.*

Proof. Let $a \in \mathbb{Z}/q\mathbb{Z}$. By symmetry of the Lee weight, $q - a$ is the only element having the same Lee weight as a. Let $b \in \mathcal{O}_{q-a}$ be arbitray. By the definition of an orbit (see Equation (6.7)), there exists a unit element $u \in (\mathbb{Z}/q\mathbb{Z})^\times$ such that $b \equiv u(q - a) \equiv -ua \mod q$. Since (-1) and u are units, also $(-u)$ is a unit modulo q and thus $b \in \mathcal{O}_a$. Since b was chosen arbitrarily, we have $\mathcal{O}_{q-a} = \mathcal{O}_a$. □

Lemma 6.2.3 indicates that we only have to consider elements up to $\lfloor q/2 \rfloor$. Recall from the generating function of the Lee weight of the elements in $\mathbb{Z}/q\mathbb{Z}$ (see (3.10)), if q is odd, then zero is the only element of Lee weight 0. All other weights in this case are represented by two elements. If instead q is even additionally the Lee weight $\lfloor q/2 \rfloor$ is represented only by one element, namely $\lfloor q/2 \rfloor$ itself. This fact is important when studying the number of configurations of a fixed Lee weight. Given the Lee type θ_x of a vector x we denote the fraction of Lee weights with only one representative element by

$$\widehat{\theta}_x := \begin{cases} 1 - \theta_x(0) & \text{if } q \text{ is odd,} \\ 1 - \theta_x(0) - \theta_x(\lfloor q/2 \rfloor) & \text{if } q \text{ is even.} \end{cases}$$

We are then able to state the result on the expression for the probability $f^{(n)}(\omega \mid \theta)$ over $\mathbb{Z}/q\mathbb{Z}$.

Theorem 6.2.4. *Consider a random $c \in (\mathbb{Z}/q\mathbb{Z})^n$ of Lee type θ_c. Let $z \in (\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}$ be the resulting vector when repeating the entries of c d_v times and multiplying each position by a randomly chosen unit element. Furthermore, we denote by ω_z the Lee type of z. Given the set of divisors $\mathbb{D}_q = \{d_1, \ldots, d_r\}$, then*

$$f^{(n)}(\omega_z \mid \theta_c) = \begin{cases} \dfrac{\binom{nd_\mathsf{v}}{nd_\mathsf{v}\omega_z} 2^{nd_\mathsf{v}\widehat{\omega_z}}}{\binom{nd_\mathsf{v}}{nd_\mathsf{v}\theta_{c,\mathcal{O}}} \prod_{d \in \mathbb{D}_q} |\mathcal{O}_d|^{nd_\mathsf{v}\theta_c(\mathcal{O}_d)}} & \text{if } \omega_z \in \mathcal{T}_{\theta_c}\left((\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}\right), \\ 0 & \text{otherwise,} \end{cases} \tag{6.9}$$

where $\mathcal{T}_{\theta_c}\left((\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}\right) := \left\{\omega \in \mathcal{T}((\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}) \mid \omega(\mathcal{O}_d) = \theta_c(\mathcal{O}_d)\,\forall d \in \mathbb{D}_q\right\}$.

Proof. Assume the Lee type θ_c of c is given by θ and let the Lee type ω_z be equal to ω. By the above discussion, when multiplying an element a of a given orbit $\mathcal{O}_d$ with a randomly chosen unit $u \in (\mathbb{Z}/d\mathbb{Z})^\times$, the product is still an element of $\mathcal{O}_d$. In fact, au can take each element of $\mathcal{O}_d$ with the same probability. Therefore, z must have the same fraction of elements in orbit $\mathcal{O}_d$ as the codeword c which also yields, that $f^{(n)}(\omega \mid \theta) = 0$ if this is not fulfilled.

Let us assume then that for every divisor d of q it holds that $\omega(\mathcal{O}_d) = \theta(\mathcal{O}_d)$. The probability that $\omega_z = \omega$ given that $\theta_c = \theta$ is given by the number of vectors of length nd_v over $\mathbb{Z}/q\mathbb{Z}$ of Lee type ω divided by the total number of vectors of a Lee types fulfilling the constraint on the fraction of orbit elements. The number of configurations of vectors with Lee type ω is given by the multinomial coefficient

$$\binom{nd_\mathsf{v}}{nd_\mathsf{v}\omega} = \binom{nd_\mathsf{v}}{nd_\mathsf{v}\omega(0), \ldots, nd_\mathsf{v}\omega(\lfloor q/2 \rfloor)}.$$

Since the Lee type gives rise only to the number of elements of a certain Lee weight, we must consider Lee weights reached by two different elements. We hence have to multiply the multinomial coefficient by a power of 2 considering the two options for Lee weights admitting two representative elements given by $2^{nd_\mathsf{v}\widehat{\omega}}$. This yields us the numerator of the probability $f^{(n)}(\omega \mid \theta)$ and hence the number of vectors $v \in (\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}$ of Lee type ω.

We are now interested in finding the number of vectors $v \in (\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}$ of Lee type ω_v, satisfying $\omega_{v,\mathcal{O}}(\mathcal{O}_d) = \theta_\mathcal{O}$. This number splits into two quantities: first, focusing only on the orbits, the number of constellation of the orbits, and second the number of choices in each orbit. The first quantity is again given by a multinomial coefficient regarding the fraction of elements in orbit $\mathcal{O}_d$ for every $d \in \mathbb{D}_q$ given in (6.8). To obtain the latter quantity we raise the cardinality of the orbit $\mathcal{O}_d$ to the power of the number of positions with elements in that orbit. Combining the results yields the denominator and hence, the desired result on the probability $f^{(n)}(\omega \mid \theta)$. □

Note that if q is a prime number, there are only two orbits; one containing only the zero element, and one corresponding to the set of units modulo q (which are all nonzero elements). Then the expression in Theorem 6.2.4 simplifies to

$$f^{(n)}(\omega \mid \theta) = \begin{cases} \frac{2^{nd_\mathsf{v}\widehat{\omega}}}{(q-1)^{nd_\mathsf{v}(1-\theta(0))}} & \text{if } \omega(0) = \theta(0) \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, there is a closed form for the cardinalities of the orbits which allow for a simple implementation of the formula given in Theorem 6.2.4.

Lemma 6.2.5. *Let q be a positive integer and let $\mathbb{D}_q$ be the set of divisors of q. Furthermore, let $\varphi(q)$ denote the Euler totient function. Then, for every $d \in \mathbb{D}_q$ the cardinality of its orbit is given by*

$$|\,\mathcal{O}_d\,| = \begin{cases} 1 & \text{if } d = q, \\ \varphi(q)/d & \text{otherwise.} \end{cases}$$

Proof. The case where $d = q$ is obvious by the definition of the orbit given in (6.7). Hence, consider $d \in \mathbb{D}_q \setminus \{q\}$. Recall that the orbit $\mathcal{O}_d$ consists of all multiplications of d with a unit element, i.e., $\mathcal{O}_d = \{d \cdot u \mid u \in (\mathbb{Z}/q\mathbb{Z})^\times\}$. Thus, there are $\varphi(q)$ multiplications by d. As the group $\mathbb{Z}/q\mathbb{Z}$ is cyclic, these multiplications repeat exactly d times. Hence, $\mathcal{O}_d = \varphi(q)/d$. □

Using Lemma 6.2.5 the formula in Theorem 6.2.4 becomes

$$f^{(n)}(\omega_z \mid \theta_c) = \begin{cases} \frac{\binom{nd_\mathsf{v}}{nd_\mathsf{v}\omega_z} 2^{nd_\mathsf{v}\widehat{\omega_z}}}{\binom{nd_\mathsf{v}}{nd_\mathsf{v}\theta_{c,\mathcal{O}}} \left(\varphi(q)^{(1-\theta_c(\mathcal{O}_0))} \prod_{d \in \mathbb{D}_q \setminus \{q\}} d^{-\theta_c(\mathcal{O}_d)}\right)^{nd_\mathsf{v}}} & \text{if } \omega_z \in \mathcal{T}_{\theta_c}\left((\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}\right), \\ 0 & \text{otherwise.} \end{cases}$$

Example 6.2.6. To illustrate (6.9) presented in Theorem 6.2.4 consider the following example over $\mathbb{Z}/16\mathbb{Z}$. Note that $\mathbb{Z}/16\mathbb{Z}$ consists of the following five orbits:

$$\mathcal{O}_{16} = \{0\}\,,\ \mathcal{O}_1 = (\mathbb{Z}/16\mathbb{Z})^\times,\ \mathcal{O}_2 = \{2,6,10,14\}\,,\ \mathcal{O}_4 = \{4,12\}\ \text{and}\ \mathcal{O}_8 = \{8\}\,.$$

Let $\mathcal{C} \subset (\mathbb{Z}/16\mathbb{Z})^2$ be a regular code with regular variable node degree $d_\mathsf{v} = 2$. Let $c \in \mathcal{C}$ be a codeword of Lee type $\theta_c = (0,0,1/2,0,1/2,0,0,0,0)$. Without loss of generality, we can assume that $c = (2,4)$. Following the procedure described by Figure 6.2 yields

$$z' = (2,2,4,4).$$

When multiplying each of the entries by a randomly chosen unit, we observe that z can be one of the following vectors (up to permutation and Lee weight)

$$(2,2,4,4),\ (2,6,4,4),\ \text{and}\ (6,6,4,4).$$

Hence, the possible types for z are

$$\begin{aligned}\omega^{(1)} &= (0,0,1/2,0,1/2,0,0,0,0),\\ \omega^{(2)} &= (0,0,1/4,0,1/2,0,1/4,0,0)\ \text{and}\\ \omega^{(3)} &= (0,0,0,0,1/2,0,1/2,0,0).\end{aligned}$$

The number of permutations for each case is given by the multinomial coefficient with respect to the type $\omega^{(i)}$. For instance, the vector $(2,2,4,4)$ admits 6 permutations, i.e.,

$$\binom{nd_\mathsf{v}}{nd_\mathsf{v}\omega^{(1)}(0),\ldots,nd_\mathsf{v}\omega^{(1)}(8)} = \binom{2\cdot 2}{2\cdot 2\cdot(1/2),2\cdot 2\cdot(1/2)} = \frac{4!}{2!2!} = 6.$$

Since the Lee type focuses on the Lee weight only and since every nonzero entry different from $\lfloor q/2\rfloor$ admits two representatives, we have two possible entries for each position. In the case of type $\omega^{(1)}$ we would, hence, have $6\cdot 16 = 96$ possible vectors of that type. Similarly, we have 96 vectors of type $\omega^{(3)}$ and 192 vectors of type $\omega^{(2)}$. This yields a total of 384 vectors. Note that this indeed coincides with

$$\binom{nd_\mathsf{v}}{nd_\mathsf{v}\theta_c(\mathcal{O}_1),\ldots,nd_\mathsf{v}\theta_c(\mathcal{O}_{16})}\prod_{d\in\mathbb{D}_q}|\,\mathcal{O}_d\,|^{nd_\mathsf{v}\theta_c(\mathcal{O}_d)} = \binom{4}{2}|\,\mathcal{O}_2\,|^2\,|\,\mathcal{O}_4\,|^2 = 384.$$

Thus, the probability that z has Lee type $\omega^{(1)}$ given that the Lee type of the codeword c is θ_c is $f^{(n)}(\omega^{(1)}\mid\theta_c) = \frac{96}{384} = \frac{1}{4}$.

Consequently, to Theorem 6.2.4 we determine the asymptotic growth rate of $f^{(n)}(\omega\mid\theta)$ in Corollary 6.2.7

Corollary 6.2.7. *Let $z \in (\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}$ be the vector resulting from a vector $c \in (\mathbb{Z}/q\mathbb{Z})^n$ of Lee type θ after repetition and permutation. Then we obtain the following asymptotic expression for the probability that z is of Lee type ω.*

$$\phi(\omega\mid\theta) := \lim_{n\longrightarrow\infty}\frac{1}{n}\log(f^{(n)}(\omega\mid\theta)) = d_\mathsf{v}\left(H(\omega)+\widehat{\omega}-H(\theta_\mathcal{O})-\sum_{d\in\mathbb{D}_q}\theta(\mathcal{O}_d)\log(|\,\mathcal{O}_d\,|)\right).$$

Proof. The proof follows by taking the limit of each summand. □

Moreover, Lemma 6.2.8 shows us an even stronger form of convergence.

Lemma 6.2.8. *Given a random regular $(d_\mathsf{v},d_\mathsf{c})$ LDPC code over $\mathbb{Z}/q\mathbb{Z}$. Given a Lee type $\theta \in \mathcal{T}((\mathbb{Z}/q\mathbb{Z})^n)$ and the sequence $f^{(n)}(\omega\mid\theta)$ defined in (6.9) with $\omega \in \mathcal{T}_\theta\left((\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}\right)$. Then the sequence of functions $\left(\frac{1}{n}\log(f^{(n)}(\omega\mid\theta))\right)_{n\in\mathbb{N}}$ is uniformly convergent to $\phi(\omega\mid\theta)$ as $n\longrightarrow\infty$.*

Proof. We have to show that for every $\varepsilon > 0$ there is a natural number $n_\varepsilon \in \mathbb{N}$ such that for all $n \geq n_\varepsilon$ it holds

$$\left| \frac{1}{n} \log(f^{(n)}(\omega \mid \theta)) - \phi(\omega \mid \theta) \right| < \varepsilon.$$

Applying Theorem 6.2.4 and Corollary 6.2.7, and by using the triangle inequality, we get

$$\begin{aligned}
&\left| \frac{1}{n} \log\left(f^{(n)}(\omega \mid \theta)\right) - \phi(\omega \mid \theta) \right| \\
&\qquad = \left| \frac{1}{n} \log\left(\!\!\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\omega}\!\!\right) - d_{\mathsf{v}} H(\omega) - \frac{1}{n} \log\left(\!\!\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\theta_{\mathcal{O}}}\!\!\right) + d_{\mathsf{v}} H(\theta_{\mathcal{O}}) \right| \\
&\qquad \leq \left| \frac{1}{n} \log\left(\!\!\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\omega}\!\!\right) - d_{\mathsf{v}} H(\omega) \right| + \left| d_{\mathsf{v}} H(\theta_{\mathcal{O}}) - \frac{1}{n} \log\left(\!\!\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\theta_{\mathcal{O}}}\!\!\right) \right|.
\end{aligned}$$

Let us focus now on $\left| \frac{1}{n} \log\left(\!\!\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\omega}\!\!\right) - d_{\mathsf{v}} H(\omega) \right|$. Recall from (2.4) that we have the following bounds on $\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\omega}$:

$$\frac{1}{(nd_{\mathsf{v}}+1)^{\lfloor q/2 \rfloor + 1}} 2^{nd_{\mathsf{v}} H(\omega)} \leq \binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\omega} \leq 2^{nd_{\mathsf{v}} H(\omega)}.$$

Hence, if $\frac{1}{n} \log\left(\!\!\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\omega}\!\!\right) > d_{\mathsf{v}} H(\omega)$, we get

$$\left| \frac{1}{n} \log\left(\!\!\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\omega}\!\!\right) - d_{\mathsf{v}} H(\omega) \right| = 0.$$

On the other hand, we obtain

$$\left| \frac{1}{n} \log\left(\!\!\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\omega}\!\!\right) - d_{\mathsf{v}} H(\omega) \right| \leq (\lfloor q/2 \rfloor + 1) \frac{1}{n} \log(nd_{\mathsf{v}} + 1).$$

By l'Hôpital's rule this converges to zero as $n \longrightarrow \infty$.

Note that the same argument holds for $\left| d_{\mathsf{v}} H(\theta_{\mathcal{O}}) - \frac{1}{n} \log\left(\!\!\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\theta_{\mathcal{O}}}\!\!\right) \right|$ and thus the result follows. □

6.2.2 Valid Check Node Assignment

We now discuss the probability $a^{(n)}(\omega)$ given in (6.6). We make use of generating functions to describe the situation at one check node and then extend the generating function to m check nodes. In the following let w denote the Lee weight decomposition of a vector $x \in (\mathbb{Z}/q\mathbb{Z})^n$. That is, for every $i = 0, \ldots, \lfloor q/2 \rfloor$,

$$w_i = |\{k = 1, \ldots, n \mid \mathrm{wt}_{\mathsf{L}}(x_k) = i\}|.$$

Furthermore, recall from Equation (6.9) in Theorem 6.2.4 that given a type θ of c, the type ω of a valid check node assignment has to show the same orbit distribution. Hence, there is a restricted choice.

Theorem 6.2.9. *Consider a vector $z \in (\mathbb{Z}/q\mathbb{Z})^{nd_{\mathsf{v}}}$ of Lee type ω and weight decomposition w. Furthermore, consider a random regular LDPC code of variable degree d_{v} and check node degree d_{c}. Then, the probability that z fulfills the check node equations is given by*

$$a^{(n)}(\omega) = \frac{\mathrm{coeff}\left[G(t), t^{\omega n d_{\mathsf{v}}}\right]}{\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\omega}},$$

where

$$G(t) = \frac{1}{q^m} \left[\sum_{\substack{z_i \in (\mathbb{Z}/q\mathbb{Z})^{d_\mathsf{c}} \\ d_\mathsf{c}\omega_{z_i} = w}} \sum_{s=0}^{q-1} \prod_{k=1}^{d_\mathsf{c}} \mathrm{e}^{\frac{2\pi i}{q} s z_k} t_1^{n d_\mathsf{v} \omega(1)} \cdots t_{\lfloor q/2 \rfloor}^{n d_\mathsf{v} \omega(\lfloor q/2 \rfloor)} \right]^m . \tag{6.10}$$

Proof. Recall that $a^{(n)}(\omega)$ describes the probability of $z \in (\mathbb{Z}/q\mathbb{Z})^{nd_\mathsf{v}}$ satisfying the check node equations and being of a given Lee type ω. Furthermore, we have m check nodes each of degree d_c. Hence, we can split z into m parts $z_1, \ldots, z_m$ each one corresponding check node $\mathsf{c}_1, \ldots, \mathsf{c}_m$, respectively.

We now focus on one check node only and describe a generating function for the number of z_i's satisfying the check node equations of check node c_i and having Lee weight decomposition given by $w = (w_0, \ldots, w_{\lfloor q/2 \rfloor})$. We turn our attention at this point only to the nonzero elements and note that $w_0 = d_\mathsf{c} - \sum_{i=1}^{\lfloor q/2 \rfloor} w_i$. In that sense, let us define

$$g_{(w_1,\ldots,w_{\lfloor q/2 \rfloor})} := \Big| \Big\{ z_i \in (\mathbb{Z}/q\mathbb{Z})^{d_\mathsf{c}} \mid z_i \text{ satisfies the check-equation and } \big| \{ j = 1, \ldots, d_\mathsf{c} \mid \mathrm{wt}_\mathsf{L}(z_{i_j}) = k \} \big| = w_k \Big\} \Big|.$$

We can describe this quantity summing over all d_c-tuples that sum up to zero using an indicator function, that is for a given statement Σ the function $\mathbb{1}(\Sigma)$ is one whenever the statement Σ is fulfilled, and zero otherwise. Indeed,

$$g_{(w_1,\ldots,w_{\lfloor q/2 \rfloor})} = \sum_{\substack{z_i \in (\mathbb{Z}/q\mathbb{Z})^{d_\mathsf{c}} \\ d_\mathsf{c}\omega_{z_i} = w}} \mathbb{1} \left(\sum_{k=1}^{d_\mathsf{c}} z_k = 0 \right).$$

Applying the inversion formula for the discrete Fourier transform over $\mathbb{Z}/q\mathbb{Z}$ yields

$$g_{(w_1,\ldots,w_{\lfloor q/2 \rfloor})} = \sum_{\substack{z_i \in (\mathbb{Z}/q\mathbb{Z})^{d_\mathsf{c}} \\ d_\mathsf{c}\omega_{z_i} = w}} \frac{1}{q} \sum_{\chi \text{ character}} \chi \left(\sum_{k=1}^{d_\mathsf{c}} z_k \right). \tag{6.11}$$

Over the finite Abelian group $\mathbb{Z}/q\mathbb{Z}$ there are q characters $\chi_0, \ldots, \chi_{q-1}$ defined by $\chi_k(a) := \mathrm{e}^{\frac{2\pi i}{q} k a}$ for each element $a \in \mathbb{Z}/q\mathbb{Z}$. Hence, we can rewrite (6.11) as

$$g_{(w_1,\ldots,w_{\lfloor q/2 \rfloor})} = \frac{1}{q} \sum_{\substack{z_i \in (\mathbb{Z}/q\mathbb{Z})^{d_\mathsf{c}} \\ d_\mathsf{c}\omega_{z_i} = w}} \sum_{s=0}^{q-1} \mathrm{e}^{\frac{2\pi i}{q} s \sum_{k=1}^{d_\mathsf{c}} z_k}. \tag{6.12}$$

We then define the generating function $g(t)$ by

$$g(t) := \sum_{\substack{w \text{ composition} \\ \text{of } d_\mathsf{c}}} g_{(w_1,\ldots,w_{\lfloor q/2 \rfloor})} t_1^{w_1} \cdots t_{\lfloor q/2 \rfloor}^{w_{\lfloor q/2 \rfloor}}.$$

To obtain a similar expression for a configuration regarding all the check nodes, we take the m-fold convolution of $g_{(w_1,\ldots,w_{\lfloor q/2 \rfloor})}$, i.e.,

$$G_{(w_1,\ldots,w_{\lfloor q/2 \rfloor})} := g_{(w_1,\ldots,w_{\lfloor q/2 \rfloor})} \circledast \cdots \circledast g_{(w_1,\ldots,w_{\lfloor q/2 \rfloor})}.$$

Hence, the corresponding generating function for m check nodes is

$$G(t) := \sum_{\substack{w \text{ composition} \\ \text{of } m d_\mathsf{c}}} g_{(w_1,\ldots,w_{\lfloor q/2 \rfloor})} t_1^{w_1} \cdots t_{\lfloor q/2 \rfloor}^{w_{\lfloor q/2 \rfloor}} = g(t)^m.$$

Let ω denote the type of the decomposition w, i.e., $nd_{\mathsf{v}}\omega(i) = w_i$ for every $i \in \{0, \dots, \lfloor q/2 \rfloor\}$. The number of configurations of given Lee type ω is then the coefficient of the polynomial $G(t)$ at $t^{nd_{\mathsf{v}}\omega} = t_1^{w_1} \cdots t_{\lfloor q/2 \rfloor}^{w_{\lfloor q/2 \rfloor}}$. Finally, the probability $a^{(n)}(\omega)$ is obtained by dividing the $(nd_{\mathsf{v}}\omega)$-th coefficient of $G(t)$ by all the possible permutations of a vector $x \in (\mathbb{Z}/q\mathbb{Z})^{nd_{\mathsf{v}}}$ of Lee type ω, which is given by the multinomial coefficient $\binom{nd_{\mathsf{v}}}{nd_{\mathsf{v}}\omega}$. □

At this point, to simplify the understanding, we would like to discuss the expression in (6.12) with an example.

Example 6.2.10. Assume the check node degree is $d_{\mathsf{c}} = 2$ and that the underlying integer ring is $\mathbb{Z}/5\mathbb{Z}$. Let us furthermore assume that the Lee weight decomposition of a tuple z_i at a check node is $w = (0, 2, 0)$. This means that z_i is one of the following tuples

$$(1,1), \quad (1,4), \quad (4,1), \quad \text{or} \quad (4,4).$$

Since only $(1, 4)$ and $(4, 1)$ satisfy the check equation (i.e., sum up to zero modulo five), the enumerator $g_{(0,2,0)}$ should equal two. In fact, the exponential expression in Equation (6.12) equals 1 for all tuples satisfying the check equation. For those not satisfying the check equation the sum of exponentials is the sum of n-th roots of unity (in our case $n = 5$) and is hence equal to zero.

Let us now focus on the asymptotic growth rate of $a^{(n)}$ which we define as

$$\alpha(\omega) := \lim_{n \longrightarrow \infty} \frac{1}{n} \log(a^{(n)}(\omega)).$$

A direct consequence of taking the logarithm and the limit of the sequence $a^{(n)}$ is captured in Corollary 6.2.11.

Corollary 6.2.11. *Let $z \in (\mathbb{Z}/q\mathbb{Z})^{nd_{\mathsf{v}}}$ satisfy the m check equations and denote by ω its Lee type. Then we obtain the following asymptotic expression for the probability $a^{(n)}(\omega)$.*

$$\alpha(\omega) = -d_{\mathsf{v}} H(\omega) + (1 - R) \inf_{t \succ 0} \log \left(\frac{g(t)}{t^{\omega n d_{\mathsf{v}}}} \right),$$

where $t \succ 0$ means that not every entry of $t = (t_1, \dots, t_{\lfloor q/2 \rfloor})$ is equal to zero.

Taking the infimum over all possibilities of $t = (t_1, \dots, t_{\lfloor q/2 \rfloor})$ is impractical. We will use the asymptotic method of Hayman for multivariate polynomials (see [49, 72, 134]) to establish $\lim_{n \longrightarrow \infty} 1/n \log \left(\text{coeff} \left[G(t), t^{\omega n d_{\mathsf{v}}} \right] \right)$.

Lemma 6.2.12 (Hayman Formula)**.** *Let $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ and let $p(x)$ be a multivariate polynomial with $p(0) \neq 0$. Furthermore, let $\beta = (\beta_1, \dots, \beta_d)$ such that $0 \leq \beta_i \leq 1$ and $\beta_i n \in \mathbb{N}$ for all $i = 1, \dots, d$. Assume that $x^\star = (x_1^\star, \dots, x_d^\star)$ is the unique positive real solution to the system of equations given by*

$$x_1 \frac{\partial p(x)}{\partial x_1} = \beta_1 p(x), \ \dots, \ x_d \frac{\partial p(x)}{\partial x_d} = \beta_d p(x).$$

Then, as $n \longrightarrow \infty$, it holds

$$\lim_{n \longrightarrow \infty} \frac{1}{n} \ln \left(\text{coeff} \left[(p(z))^n, z^{n\beta} \right] \right) = \left(\ln(p(x)) - \sum_{i=1}^{d} \beta_i \ln(x_i) \right).$$

In our case, we have that

$$\lim_{n \longrightarrow \infty} \frac{1}{n} \ln \left(\text{coeff} \left[(g(t)^{1/d_{\mathsf{c}}})^{nd_{\mathsf{v}}}, t^{\omega n d_{\mathsf{v}}} \right] \right) = d_{\mathsf{v}} \lim_{n' \longrightarrow \infty} \frac{1}{n'} \ln \left(\text{coeff} \left[(g(t)^{1/d_{\mathsf{c}}})^{n'}, t^{\omega n'} \right] \right).$$

Hence, Corollary 6.2.13 is a direct consequence of Hayman's Formula.

Corollary 6.2.13. *Let* $\omega = (\omega(0), \ldots, \omega(\lfloor q/2 \rfloor)) \in [0,1]^{\lfloor q/2 \rfloor + 1}$ *such that* $\omega(i) n d_{\mathsf{v}} \in \mathbb{N}$ *for every* $i = 1, \ldots, d$. *Then*

$$\alpha(\omega) = d_{\mathsf{v}} \left(H(\omega) + \ln\left(g(t^\star)^{1/d_{\mathsf{c}}} \right) - \sum_{i=1}^{\lfloor q/2 \rfloor} \omega(i) \ln(t_i^\star) \right),$$

where $t^\star = (t_1^\star, \ldots, t_{\lfloor q/2 \rfloor}^\star)$ *is the unique positive real solution to the equations*

$$t_i \frac{\partial g(t)^{1/d_{\mathsf{c}}}}{\partial t_i} = \omega(i) g(t)^{1/d_{\mathsf{c}}}, \quad i = 1, \ldots, \lfloor q/2 \rfloor .$$

In his paper, Hayman gave an explicit expression for the coefficient of an admissible function (see [72, p. 69]). Hence, it easily follows that the sequence of functions $\left(\frac{1}{n}\log(a^{(n)})\right)_{n\in\mathbb{N}}$ is uniformly convergent.

6.2.3 Asymptotic Growth Rate

Having determined the two probabilities defined in Equations (6.5) and (6.6), respectively, the expression for the average type enumerator $\overline{A}_\theta^{(n)}(\mathcal{C})$ follows immediately. We can then deduce immediately the asymptotics of the average type enumerator and average weight enumerator, respectively.

Corollary 6.2.14. *Let* $\mathcal{C}$ *be a random* $(d_{\mathsf{v}}, d_{\mathsf{c}})$*-regular LDPC code of length* n *over* $\mathbb{Z}/q\mathbb{Z}$. *Let* $\mathcal{A}(\theta) := \lim_{n\longrightarrow\infty} \frac{1}{n}\log(\overline{A}_\theta^{(n)}(\mathcal{C}))$ *and* $\mathcal{W}(\ell) := \lim_{n\longrightarrow\infty} \frac{1}{n}\log(\overline{W_\ell}^{(n)}(\mathcal{C}))$ *be spectral growth rate of the average Lee type enumerator and weight enumerator, respectively. Then*

$$\begin{aligned} \mathcal{A}(\theta) &\leq H(\theta) + \sup_{\omega \in \mathcal{T}_\theta((\mathbb{Z}/q\mathbb{Z})^{n d_{\mathsf{v}}})} \left(\phi(\omega \mid \theta) + \alpha(\omega)\right), \quad \text{and} \\ \mathcal{W}(\ell) &\leq \sup_{\theta \in \mathcal{T}((\mathbb{Z}/q\mathbb{Z})^n) \,:\, \mathrm{wt}_{\mathsf{L}}(\theta) = \ell} \mathcal{A}(\theta). \end{aligned} \tag{6.13}$$

Proof. From Equation (6.4) we observe that

$$\overline{A}v_\theta^{(n)}(\mathcal{C}) = \binom{n}{n\theta} \sum_{\omega \in \mathcal{T}_\theta((\mathbb{Z}/q\mathbb{Z})^{n d_{\mathsf{v}}})} f^{(n)}(\omega \mid \theta) a^{(n)}(\omega).$$

Hence, we have

$$\begin{aligned} \mathcal{A}(\theta) &= H(\theta) + \lim_{n\longrightarrow\infty} \frac{1}{n} \log\left(\sum_{\omega \in \mathcal{T}_\theta((\mathbb{Z}/q\mathbb{Z})^{n d_{\mathsf{v}}})} f^{(n)}(\omega \mid \theta) a^{(n)}(\omega) \right) \\ &\leq H(\theta) + \lim_{n\longrightarrow\infty} \frac{1}{n} \log\left(\sup_{\omega \in \mathcal{T}_\theta((\mathbb{Z}/q\mathbb{Z})^{n d_{\mathsf{v}}})} \left[f^{(n)}(\omega \mid \theta) a^{(n)}(\omega) \right] \left| \, \mathcal{T}_\theta\left((\mathbb{Z}/q\mathbb{Z})^{n d_{\mathsf{v}}}\right) \right| \right) \\ &\overset{(a)}{=} H(\theta) + \lim_{n\longrightarrow\infty} \sup_{\omega \in \mathcal{T}_\theta((\mathbb{Z}/q\mathbb{Z})^{n d_{\mathsf{v}}})} \left[\frac{1}{n} \log\left(f^{(n)}(\omega \mid \theta) a^{(n)}(\omega) \right) \right] \\ &= H(\theta) + \lim_{n\longrightarrow\infty} \sup_{\omega \in \mathcal{T}_\theta((\mathbb{Z}/q\mathbb{Z})^{n d_{\mathsf{v}}})} \left[\frac{1}{n} \log\left(f^{(n)}(\omega \mid \theta) \right) + \frac{1}{n} \log\left(a^{(n)}(\omega) \right) \right], \end{aligned}$$

where for (a) we used, that $\left| \, \mathcal{T}_\theta\left((\mathbb{Z}/q\mathbb{Z})^{n d_{\mathsf{v}}}\right) \right|$ is polynomial in n. By the uniform convergence shown in Lemma 6.2.8 and in [72], we can switch the limit with the supremum and the statement follows. The bound in (6.13) for $\mathcal{W}(\ell)$ follows in an analogous manner. □

Figures 6.3, 6.4 and 6.5 show the spectral growth rate of the average weight enumerator of a random regular $(3, 6)$ LDPC code over $\mathbb{Z}/2\mathbb{Z}$, $\mathbb{Z}/3\mathbb{Z}$ and $\mathbb{Z}/4\mathbb{Z}$, respectively. Furthermore, it shows on the little frame the expected smallest weight of a codeword in the respective LDPC ensemble.

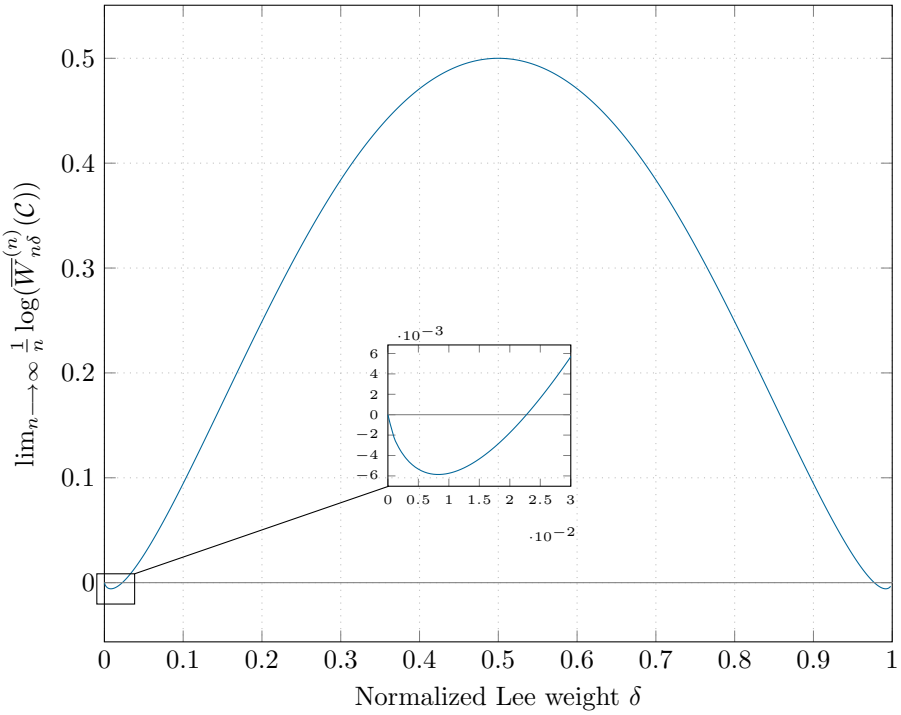

FIGURE 6.3: Spectral growth rate of the average weight enumerator of a regular $(3, 6)$ LDPC code ensembles over $\mathbb{Z}/2\mathbb{Z}$.

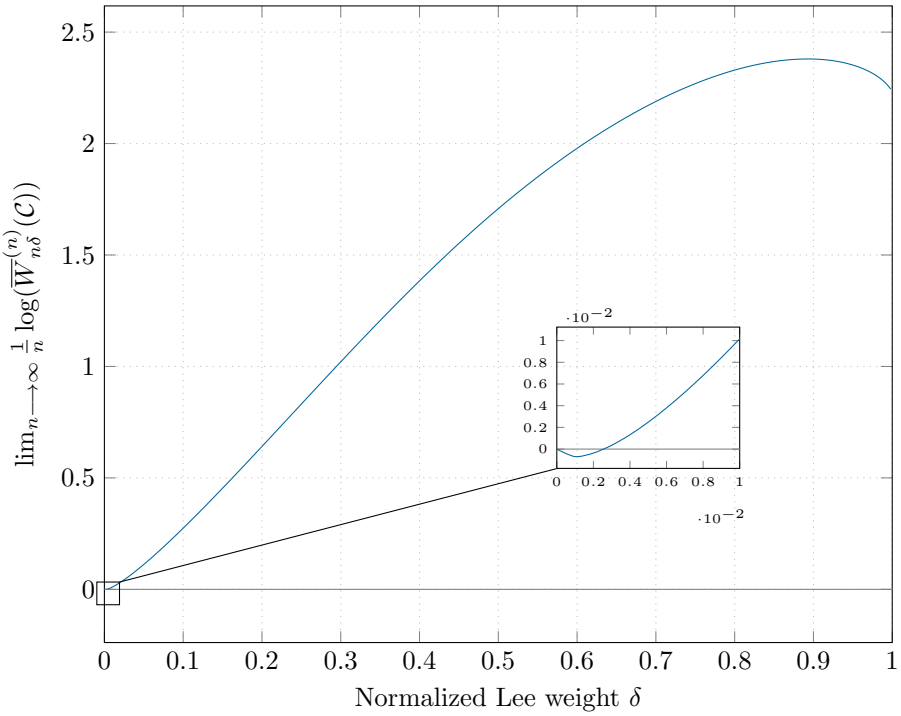

FIGURE 6.4: Spectral growth rate of the average weight enumerator of a regular $(3, 6)$ LDPC code ensembles over $\mathbb{Z}/3\mathbb{Z}$.

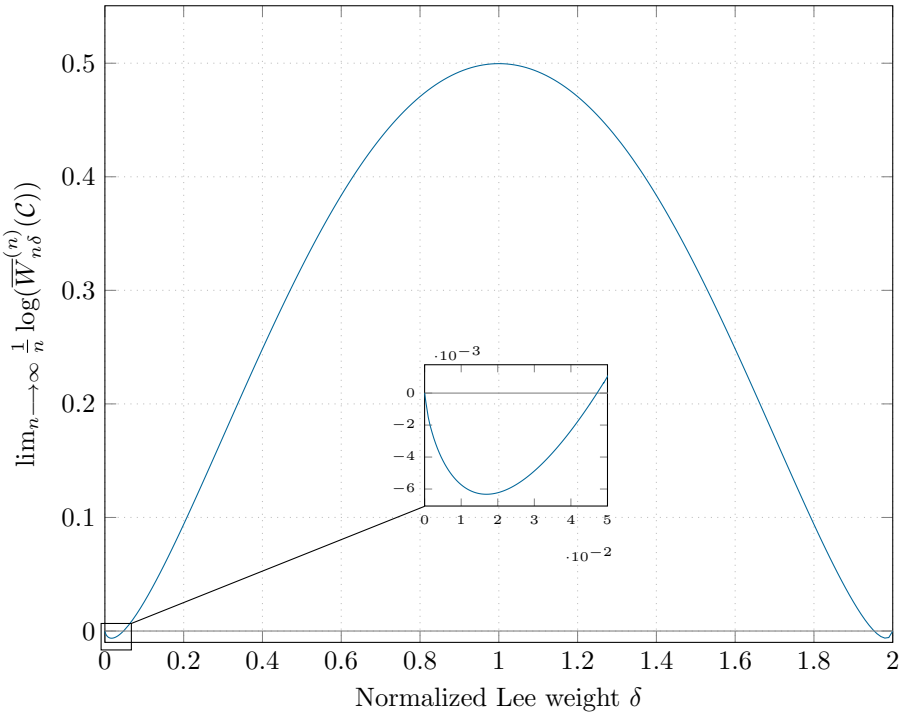

FIGURE 6.5: Spectral growth rate of the average weight enumerator of a regular $(3, 6)$ LDPC code ensembles over $\mathbb{Z}/4\mathbb{Z}$.

The average weight spectrum of LDPC codes has been studied mainly over the binary field $\mathbb{F}_2$. It has been shown [58] that for a random regular $(d_\mathsf{v}, d_\mathsf{c})$ LDPC code $\mathcal{C}$ over $\mathbb{F}_2$ its average weight spectrum has the following form

$$(1-d_\mathsf{v})H(\omega) + d_\mathsf{v} \lim_{n\longrightarrow\infty} \frac{1}{n} \log\left(\text{coeff}\left[\left(\frac{(1+x)^{d_\mathsf{c}} + (1-x)^{d_\mathsf{c}}}{2}\right)^{nd_\mathsf{v}/d_\mathsf{c}}, x^{nd_\mathsf{v}\omega}\right]\right). \quad (6.14)$$

In fact, this form corresponds to the expression in (6.13) which we can rewrite as

$$\sup_{\theta\in\mathcal{T}((\mathbb{Z}/q\mathbb{Z})^n)} \left(H(\theta) - d_\mathsf{v}\left[H(\theta_\mathcal{O}) - \sum_{d\in\mathbb{D}_q} \theta(\mathcal{O}_d)\log(|\,\mathcal{O}_d\,|)\right] + d_\mathsf{v}\left[\widehat{\omega} + \lim_{n\longrightarrow\infty}\frac{1}{n}\log\left(\text{coeff}\left[G(t), t^{nd_\mathsf{v}\omega}\right]\right)\right]\right). \quad (6.15)$$

It is easy to verify that, for $q = 2$, the generating function $G(x)$ given in Equation (6.10), coincides with the polynomial

$$\bar{G}(t) := \left(\frac{(1+x)^{d_\mathsf{c}} + (1-x)^{d_\mathsf{c}}}{2}\right)^{nd_\mathsf{v}/d_\mathsf{c}}.$$

Note as well that $\widehat{\omega} = 1 - 1 = 0$ and that the orbits are given by $\mathcal{O}_0 = \{0\}$ and $\mathcal{O}_1 = \{1\}$. Hence, we observe

$$H(\theta) = H(\theta_\mathcal{O}) \quad \text{and} \quad \sum_{d\in\mathbb{D}_q} \theta(\mathcal{O}_d)\log(|\,\mathcal{O}_d\,|) = 0.$$

Thus, Expression (6.14) is identical to Expression (6.15).

6.3 Decoding Performance over Lee Channels

In this section, we analyze the error-correction performance of regular LDPC codes over the two channel models presented in Sections 5.1.1 and 5.1.2, respectively. First and foremost, we discuss an upper bound on the block error probability under maximum likelihood decoding over the memoryless Lee channel using a union bound argument. We then focus on the performance with respect to the belief propagation algorithm and the symbol message-passing decoder, respectively. For both decoders we start by adapting the decoders to the Lee metric over integer residue rings discussing the main changes and assumptions needed for providing a full density evolution analysis.

6.3.1 Bounds on the Block Error Probability Based on the Lee Weight Spectrum

We are interested in the average block error probability under maximum likelihood decoding of random regular LDPC code ensembles over $\mathbb{Z}/q\mathbb{Z}$ in the memoryless Lee channel. As the channel is symmetric, we can assume the transmission of the zero codeword. The maximum likelihood decoder fails if and only if there is a nonzero codeword $c \in \mathcal{C} \setminus \{0\}$ satisfying

$$P_{Y \mid x}(y \mid 0) \leq P_{Y \mid x}(y \mid c).$$

We refer to the probability of this event as the pairwise error probability and denote it by $\mathsf{PEP}(0 \to c)$. Note that, in the spirit of obtaining an upper bound on the block error probability, we break ties always in favor of the erroneous codeword. Using a union bound argument, we observe that the block error probability is upper bounded by the sum of all pairwise error probabilities, i.e.,

$$P_B(\mathcal{C}) \leq \sum_{c \in \mathcal{C} \setminus \{0\}} \mathsf{PEP}(0 \to c). \tag{6.16}$$

We can rewrite the pairwise error probability as

$$\mathsf{PEP}(0 \to c) = \mathbb{P}\left(\frac{P_{Y \mid x}(y \mid 0)}{P_{Y \mid x}(y \mid c)} \leq 1 \right). \tag{6.17}$$

Denoting the log-likelihood ratio as

$$\Lambda(y, c) := \log\left(\frac{P_{Y \mid X}(y \mid 0)}{P_{Y \mid X}(y \mid c)} \right)$$

we have $\mathsf{PEP}(0 \to c) = \mathbb{P}\left(\sum_{i=1}^{n} \Lambda(y_i, c_i) \leq 0\right)$. Hence, the analysis reduces to the analysis of the distribution of the random variables $\Lambda_\ell := \Lambda(Y, c = \ell)$, where Y is a random variable distributed following the Boltzmann-like distribution B_δ as defined in (5.5). Owing to the symmetry of the Boltzmann distribution, we have that

$$P_{Y \mid X}(y \mid c) = P_{Y \mid X}(-y \mid -c),$$

and therefore also

$$\Lambda(y, c = \ell) = \Lambda(-y, c = -\ell).$$

It follows that the distribution of Λ_ℓ equals the distribution of $\Lambda_{-\ell}$. Hence, the evaluation of (6.17) can be carried out by counting the number of elements in c possessing Lee weight ℓ with $\ell \in \{0, \dots \lfloor q/2 \rfloor\}$. We will therefore again make use of the Lee type of a codeword (see Definition 6.2.1). Thus, we can rewrite the pairwise error probability for any nonzero codeword $c \in \mathcal{C}$ as follows

$$\mathsf{PEP}(0 \to c) = \mathbb{P}\left(\sum_{\ell=1}^{\lfloor q/2 \rfloor} \sum_{j=1}^{n\theta_c(\ell)} \Lambda_j \leq 0 \right).$$

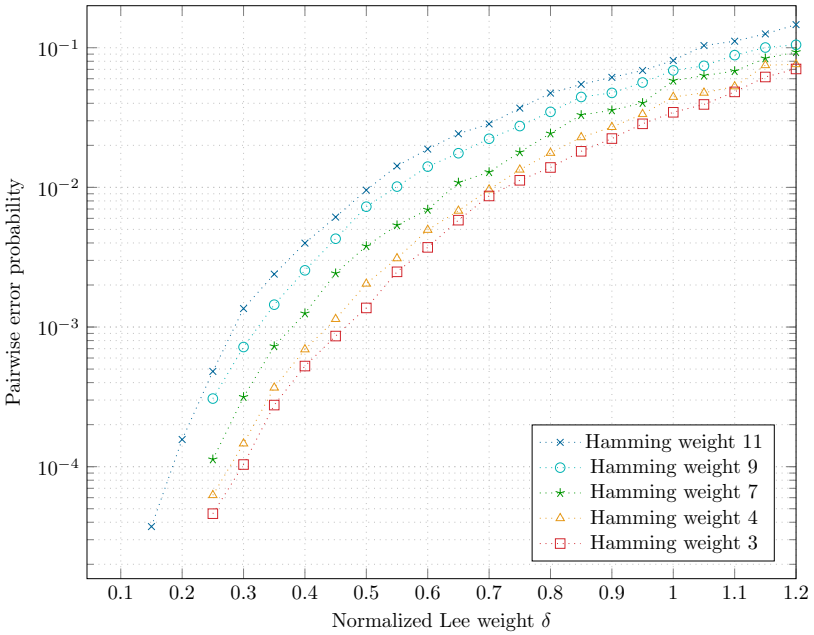

FIGURE 6.6: Comparison of the pairwise error probabilities over $\mathbb{Z}/9\mathbb{Z}$ of vectors of Lee weight 11 and varying Hamming weight.

This gives us an exact value of the pairwise error probability under maximum likelihood decoding. However, this expression requires eventually to iterate over every Lee type in the code $\mathcal{C}$ and is therefore inefficient for codes with large parameters. In the following we present a "worst case" candidate for the pairwise error probability, as long as $\mathrm{wt}_{\mathsf{L}}(c) = t \leq n$, which ultimately serves to upper bound the block error probability.

Lemma 6.3.1. *Consider a nonzero codeword $c \in \mathcal{C}$ such that $\mathrm{wt}_{\mathsf{L}}(c) = t \leq n$. Let $x^{(t)} \in (\mathbb{Z}/q\mathbb{Z})^n$ be of Lee type $\theta_{x^{(t)}} = (1 - t/n, t/n, 0, \ldots, 0)$. Over a memoryless Lee channel with $\delta \leq \delta_q$ we have*

$$\mathsf{PEP}(0 \to c) \leq \mathsf{PEP}(0 \to x^{(t)}),$$

where equality holds if and only if c is of the same Lee type $\theta_c = \theta_{x^{(t)}}$.

Observe that the nonzero positions of $x^{(t)}$ consist only of elements of Lee weight 1. Therefore, it holds that $\mathrm{wt}_{\mathsf{L}}(x^{(t)}) = \mathrm{wt}_{\mathsf{H}}(x^{(t)}) = \mathrm{wt}_{\mathsf{L}}(c)$. Figure 6.6 gives empirical evidence supporting the result of Lemma 6.3.1.

For the case $t > n$ there is a similar scenario stated in Lemma 6.3.2.

Lemma 6.3.2. *Consider a nonzero codeword $c \in \mathcal{C}$ such that $\mathrm{wt}_{\mathsf{L}}(c) = t > n$. Let $x^{(n)} \in (\mathbb{Z}/q\mathbb{Z})^n$ be of Lee type $\theta_{x^{(n)}} = (0, 1, 0, \ldots, 0)$. Over a memoryless Lee channel with $\delta \leq \delta_q$ we have*

$$\mathsf{PEP}(0 \to c) \leq \mathsf{PEP}\left(0 \to x^{(n)}\right).$$

We can use these results to upper bound on the block error probability of a linear code over the memoryless Lee channel as a function of the codes Lee distance spectrum, for $\delta \leq \delta_q$.

Corollary 6.3.3. *Consider an $[n, k]$ linear code $\mathcal{C} \subset (\mathbb{Z}/q\mathbb{Z})^n$. For all $\ell \in \{0, \ldots, n\lfloor q/2 \rfloor\}$ let $W_\ell^{(n)}(\mathcal{C})$ denote the Lee weight enumerator of $\mathcal{C}$. The block error probability of $\mathcal{C}$ under maximum likelihood decoding over the memoryless Lee channel $\delta \leq \delta_q$ is upper bounded as*

$$P_B(\mathcal{C}) \leq \sum_{\ell=1}^{n\lfloor q/2 \rfloor} W_\ell^{(n)}(\mathcal{C}) \mathbb{P}\left(\sum_{i=1}^{\min(\ell,n)} \Lambda_1 < 0 \right). \tag{6.18}$$

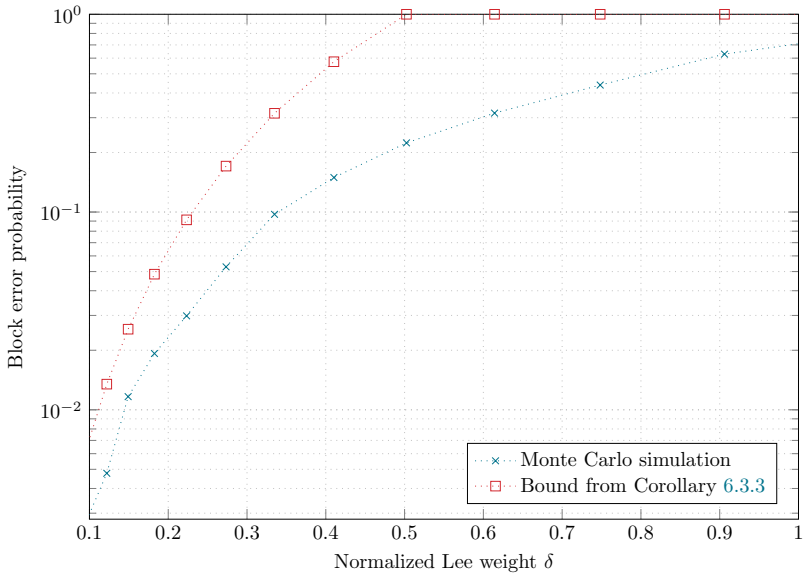

FIGURE 6.7: Comparison of the union bound from Corollary 6.3.3 with respect to the performance measured via Monte Carlo simulation for the linear code over $\mathbb{Z}/7\mathbb{Z}$ of length $n = 6$ and dimension $k = 2$ from Example 6.3.4.

Proof. The proof follows by applying Lemma 6.3.1 and Lemma 6.3.2 to the PEP-terms in the classical union bound. In fact, we can upper bound the block error probability (6.16) by

$$\sum_{\substack{c\in\mathcal{C}\setminus\{0\}\\ \mathrm{wt_L}(c)<n}} \mathsf{PEP}\left(0\to x^{(\mathrm{wt_L}(c))}\right) + \sum_{\substack{c\in\mathcal{C}\setminus\{0\}\\ \mathrm{wt_L}(c)\geq n}} \mathsf{PEP}\left(0\to x^{(n)}\right).$$

Now for both $x^{(\mathrm{wt_L}(c))}$ and $x^{(n)}$ the nonzero elements have Lee weight 1. Hence, we obtain

$$\begin{aligned}
&\sum_{\substack{c\in\mathcal{C}\setminus\{0\}\\ \mathrm{wt_L}(c)<n}} \mathsf{PEP}\left(0\to x^{(\mathrm{wt_L}(c))}\right) + \sum_{\substack{c\in\mathcal{C}\setminus\{0\}\\ \mathrm{wt_L}(c)\geq n}} \mathsf{PEP}\left(0\to x^{(n)}\right)\\
&= \sum_{\substack{c\in\mathcal{C}\setminus\{0\}\\ \mathrm{wt_L}(c)<n}} \mathbb{P}\left(\sum_{i=1}^{\mathrm{wt_L}(c)}\Lambda_1<0\right) + \sum_{\substack{c\in\mathcal{C}\setminus\{0\}\\ \mathrm{wt_L}(c)\geq n}} \mathbb{P}\left(\sum_{i=1}^{n}\Lambda_1<0\right)\\
&= \sum_{c\in\mathcal{C}\setminus\{0\}} \mathbb{P}\left(\sum_{i=1}^{\min(\mathrm{wt_L}(c),n)}\Lambda_1<0\right). \qquad (6.19)
\end{aligned}$$

Since $\mathbb{P}\left(\sum_{i=1}^{\min(\mathrm{wt_L}(c),n)}\Lambda_1<0\right)$ is the same for all codewords sharing the same Lee weight, we can take the sum over all possible Lee weights and multiply this probability by the number of codewords of that Lee weight. This yields the result. □

Example 6.3.4. Figure 6.7 depicts the union bound provided in Corollary 6.3.3, together with the block error probability estimated via Monte Carlo simulation. For the comparison we used a linear code over $\mathbb{Z}/7\mathbb{Z}$ of length $n = 6$ and dimension $k = 2$ with generator matrix

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 3 & 3 & 3 & 0\\ 0 & 1 & 0 & 4 & 3 & 3\end{pmatrix}.$$

As usually observed, the union bound provide accurate estimates at sufficiently low error probability.

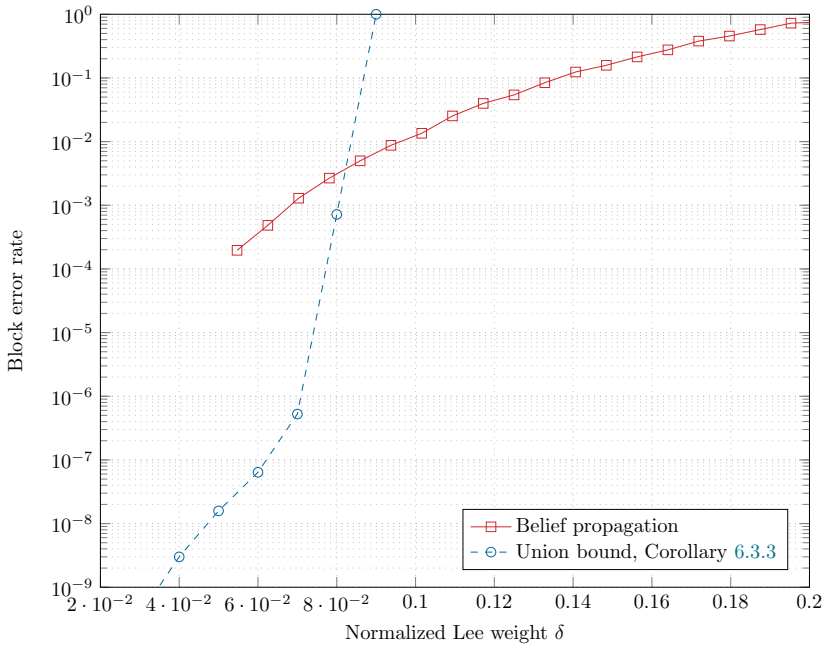

FIGURE 6.8: Union bound versus belief propagation over $\mathbb{Z}/4\mathbb{Z}$ for a random $(3, 6)$ LDPC code of length $n = 256$.

The union bound of Corollary 6.3.3 can be readily used to study the error floor performance of regular LDPC code ensembles. To do so, it is sufficient to replace the weight enumerator $W_\ell^{(n)}$ in (6.18) with the ensemble average enumerator $\overline{W}_\ell^{(n)}$. An example is provided in Figure 6.8, where the union bound on the maximum likelihood decoding average block error probability for the $(3, 6)$-LDPC code ensemble of length $n = 256$ over $\mathbb{Z}/4\mathbb{Z}$ is depicted. The result is compared with the numerical simulation for a code from the ensemble, under belief propagation decoding. As typical of union bounds on the block error probability, the bound is not informative above the cut-off rate of the channel. However, it provides an indication of the error probability regime at which an error floor may be observed, allowing for a quick estimation of the capability of certain code ensembles to attain given target error probabilities.

To get a better impression of this bound, let us focus on Λ_1 a bit more. The following result states mean and variance of the random variable Λ_1.

Lemma 6.3.5. *Let Λ_1 be defined as above for every $i \in \mathbb{Z}/q\mathbb{Z}$. Then Λ_1 has the following mean and variance.*

$$\mathbb{E}(\Lambda_1) = \mathbb{P}(Y = 1) - \mathbb{P}(Y = \lfloor q/2 \rfloor),$$
$$\mathrm{Var}(\Lambda_1) = \begin{cases} \big(1 - \mathbb{P}(Y = \lfloor q/2 \rfloor)\big) - \mathbb{E}(\Lambda_1)^2 & q \text{ odd} \\ 1 - \mathbb{E}(\Lambda_1)^2 & q \text{ even} \end{cases}.$$

Proof. Recall that $\Lambda_1 := \mathrm{wt_L}(Y - 1) - \mathrm{wt_L}(Y)$, where Y is a random variable following the distribution in (5.5). Applying the definition of the Lee weight, we easily observe that $\Lambda_1 \in \{-1, 0, 1\}$ if q is odd and $\Lambda_1 \in \{-1, 1\}$ if q is even. Moreover, for $k \in \mathbb{Z}/q\mathbb{Z}$ we obtain

$$\mathbb{P}(\Lambda_1 = k) = \begin{cases} \mathbb{P}(Y \in \{1, \ldots, \lfloor q/2 \rfloor\}) & \text{if } k = -1 \\ \mathbb{P}(Y = \lfloor q/2 \rfloor + 1) & \text{if } k = 0 \\ \mathbb{P}(Y \in \{\lfloor q/2 \rfloor + 2, \ldots, q\}) & \text{if } k = 1 \end{cases}, \quad \text{for } q \text{ odd, and}$$
$$\mathbb{P}(\Lambda_1 = k) = \begin{cases} \mathbb{P}(Y \in \{1, \ldots, \lfloor q/2 \rfloor\}) & \text{if } k = -1 \\ \mathbb{P}(Y \in \{\lfloor q/2 \rfloor + 1, \ldots, q\}) & \text{if } k = 1 \end{cases}, \quad \text{for } q \text{ even.}$$

The proof of the expected value then follows easily by applying its definition. For the variance, we use $\mathrm{Var}(\Lambda_1) = \mathbb{E}(\Lambda_1^2) - \mathbb{E}(\Lambda_1)^2$. □

Let $I \subset \{1, \ldots, n\}$ be an index set. With the help of the law of large numbers and the central limit theorem, we deduce that the sum $\sum_{i \in I} \Lambda_1$, as $|\, I \,|$ grows, follows a Gaussian distribution with mean and variance equal to the $|\, I \,|$-th multiple of the values in Lemma 6.3.5, respectively. Hence, the probabilities in (6.19) can be approximated by the tail of a Gaussian distribution approximating the distribution of Λ_1 with corresponding $|\, I \,|$. However, if I is relatively small it is convenient to work with the empirical distribution, given by the $|\, I \,|$-fold convolution of the distribution of Λ_1. Furthermore, for a fixed I we notice that the distribution of $\sum_{i \in I} \Lambda_1$ varies depending on whether I is even or odd. In fact, if I is even, it is more likely that the sum $\sum_{i \in I} \Lambda_1$ is even too and vice versa. Both cases can mutually be approximated by a Gaussian distribution and both cases will coincide as I tends to infinity.

6.3.2 Density Evolution Analysis

The analysis of the Lee spectrum of LDPC code ensembles can be used, in conjunction with the union bound, to analyze the ensembles behaviour under maximum likelihood decoding at low error rates. Nevertheless, it fails to capture the block error probability behaviour in the waterfall region, under iterative decoding. We hence complement the distance spectrum analysis with a density evolution characterization of the ensemble in the limit of large block lengths. In particular, we estimate the asymptotic iterative decoding threshold over the memoryless Lee channel under belief propagation and symbol message-passing decoding. The iterative decoding threshold $\delta^\star$ is defined as the largest value of the channel parameter δ for which, in the limit of large n and large maximal number of iterations $\ell_{\max}$, the symbol error probability of an LDPC code picked randomly from a (d_v, d_c) code ensemble becomes vanishing small [107]. Owing to the complexity of tracking the evolution of the distribution of multidimensional messages, under belief propagation decoding we resort to the Monte Carlo method [47]. We denote by $\delta^\star_{\mathsf{BP}}$ the decoding threshold under belief propagation decoding.

For the sake of completeness let us quickly recall the Monte Carlo method used in each iteration of the decoders to generate as much randomness as possible. Given a sparse parity-check matrix H of a regular LDPC code, the overall idea for each iteration i is the following:

1. Each variable node receives a vector $\mathbf{L}^{(i)} = (\log(y_1), \ldots, \log(y_n))$, where each $y_i \in \mathbb{Z}/q\mathbb{Z}$ is independently and uniformly generated following the distribution (5.5).

2. Scramble the entries of the parity-check matrix H, i.e., permute the edges between the variable nodes and the check nodes with some permutation $\Pi_1^{(i)}$.

3. Label each edge in the graph by a randomly chosen unit $u \in (\mathbb{Z}/q\mathbb{Z})^\times$ and perform the variable node-to-check node messages.

4. Randomly permute the edges between the check nodes and variable nodes again using some permutation $\Pi_2^{(i)}$. Transmit the check node-to-variable messages.

Figure 6.9 illustrates this process in a message-passing algorithm between check nodes and variable nodes.
The Monte Carlo simulation allows to model the labels of the edges as random variables over $(\mathbb{Z}/q\mathbb{Z})^\times$ which yields a distribution for the check node-to-variable node messages. This means, we are able to model the extrinsic channel according to some probability distribution. In fact, the extrinsic channel can then be viewed as a q-SC which we discuss below. Knowing the extrinsic channel model is then used in the density evolution analysis (see Section 6.3.2) to determine the iterative decoding thresholds.

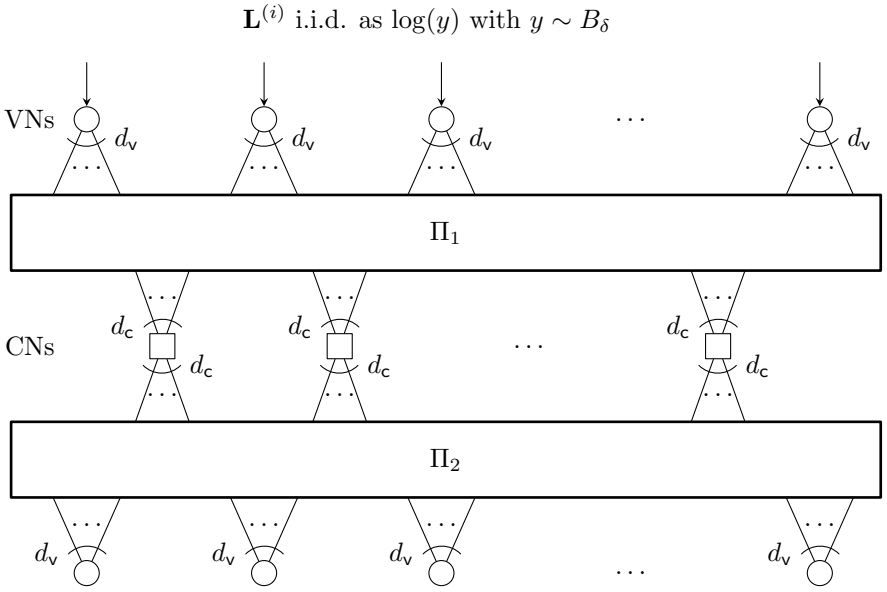

FIGURE 6.9: Illustration of one iteration using a Monte Carlo simulation.

The density evolution analysis for the symbol message-passing decoder has been introduced in [81, Sec. IV]. We briefly sketch the idea and emphasize the respective modifications according to the new memoryless Lee channel. For the symbol message-passing decoder the density evolution analysis not only aims at estimating the decoding threshold $\delta^\star_{\mathsf{SMP}}$, but it also provides bounds on the error probabilities ξ of the extrinsic channel modelled as q-SC which are needed in the computation of the aggregated extrinsic log-likelihood vector (6.3). Since the memoryless Lee channel is symmetric and the code is linear, we can assume that the all-zero codeword has been transmitted. Similar to the notation used in the description of the symbol message-passing decoder, we let $m^{(\ell)}_{\mathsf{v}\to\mathsf{c}}$ denote the message sent from variable node v to check node c in the ℓ-th iteration. For every $a \in \mathbb{Z}/q\mathbb{Z}$, let us define the probability of sending a message $m^{(\ell)}_{\mathsf{v}\to\mathsf{c}} = a$, knowing that originally zero has been transmitted as

$$p^{(\ell)}_a := \mathbb{P}\left(m^{(\ell)}_{\mathsf{v}\to\mathsf{c}} = a \mid X = 0\right).$$

Hence, recalling the memoryless Lee channel transition probability $P_{Y \mid X}(y \mid x)$ from (5.5), we initialize the density evolution analysis routine by computing for each $a \in \mathbb{Z}/q\mathbb{Z}$ the probabilities

$$p^{(0)}_a = P_{Y \mid X}(a \mid 0).$$

As indicated above, except from the computation of the aggregated extrinsic likelihood vector, the remaining steps of the density evolution analysis are identical to [81, Sec. IV]. In particular, we employ the q-SC approximation for the extrinsic channel.

Table 6.1 records the decoding thresholds $\delta^\star_{\mathsf{SMP}}$ and $\delta^\star_{\mathsf{BP}}$ for the symbol message-passing and belief propagation decoder, respectively, for both $(3, 6)$ and $(4, 8)$ regular LDPC code ensembles with q ranging from 5 to 8, as well as the Shannon limit $\delta^\star_{\mathsf{SH}}$ for the rate $R = 1/2$.

Remark 6.3.6. The choice of the discrete memoryless channel used to model the extrinsic channel plays a crucial role for the symbol message-passing algorithm, especially concerning the decoding performance. In [82], for the case of binary message-passing decoding, it was suggested to model the variable node inbound messages as observations of a binary symmetric channel, whose transition probability was estimated by means of density evolution analysis. The approach was generalized in [81] for symbol message-passing, where the variable node inbound messages are modelled as observations of a q-SC. In our setting we also model the extrinsic channel as a q-SC defined in (6.2). However, the q-SC model holds only in an approximate sense.

TABLE 6.1: Decoding thresholds for regular LDPC code ensembles under belief propagation and symbol message-passing decoding.

q	(v, c)	$\delta^\star_{\mathsf{BP}}$	$\delta^\star_{\mathsf{SMP}}$	$\delta^\star_{\mathsf{SH}}$
5	$(3, 6)$	0.2148	0.1039	0.2684
	$(4, 8)$	0.1802	0.1200	
6	$(3, 6)$	0.2485	0.1151	0.3147
	$(4, 8)$	0.2217	0.1405	
7	$(3, 6)$	0.3086	0.1261	0.3560
	$(4, 8)$	0.2686	0.1539	
8	$(3, 6)$	0.3135	0.1374	0.3950
	$(4, 8)$	0.2690	0.1623	

The adoption of the q-SC approximation is particularly useful from a practical viewpoint since the variable node processing in symbol message-passing decoding becomes particularly simple if the variable node-to-check node messages are assumed to be observations of an extrinsic q-SC. Moreover, this specific choice is motivated by the fact that, for LDPC codes over finite fields, the extrinsic channel transition probabilities, averaged over a uniform distribution of nonzero elements in the parity-check matrix, yield (in the limit of a large block length) a q-SC [81]. The Lemma 6.3.7 for q prime, supports this statement.

Lemma 6.3.7. *Consider a prime number q. Let H be a random variable drawn uniformly at random form the multiplicative group $(\mathbb{Z}/q\mathbb{Z})^\times$ and let X be any random variable over $\mathbb{Z}/q\mathbb{Z}$. Define the random variable $V = X \cdot H$. Then V follows a q-SC-like distribution given as*

$$\mathbb{P}(V = v) = \begin{cases} \mathbb{P}(X = 0) & \text{if } v = 0 \\ \frac{1}{q-1}(1 - \mathbb{P}(X = 0)) & \text{else.} \end{cases}$$

Proof. Since H is drawn uniformly at random from $(\mathbb{Z}/q\mathbb{Z})^\times$ and q is a prime, for every $h \in (\mathbb{Z}/q\mathbb{Z})^\times$ it holds that

$$\mathbb{P}(H = h) = \frac{1}{q-1}. \tag{6.20}$$

Firstly, let us focus on $\mathbb{P}(V = 0)$. Since over a finite field zero is the only zero-divisor and since H is defined only over the multiplicative group $(\mathbb{Z}/q\mathbb{Z})^\times$, the first case follows, i.e.,

$$\mathbb{P}(V = 0) = \mathbb{P}(X = 0).$$

Consider than $\mathbb{P}(V = v)$, where $v \in (\mathbb{Z}/q\mathbb{Z})^\times$. Recall that over a finite field for every unit $x \in (\mathbb{Z}/q\mathbb{Z})^\times$ there exists a unique $h \in (\mathbb{Z}/q\mathbb{Z})^\times$ such that $x \cdot h = v$. We denote this by $(\star)$. Furthermore, note that X and H are independent, denoted by $X \perp\!\!\!\perp H$.

$$\begin{aligned}
\mathbb{P}(V = v) = \mathbb{P}(X \cdot H = v) &= \sum_{x \in (\mathbb{Z}/q\mathbb{Z})^\times} \sum_{\substack{h \in (\mathbb{Z}/q\mathbb{Z})^\times \\ x \cdot h = v}} \mathbb{P}(X = x,\, H = h) \\
&\overset{\perp\!\!\!\perp}{=} \sum_{x \in (\mathbb{Z}/q\mathbb{Z})^\times} \sum_{\substack{h \in (\mathbb{Z}/q\mathbb{Z})^\times \\ x \cdot h = v}} \mathbb{P}(X = x)\mathbb{P}(H = h) \\
&\overset{(\star),\,(6.20)}{=} \sum_{x \in (\mathbb{Z}/q\mathbb{Z})^\times} \frac{1}{q-1}\mathbb{P}(X = x) \\
&= \frac{1}{q-1}(1 - \mathbb{P}(X = 0)).
\end{aligned}$$

□

Even though for q is non-prime the average extrinsic channel transition probabilities can not be represented by a q-SC, we still make this assumption. This might result in a suboptimal decoding performance but yields a good estimate on the asymptotic density evolution. Recall, that the nonzero entries of a parity-check matrix of an LDPC code over $\mathbb{Z}/q\mathbb{Z}$ lie in the set of units $(\mathbb{Z}/q\mathbb{Z})^\times$ and label the edges of the corresponding bipartite graph. Assuming that we consider an integer ring $\mathbb{Z}/q\mathbb{Z}$ consisting of relatively many units and that these nonzero entries are chosen uniformly at random from $(\mathbb{Z}/q\mathbb{Z})^\times$, the q-SC assumption is fair. In fact similar assumptions for other channels have been made in the past (see for instance [138]) and helps to provide a density evolution analysis for the considered message-passing algorithms. Empirical evidence obtained by measuring the total variation distance between the true extrinsic channel and the q-SC shows that the q-SC can still be used to accurately model the actual extrinsic channel, especially if the ring possesses relatively many unit elements. More precisely, we show numerically that the total variation distance between the two message distributions tends to zero as the number of iteration grows. We denote by $\mathcal{U}_q$ the fraction of units in $\mathbb{Z}/q\mathbb{Z}$, i.e.,

$$\mathcal{U}_q := \frac{|\,(\mathbb{Z}/q\mathbb{Z})^\times\,|}{|\,\mathbb{Z}/q\mathbb{Z}\,|}.$$

In order to cover different cases and support the conjecture that the q-SC assumption is especially accurate for integer rings with relatively many units, we chose three integer rings having different fractions of units. Namely, we chose $\mathbb{Z}/8\mathbb{Z}$, $\mathbb{Z}/9\mathbb{Z}$ and $\mathbb{Z}/12\mathbb{Z}$ with corresponding fraction of units

$$\mathcal{U}_8 = 1/2,\ \mathcal{U}_9 = 2/3,\ \mathcal{U}_{12} = 1/3.$$

Figures 6.10 and 6.11 show the evolution of the total variation distance with the number of iterations for $(3, 6)$-regular and $(4, 8)$-regular LDPC code ensembles, respectively. In each figure and for each integer ring, we consider three different situations:

1. the relative Lee weight δ is below $\delta^\star_{\mathsf{SMP}}$,
2. the relative Lee weight δ is close to $\delta^\star_{\mathsf{SMP}}$, and
3. the relative Lee weight δ exceeds $\delta^\star_{\mathsf{SMP}}$.

As shown by the Figures 6.10 and 6.11 in all the three cases the total variation distance between the extrinsic channel distribution and the q-SC converges to zero as the number of iterations increase. Hence, the figures clearly support the conjecture on the fraction of units $\mathcal{U}_q$ as well as the choice to model the average extrinsic channel transition probabilities by a q-SC.

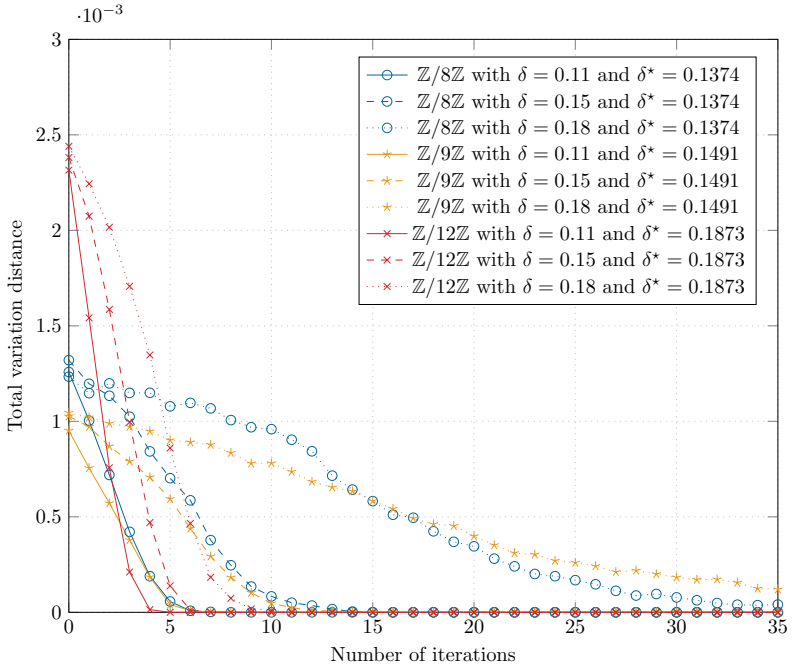

FIGURE 6.10: Evolution of the total variation distance between the extrinsic channel distribution and the q-SC for regular $(3, 6)$ LDPC code ensembles in the symbol message-passing decoder.

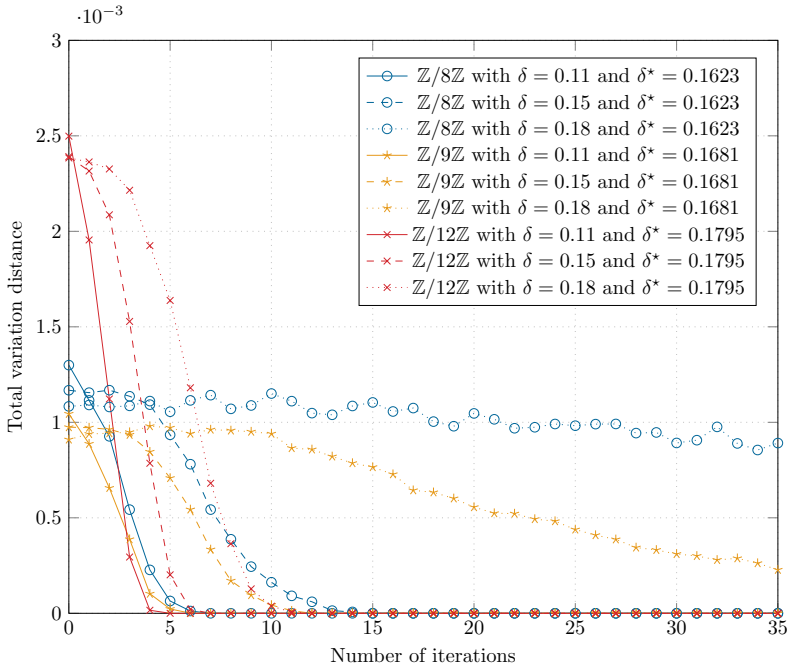

FIGURE 6.11: Evolution of the total variation distance between the extrinsic channel distribution and the q-SC for regular $(4, 8)$ LDPC code ensembles in the symbol message-passing decoder.

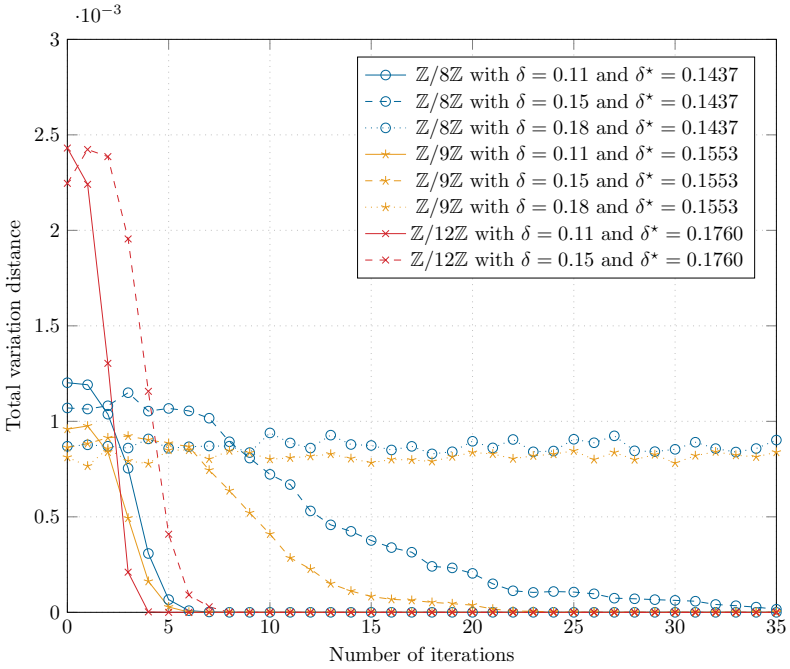

FIGURE 6.12: Evolution of the total variation distance between the extrinsic channel distribution and the q-SC for regular $(5, 10)$ LDPC code ensembles in the symbol message-passing decoder.

6.3.3 Numerical Results

We finally present numerical results showing the decoding performance (in terms of block error rates) of $(3, 6)$ regular LDPC codes of length $n = 256$ under both belief propagation and symbol message-passing decoding. We chose to analyze the performances over three different integer rings, namely $\mathbb{Z}/5\mathbb{Z}$, $\mathbb{Z}/7\mathbb{Z}$ and $\mathbb{Z}/8\mathbb{Z}$. The performances will additionally be compared to the Lee-symbol flipping decoder presented in [114, Algorithm 2]. Following the suggestions of [114], we assumed a decoding threshold $\tau = \frac{d_v}{2}$ for the Lee-symbol flipping decoder. All the results were obtained using Monte Carlo simulations where we generated the parity-check matrices via the progressive edge growth (PEG) algorithm [77] assuming that the nonzero entries are chosen uniformly at random from $(\mathbb{Z}/q\mathbb{Z})^\times$. The error vectors in the constant Lee-weight channel are drawn uniformly at random from the Lee sphere of a given radius representing the desired weight according to [16, Algorithm 1 and 2], whereas in the memoryless Lee channel the entries of the error vector are drawn according to the distribution defined in (5.5). In both cases, the performance is compared to the random coding union bounds established in Corollary 5.2.4 and Theorem 5.2.7, respectively.

The block error probability evaluated over the memoryless Lee channel is shown in Figure 6.13. The random coding union bounds (dotted in the graph) show clearly the impact of the size q of the finite integer ring, i.e., larger q admit a larger relative Lee weight δ. This is also observed in the performance under both belief propagation and symbol message-passing decoding as well as in the Lee-symbol flipping decoder. The impact of q in the symbol message-passing is not only important for the admissible choices of δ, it also shows clearly the difference between q being prime and composite. While a small gain is achieved when considering $\mathbb{Z}/8\mathbb{Z}$ instead of $\mathbb{Z}/7\mathbb{Z}$ under belief propagation decoding, the performance slightly suffers under symbol message-passing decoding meaning there is almost no gain. This might be due to the q-SC assumption which holds only in an asymptotic sense for the non-field case, as discussed in Section 6.3.2.

We observe the same effect in the performance over the constant Lee-weight channel in Figure 6.14, i.e., there is almost no gain visible when moving from $q = 7$ to $q = 8$ under the symbol message-passing decoder. Analogous to the memoryless case, we observe the same impact of the size of $\mathbb{Z}/q\mathbb{Z}$ on the possible choices of δ which is captured by the RCU bound for the constant Lee-weight channel. In both channel models we observe that

the symbol message-passing decoder outperforms the Lee-symbol flipping decoder despite the q-SC assumption in the extrinsic channel of the symbol message-passing. We want to emphasize and acknowledge here that the Lee-symbol flipping was originally designed for low-Lee-density parity-check codes which form a special class of LDPC codes. Hence, when comparing the performances over the two decoders the difference of the code classes might be taken in consideration. Nevertheless, we will not focus deeper on this argument and leave this subject to future investigations. However, we believe that the additional knowledge about the marginal distribution plays a crucial part in the performance gain under symbol message-passing decoding. Observe that the estimated threshold values obtained via density evolution analysis and stored in Table 6.1 match well to the actual block error rates achieved by both belief propagation and symbol message-passing decoding. As expected from the predictions in Table 6.1, belief propagation clearly outperforms symbol message-passing decoding. However, the symbol message-passing algorithm shows a performance that is appealing for applications demanding low-complexity decoding [114].

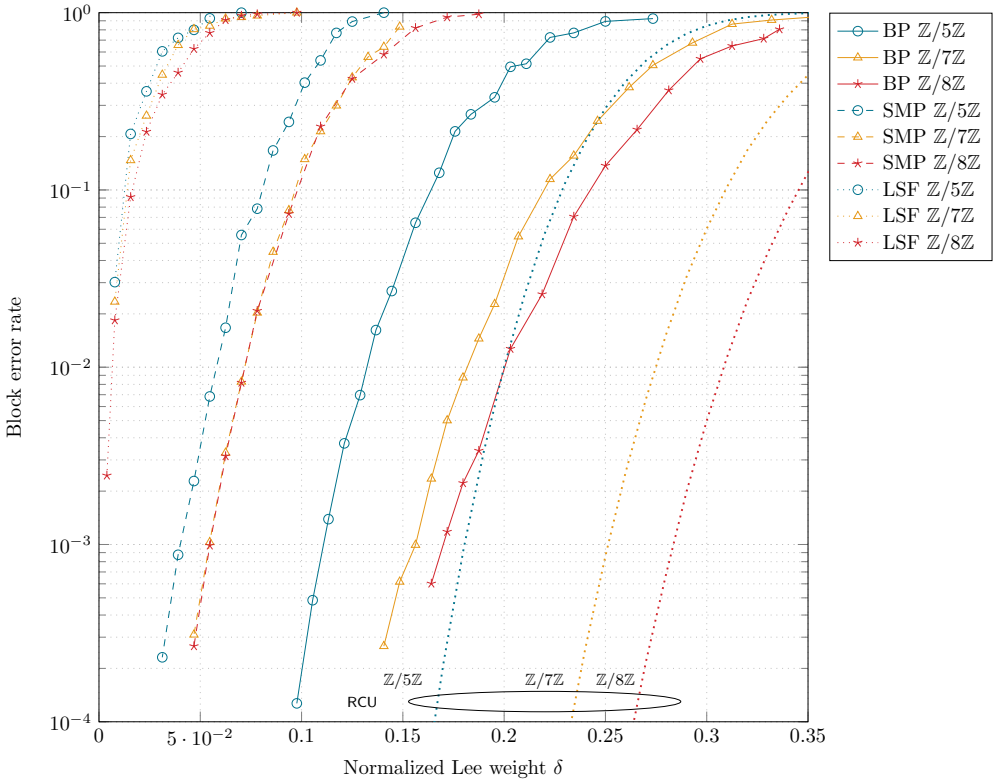

FIGURE 6.13: Block error rate vs. δ for regular $(3, 6)$ nonbinary LDPC codes of length $n = 256$ over the memoryless Lee channel. Lee-symbol flipping compared to the random coding union bound from Theorem 5.2.7, symbol message-passing and belief propagation decoding.

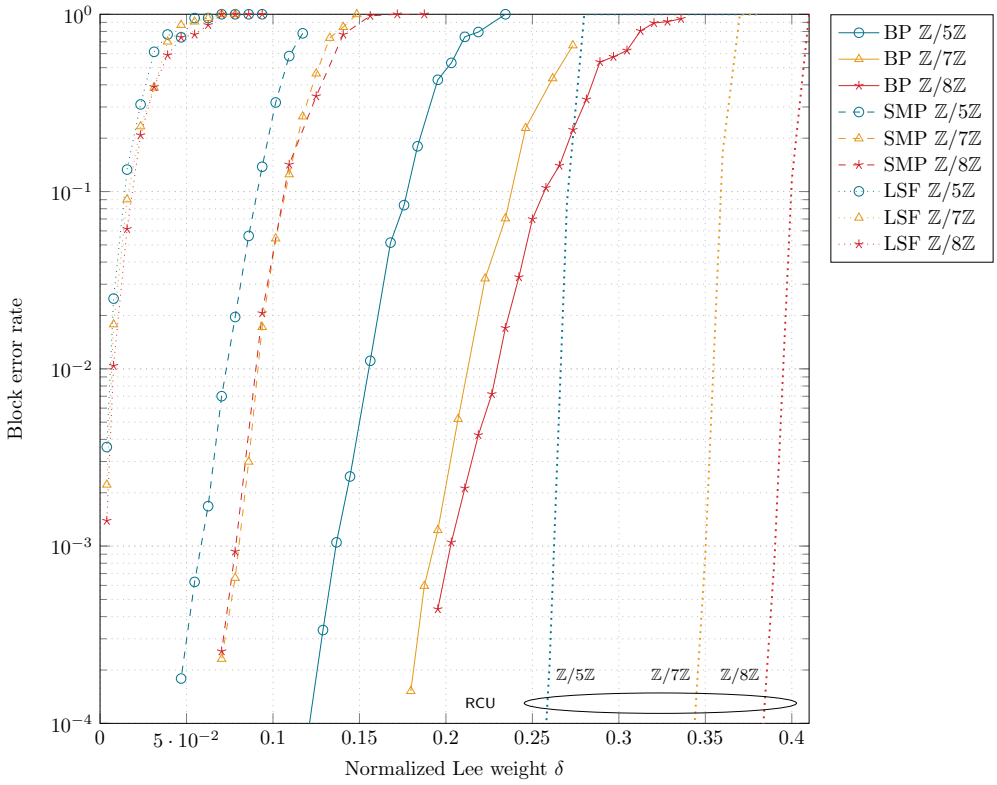

FIGURE 6.14: Block error rate vs. δ for regular $(3, 6)$ nonbinary LDPC code ensembles of length $n = 256$ and rate $R = 1/2$ on the constant Lee-weight channel under Lee-symbol flipping, symbol message-passing and belief propagation decoding compared to the random coding union bound from Theorem 5.2.3.

6.4 Summary and Outlook

In this chapter we introduced a code family in the Lee metric, namely the family of regular Lee-LDPC codes over the integer residue ring $\mathbb{Z}/q\mathbb{Z}$, for any positive integer q. We analyzed the ensemble in its algebraic structure and in terms of its error-correction performance over the Lee channels. More specifically, the group action of the set of units $(\mathbb{Z}/q\mathbb{Z})^\times$ on $\mathbb{Z}/q\mathbb{Z}$ was used to understand how the Lee type of a vector changes when passing it along the edges of a parity-check matrix. Together with generating functions to enumerate the number of valid check node assignments an expression for the expected weight enumerator of a regular LDPC ensemble in the Lee metric with its asymptotic growth rate has been discussed (see Section 6.2 and especially Corollary 6.2.14). The average weight enumerator of the regular LDPC ensemble was then used to derive an upper bound on the maximum likelihood block error probability. Especially in the regime of low error probability, generally known as the error floor region, the bound provides relevant information about the code's performance. For a full analysis we simulated the decoding performance of $(3, 6)$ regular LDPC codes of length $n = 256$ over the Lee channels. Here, two decoding algorithms were considered: a nonbinary belief propagation algorithm, and a low-complexity message-passing algorithm where the exchanged messages are elements of the residue ring $\mathbb{Z}/q\mathbb{Z}$. The simulation results confirmed the outcomes of the density evolution analysis, that is belief propagation decoding outperforms symbol message-passing decoding. Nevertheless, the performance under symbol message-passing decoding seems a promising option for applications asking for low complexity (such as code-based cryptosystems involving the Lee metric).

For this thesis only regular LDPC code ensembles have been considered. In the future, this work could be extended to other families of LDPC codes, such as irregular or protograph-based ensembles, or to MDPC code families. On a more algebraic side, the expected weight

enumerator of the LDPC codes have been derived applying the method of types to the Lee-weight decomposition of a vector. Since the MacWilliams identities do not hold for the weight enumerator of Lee-metric codes, we might consider the Lee-type spectrum instead which would subdivide the class of codewords of given Lee weight into even smaller groups depending on their underlying weight decomposition.

Chapter 7

Restricted Information Set Decoding

In this final chapter we present an application of the marginal distribution of a vector of given Lee weight (Lemma 5.1.4) to code-based cryptography. More explicitly, we discuss its impact on information set decoding in the Lee metric. To start this chapter, in Section 7.1 we will give an overview on Information Set Decoding (ISD) as originally introduced over finite fields endowed with the Hamming metric. Since many improvements of Prange's plain ISD algorithm are based on solving sub-problems of the syndrome decoding problem using a generalized birthday algorithm approach, we recap this technique too. Information set decoding is the yet best known method to attack the syndrome decoding problem which is the underlying NP-hard problem to most code-based cryptosystems (like the McElice cryptosystem [93] or the Niederreiter scheme [97]). Even though McEliece's scheme is as old as RSA and yet unbroken, modern code-based cryptography is moving away from the classical idea of McEliece, where the distinguishability of the secret code obstructs a security reduction to the syndrome decoding problem, and moving towards ideas from lattice-based cryptography such as the ring learning with error problem. Note, that the Lee metric is the closest metric in coding theory to the Euclidean metric used in lattice-based cryptography, in the sense that both metrics take into consideration the magnitude of the entries.

Due to new challenges in code-based cryptography, such as the search for efficient signature schemes, also other metrics are now investigated. For example the rank metric has gained a lot of attention due to the NIST submission ROLLO [1] and RQC [2]. While the understanding on the hardness of the rank-metric syndrome decoding problem is still rapidly developing (see the new benchmark achieved in [13]), it is still unknown whether the rank-metric syndrome decoding problem is an NP-hard problem.

The situation for the Lee metric is quite different. The Lee-metric syndrome decoding problem was first studied for codes over $\mathbb{Z}/4\mathbb{Z}$ in [75]. Later, in [130] the problem was shown to be NP-hard over any $\mathbb{Z}/p^s\mathbb{Z}$ and several generic decoding algorithms to solve the problem have been provided. Also, the paper [40] confirmed the cost regimes of [129] and more importantly the observation, that Lee-metric ISD algorithms cost more than their Hamming metric counterparts for fixed input parameters. Thus, the Lee metric has a great potential to reduce the key sizes or signature sizes in code-based cryptosystems. This could be of special interest, since NIST recently launched a second call for post-quantum signature schemes to be standardized.

For the syndrome decoding problem in any metric, we assume that the instance is given by a randomly chosen parity-check matrix and an error vector of fixed weight which was also chosen uniformly at random. In the Lee metric, for such a vector, in the limit of its length, we know almost surely of which Lee weights its positions are composed, namely they follow the Boltzmann-like distribution discussed in 5.1.4. In Section 7.3 we use this result on the marginal distribution of such a vector of given Lee weight to reduce the cost of the Lee-metric ISD algorithms further and thus contribute to the recent advances in understanding the hardness of this problem, with the final goal of deeming this setting secure for applications.

The results presented in Section 7.2, 7.3 and 7.4 were studied in collaboration with Karan Khathuria and Violetta Weger in [18].

7.1 Background to Information Set Decoding

Code-based cryptosystems are based on a mathematically hard problem, such as the *syndrome decoding problem*. This problem aims to decode a random linear code over a finite field endowed with some metric using a parity-check matrix of the code. Originally, this metric has been defined to be the Hamming metric. Note that there is a generator-matrix equivalent description of the problem which we refer to as *generic decoding problem*.

In the following let us consider an $[n, k]$-linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$, where $q = p^s$.

Problem 7.1.1 (Generic Decoding Problem)**.** Let $k \leq n$ be two positive integers and $G \in \mathbb{F}_q^{k\times n}$ a generator matrix of $\mathcal{C}$. Given t be a positive integer, $m \in \mathbb{F}_q^k$ and $y \in \mathbb{F}_q^n$, find $e \in \mathbb{F}_q^n$ such that

$$y = mG + e \quad \text{and} \quad \mathrm{wt}_{\mathsf{H}}(e) = t.$$

Problem 7.1.2 (Syndrome Decoding Problem)**.** Let $k < n$ be two positive integers and $H \in \mathbb{F}_q^{(n-k)\times n}$ a generator matrix of $\mathcal{C}$. Given t be a positive integer and $s \in \mathbb{F}_q^{n-k}$, find $e \in \mathbb{F}_q^n$ such that

$$s = eH^\top \quad \text{and} \quad \mathrm{wt}_{\mathsf{H}}(e) = t.$$

Fixing the dimension k and letting the block length n tend to infinity, then the Gilbert-Varshamov bound provides us a threshold value τ_{GV} for the weight t of the error vector by

$$\binom{n}{\tau_{\mathsf{GV}}} = q^{n-k}.$$

If $\mathrm{wt}(e) = t < \tau_{\mathsf{GV}}$ then we expect that there is a unique solution to the syndrome (or generic) decoding problem. If t exceeds the threshold τ_{GV}, we expect $q^{k-n}\binom{n}{t}$ solutions to the problem.

Both of the problems have been shown to be NP-complete [14, 22]. A brute-force algorithm would run through every vector $x \in \mathbb{F}_q^n$ and check whether the two conditions are satisfied. This would yield a cost of

$$\binom{n}{t}(q-1)^t t(n-k) \text{ bits.}$$

Recently, these decoding problems have also been considered in other metrics like, for instance, the Lee metric [130].

If the instances of the syndrome decoding problem are random, then the best known methods to tackle the problem are ISD and the generalized birthday algorithm. If only a few solutions are given, we prefer ISD over the generalized birthday decoding. The first ISD algorithm was proposed by Prange [104] in 1962. This was even before code-based cryptosystems have been proposed. All other ISD algorithms are based on Pranges version and all of them define improvements of Pranges original decoding algorithm. Although the literature on ISD algorithms in this classical case is vast (see [21, 26, 36, 37, 39, 54, 84, 85, 92, 125]), the cost of generic decoding has only decreased little and is considered stable. The fastest algorithm over the binary until this day is called BJMM algorithm [21] and uses the idea of representation technique from [76]. For an overview of the binary case see [94]. With new cryptographic schemes proposed over general finite fields, most of these algorithms have been generalized to $\mathbb{F}_q$ (see [69, 74, 78, 96, 100]).

7.1.1 General Framework and Prange's Algorithm

For a given code $\mathcal{C}$ of length n and dimension k over the finite field $\mathbb{F}_q$ let us introduce the notion of an information set.

Definition 7.1.3. Let $k \leq n$ be two positive integers and let $\mathcal{C}$ be an $[n, k]_q$-linear code. We call a set $\mathcal{I} \subset \{1, \ldots, n\}$ an *information set* of size k if it satisfies

$$|\,\mathcal{C}\,| = |\,\mathcal{C}_{\mathcal{I}}\,|\,.$$

An information set is hence a set of k positions that uniquely determines every codeword. As we know from an encoding map induced by a generator matrix G of $\mathcal{C}$, $\mathcal{C}$ is completely defined by k positions. Hence, the definition of an information set makes sense. Furthermore, there exist at most $\binom{n}{k}$ many information sets.

Assume that a word $y = c + e \in \mathbb{F}_q^n$ has been received, where c is a codeword and e an error term of fixed weight t. In principle, the main idea behind this algorithm is to guess a random information set $\mathcal{I}$ of size k and hope that it does not contain any error positions. That means we would like to have

$$y_{\mathcal{I}} = c_{\mathcal{I}}$$

or equivalently $e_{\mathcal{I}} = 0$. More explicitly, Pranges algorithm [104] works as follows:

1. Choose an information set $\mathcal{I} \subset \{1, \dots, n\}$ of size k for $\mathcal{C}$.
2. Bring the parity-check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$ into standard form corresponding to $\mathcal{I}$. That is, find an invertible matrix $U \in \mathbb{F}_q^{(n-k)\times(n-k)}$ satisfying
$$(UH)_{\mathcal{I}} \in \mathbb{F}_q^{(n-k)\times k} \quad \text{and} \quad (UH)_{\mathcal{I}^c} = \mathbb{I}_{n-k}.$$
3. Go through every $e \in \mathbb{F}_q^n$ with $\mathrm{wt}_{\mathsf{H}}(e) = t$ and given weight distribution. Check whether $e(UH)^\top = eH^\top U^\top = sU^\top$.
   (a) If satisfied, return the vector e.
   (b) Otherwise, restart with step 1 by choosing a new information set.

Complexity Anaylsis

All the variants of ISD repeat a (large) number of independent iterations $\mathcal{N}$ all consisting of $\mathcal{K}$, the (expected) *cost* per iteration, and a success probability $\mathcal{P}$ reciprocal to $\mathcal{N}$, i.e., $\mathcal{P} = \frac{1}{\mathcal{N}}$. The cost of an ISD is then given by:

$$C := \mathcal{O}\left(\frac{1}{\mathcal{P}} \cdot \mathcal{K}\right).$$

Considering Prange's ISD algorithm, an error is found if it has the form $e = (e_1, 0, \dots, 0)$ where $e_1 \in \mathbb{F}_q^{n-k}$ is of weight t. Assuming that there exists a unique solution to the syndrome decoding problem (i.e., t is below the threshold given by the Gilbert-Varshamov bound), the success probability is given by

$$\mathcal{P}_{\text{Prange}} = \frac{\binom{n-k}{t}}{\binom{n}{t}}.$$

For each iteration, Prange's algorithm requires $\mathcal{K} = n(n-k)$ column operations (mainly due to Gaussian elimination). Hence, we obtain a cost of

$$C_{\text{Prange}} = \mathcal{O}\left(\frac{n(n-k)\binom{n}{t}}{\binom{n-k}{t}}\right).$$

7.1.2 Improved ISD Variants

Prange's ISD Algorithm has been generalized by Stern [125] and Dumer [53], respectively. In contrast to Prange's algorithm, they did not assume that all the erroneous positions lie outside an information set. Instead, their idea was to decompose the problem into a smaller instance, i.e., another syndrome decoding problem with smaller parameters. To solve the smaller instance they represented the error term as a sum of two vectors of a specific form. Given an instance of the syndrome decoding problem in the Lee metric, the highlevel idea is the following:

1. Find an invertible matrix $U \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k)\times(n-k)}$ such that
$$UH^\top = \begin{pmatrix} \mathbb{I}_{n-k-\ell} & 0 \\ A^\top & B^\top \end{pmatrix}.$$
2. Transform the syndrome equation accordingly to
$$\begin{pmatrix} e_1 & e_2 \end{pmatrix} UH^\top = \begin{pmatrix} s_1 & s_2 \end{pmatrix} = sU.$$
3. Assume, $\mathrm{wt}_{\mathsf{L}}(e_1) = t - v$ and $\mathrm{wt}_{\mathsf{L}}(e_2) = v$. Hence, we need to solve
$$\begin{aligned} e_1 + e_2 A^\top &= s_1 \\ e_2 B^\top &= s_2 \end{aligned}$$
4. Solve the smaller instance of the Lee-syndrome decoding problem given by $e_2 B^\top = s_2$. Immediately check whether $e_1 = s_1 - e_2 A^\top$ has Lee weight $t - v$.

To find all the solutions of the smaller instance $e_2 B^\top = s_2$, Stern and Dumer applied an enumeration technique In a nutshell, we split the matrix B and the syndrome s into two parts, $B_1^\top$, $B_2^\top$ and s_{11}, s_{22}, respectively. Then we enumerate the following two sets
$$\begin{aligned} \mathcal{L}_1 &:= \left\{ x_1 \in \mathbb{F}_q^{n/2} \mid x_1 B_1^\top = s_{11} \text{ and } \mathrm{wt}(x_1) = v/2 \right\}, \quad \text{and} \\ \mathcal{L}_2 &:= \left\{ x_2 \in \mathbb{F}_q^{n/2} \mid x_2 B_2^\top = s_{22} \text{ and } \mathrm{wt}(x_2) = v/2 \right\}. \end{aligned}$$
If $\mathcal{L}_1 \cap \mathcal{L}_2 \neq \emptyset$ then solutions exist.

Further improvements have been proposed by May, Meurer and Thomae [92], and by Becker, Joux, May and Meurer in [21]. In the ISD variant proposed in [21] the authors gave an improved version of the representation technique by introducing an additional level of recursive call and an requiring that the weight of the elements in the lists $\mathcal{L}_1$ and $\mathcal{L}_2$ is increased ε, i.e., given by $v/2 + \varepsilon$. creating an overlap of (in average) ε nonzero positions in two words. The idea is, that two words of weight $t/2 + \varepsilon$ and length n are expected to have a sum of weight t. We refer to this algorithm as the BJMM algorithm.

In the Lee metric, the BJMM algorithm on two levels was shown to be the fastest algorithm among the Lee-variants of the ISD algorithms [130].

7.2 Restricted Lee-Spheres

For a prime number p and a positive integer s, we focus on the integer residue ring $\mathbb{Z}/p^s\mathbb{Z}$. Recall from (4.1) that the maximum Lee weight in $\mathbb{Z}/p^s\mathbb{Z}$ is given by $M = \lfloor p^s/2 \rfloor$. In our restricted version of the BJMM-algorithm we do not use the full n-dimensional Lee-sphere of a given radius t. In fact, we restrict the entries of the vectors in the sphere to a certain maximum or minimum Lee weight threshold that we denote by $r \in \{0, \ldots, M\}$. We are interested in the expected number of entries that have Lee weight smaller or larger than this threshold r. Let $\psi(r, t, n, p^s)$ denote the expected number of entries of e which have a larger Lee weight than r and let $\varphi(r, t, n, p^s)$ denote the expected Lee weight of e without the entries of larger Lee weight than r. In addition, for some randomly chosen subset $I \subset \{1, \ldots, n\}$ of size $0 \leq \ell \leq n$, let us denote by $\sigma(\ell, t, n, p^s)$ the expected support size of e_I.

Lemma 7.2.1. *Let e be chosen uniformly at random in $\mathcal{S}_{t,p^s}^{(n)}$, $r \in \{0, \ldots, M\}$ and $0 \leq \ell \leq n$. Then*

$$\psi(r,t,n,p^s) = n \sum_{i=r+1}^{M} \mathbb{P}(\mathrm{wt_L}\, E = i),$$
$$\varphi(r,t,n,p^s) = n \sum_{i=0}^{r} i \cdot \mathbb{P}(\mathrm{wt_L}\, E = i),$$
$$\sigma(\ell,t,n,p^s) = \ell \sum_{i=1}^{M} \mathbb{P}(\mathrm{wt_L}\, E = i).$$

Proof. The proof easily follows from (5.7) and using the assumption that each entry of e is independent. □ □

Thus, we let $\mathcal{S}_{v,p^s}^{(n)}(\underline{r})$, respectively $\mathcal{S}_{v,p^s}^{(n)}(\overline{r})$, denote the Lee-sphere of radius $v \in \mathbb{N}$ centered at the origin with entries restricted to $\{0, \pm 1, \ldots, \pm r\}$, respectively to $\{\pm r, \ldots, \pm M\}$. That is,

$$\mathcal{S}_{v,p^s}^{(n)}(\underline{r}) := \{x \in \{0, \pm 1, \ldots, \pm r\}^n | \, \mathrm{wt_L}\,(x) = v \,\},$$
$$\mathcal{S}_{v,p^s}^{(n)}(\overline{r}) := \{x \in \{\pm r, \ldots, \pm M\}^n | \, \mathrm{wt_L}\,(x) = v \,\}.$$

The size of the sphere is crucial to understand the number of representatives and to analyze the complexity of the proposed algorithm, as the list sizes depend on the sphere size. Let us describe the sphere sizes in terms of generating functions. Then, use the saddle point technique (see Section 3.3) to compute their limit in the dimension n.

Similarly to the generating function for the n-dimensional Lee-sphere of fixed radius t in (3.10), the generating function for $\mathcal{S}_{t,p^s}^{(n)}(\underline{r})$ and $\mathcal{S}_{t,p^s}^{(n)}(\overline{r})$ are given by $\Phi_{\underline{r}}(x) := f_{\underline{r}}(x)^n$ and $\Phi_{\overline{r}}(x) := f_{\overline{r}}(x)^n$, respectively, where

$$f_{\underline{r}}(x) := \begin{cases} 1 + 2\sum_{i=1}^{M-1} x^i + x^M & \text{if } p = 2 \text{ and } r = M, \\ 1 + 2\sum_{i=1}^{r} x^i & \text{otherwise.} \end{cases}$$

$$f_{\overline{r}}(x) := \begin{cases} f_M(x) & \text{if } r = 0, \\ 2\sum_{i=r}^{M-1} x^i + x^M & \text{if } p = 2 \text{ and } r > 0, \\ 2\sum_{i=r}^{M} x^i & \text{if } p \neq 2 \text{ and } r > 0. \end{cases}$$

Note that the coefficient of x^t in $\Phi_{\overline{r}}(x)$ is equal to the coefficient of x^{t-rn} in $\Psi_{\overline{r}}(x) := g_{\overline{r}}(x)^n$, where

$$g_{\overline{r}}(x) := \begin{cases} f_M(x) & \text{if } r = 0, \\ 2\sum_{i=0}^{M-1-r} x^i + x^{M-r} & \text{if } p = 2 \text{ and } r > 0, \\ 2\sum_{i=0}^{M-r} x^i & \text{if } p \neq 2 \text{ and } r > 0. \end{cases}$$

In particular, we have that

$$\left| \mathcal{S}_{t,p^s}^{(n)}(\overline{r}) \right| = \mathrm{coeff}\left[\Psi_{\overline{r}}(x), x^{t-rn}\right].$$

Hence, the sizes of the spheres are computed, respectively, as

$$\left| \mathcal{S}_{t,p^s}^{(n)}(\underline{r}) \right| = \mathrm{coeff}\left[\Phi_{\underline{r}}, x^t\right],$$
$$\left| \mathcal{S}_{t,p^s}^{(n)}(\overline{r}) \right| = \mathrm{coeff}\left[\Psi_{\overline{r}}(x), x^{t-rn}\right].$$

Using Lemma 3.3.1, we get the following asymptotic behavior of restricted Lee-spheres.

Corollary 7.2.2. *Let $T \in [0, M)$ and $t = t(n)$ be a function of n such that $t(n) := Tn$ for large n. Then,*

1. *for $p \neq 2$ or $r < M$, we get*

$$\lim_{n\to\infty} \frac{1}{n} \log_{p^s} \left(\left| \mathcal{S}^{(n)}_{t,p^s}(\underline{r}) \right| \right) = \log_{p^s}(f_{\underline{r}}(\rho)) - T \log_{p^s}(\rho),$$

where ρ is the unique real positive solution of $2\sum_{i=1}^{r}(i-T)x^i = T$ and

$$f_{\underline{r}}(\rho) = 1 + 2\sum_{i=1}^{r} \rho^i = \frac{r(\rho+1)+1}{(1-\rho)(r-T)+1},$$

2. *for $p = 2$ and $r = M$, respectively $r' = 0$, we get*

$$\lim_{n\to\infty} \frac{1}{n} \log_{p^s} \left(\left| \mathcal{S}^{(n)}_{t,p^s}(\underline{r}) \right| \right) = \log_{p^s}(f_{\underline{r}}(\rho)) - T \log_{p^s}(\rho),$$

where ρ is the unique real positive solution of $2\sum_{i=1}^{M-1}(i-T)x^i + (M-T)x^M = T$ and

$$\begin{aligned} g_{\underline{r}}(\rho) = f_{\underline{r}}(\rho) &= 1 + 2\sum_{i=1}^{M-1} \rho^i + \rho^M \\ &= \frac{\rho^{M+1}(T-M) + \rho^M(T-M+1) + \rho(T-M) + T + M + 1}{\rho(T-M) + M + 1 - T}. \end{aligned}$$

Proof. We apply Lemma 3.3.1 to the generating function $\Phi_{\underline{r}}(x)$ and obtain the mentioned results, similar to $r = M$ for $f_{\underline{r}}$ case proved in [130, Lemma 2.6]. □

Corollary 7.2.3. *Let $T \in [0, M)$ and $t = t(n)$ be a function of n such that $t(n) := Tn$ for large n. Then,*

1. *for $p = 2$ and $0 < r \leq T$, we get*

$$\lim_{n\to\infty} \frac{1}{n} \log_{p^s} \left(\left| \mathcal{S}^{(n)}_{t,p^s}(\overline{r}) \right| \right) = \log_{p^s}(g_{\overline{r}}(\rho)) - (T-r) \log_{p^s}(\rho),$$

where ρ is the unique real positive solution of

$$2\sum_{i=1}^{M-1-r} (i-T+r)x^i + (M-T)x^{M-r} = 2(T-r),$$

and

$$g_{\overline{r}}(\rho) = 2\sum_{i=0}^{M-1-r} \rho^i + \rho^{M-r} = \frac{\rho^{M-r+1} + \rho^{M-r} - 2}{\rho - 1},$$

2. *for $p \neq 2$ and $0 < r \leq T$, we get*

$$\lim_{n\to\infty} \frac{1}{n} \log_{p^s} \left(\left| \mathcal{S}^{(n)}_{t,p^s}(\overline{r}) \right| \right) = \log_{p^s}(g_{\overline{r}}(\rho)) - (T-r) \log_{p^s}(\rho),$$

where ρ is the unique real positive solution of $2\sum_{i=1}^{M-r}(i-T+r)x^i = 2(T-r)$ and

$$g_{\overline{r}}(\rho) = 2\sum_{i=0}^{M-r} \rho^i = \frac{2\rho^{M-r+1} - 2}{\rho - 1}.$$

Proof. We apply Lemma 3.3.1 to the generating function $\Psi_{\overline{r}}(x)$ and obtain the mentioned results. □

Remark 7.2.4. Note that, for p odd (respectively, even), we get $T \geq M(M+1)/(2M+1)$ (respectively, $T \geq M/2$) if and only if

$$\lim_{n\to\infty} \frac{1}{n} \log_{p^s} \left(\left| \mathcal{B}^{(n)}_{Tn,p^s} \right| \right) = 1.$$

Hence, if $0 < R$, then a code that attains the asymptotic Gilbert-Varshamov bound has

$$\lim_{n\to\infty} \frac{1}{n} \log_{p^s} \left(\left| \mathcal{B}^{(n)}_{Tn,p^s} \right| \right) = 1 - R < 1,$$

and we immediately get that $T < M(M+1)/(2M+1)$ if p is odd, or $T < M/2$ if p is even.

7.3 Restricted Lee-BJMM Algorithm

We are now going to present an adapted version of the Lee-BJMM algorithm. The idea is to make use of the marginal distribution presented in Lemma 5.1.4 of the entries of a vector $e \in (\mathbb{Z}/p^s\mathbb{Z})^n$ drawn uniformly at random. This means that with high probability we can assume that the less probable Lee weights lie outside the information set. Hence, the erroneous positions lie in a restricted subset of $\mathbb{Z}/p^s\mathbb{Z}$.

We are interested in algorithms that have as input a code generated by a matrix chosen uniformly at random. Due to the result in [32, Proposition 16], we are therefore allowed to assume that our code is free, i.e., $k = K$ and a generator matrix and a parity-check matrix have up to permutations of columns the following form

$$G = \begin{pmatrix} \mathbb{I}_k & A \end{pmatrix}, \quad H = \begin{pmatrix} \mathbb{I}_{n-k} & B \end{pmatrix},$$

where $A \in (\mathbb{Z}/p^s\mathbb{Z})^{k\times(n-k)}$ and $B \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k)\times k}$. In addition, in [32, Theorem 20] it was shown that such a random code also attains the Gilbert-Varshamov in the Lee metric (Theorem 3.4.5) bound with high probability.

Let $\mathcal{C} \subseteq (\mathbb{Z}/p^s\mathbb{Z})^n$ be a linear code with parity-check matrix H, then for an $x \in (\mathbb{Z}/p^s\mathbb{Z})^n$ we say that $s = xH^\top$ is a *syndrome*. The aim is to solve the Lee-syndrome decoding problem below which was shown to be NP-complete in [130].

Problem 7.3.1 (Lee Syndrome Decoding Problem)**.** Given $H \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k)\times n}$, $t \in \mathbb{N}$ and $s \in (\mathbb{Z}/p^s\mathbb{Z})^{n-k}$, find $e \in (\mathbb{Z}/p^s\mathbb{Z})^n$ such that $s = eH^\top$ and $\mathrm{wt_L}(e) = t$.

To this end, we assume that the input parity-check matrix H is chosen uniformly at random in $(\mathbb{Z}/p^s\mathbb{Z})^{(n-k)\times n}$ and that there exists a solution $e \in (\mathbb{Z}/p^s\mathbb{Z})^n$, which was chosen uniformly at random in the n-dimensional sphere of Lee-radius t over $\mathbb{Z}/p^s\mathbb{Z}$, $\mathcal{S}^{(n)}_{t,p^s}$, and set s to be its syndrome $s = eH^\top$. We provide two new algorithms, taking care of two different scenarios. The main idea of these new algorithms is to use the results of [16], which provide us with additional information on the unique solution $e \in (\mathbb{Z}/p^s\mathbb{Z})^n$. For example, the expected number of entries of e having a fixed Lee weight.

In the first scenario, Section 7.3.1, we want to decode up to the minimum distance of the code having H as parity-check matrix. For this, we let $\mathrm{d_L}(\mathcal{C})$ be the minimum distance from the Gilbert-Varshamov bound. With this even if we assume full distance decoding, i.e., $t = \mathrm{d_L}(\mathcal{C})$, we expect to have a unique solution e to Problem 7.3.1 for large n. In fact, the expected number of solutions to the Lee syndrome decoding problem is given by

$$N = \frac{\left|\mathcal{S}^{(n)}_{t,p^s}\right|}{p^{s(n-k)}} = \frac{\left|\mathcal{S}^{(n)}_{\mathrm{d_L}(\mathcal{C}),p^s}\right|}{p^{s(n-k)}} \leq 1.$$

In the second scenario given in Section 7.3.2, we consider a fixed Lee weight t which is beyond the minimum distance, and solve this new problem by reversing the idea of the first algorithm. For a vector $e \in (\mathbb{Z}/p^s\mathbb{Z})^n$ of given Lee weight t we are interested in its Lee weight decomposition. Let us therefore recap the definition of an integer composition. For a given integer m a *weak integer composition* of k of length n is an n-tuple $\lambda = (\lambda_1, \ldots, \lambda_n)$ of nonnegative integers satisfying

$$\lambda_1 + \cdots + \lambda_n = k.$$

We can think of the Lee weight decomposition of a vector $e \in (\mathbb{Z}/p^s\mathbb{Z})^n$ with $\mathrm{wt_L}(e)$ as a weak integer composition $\lambda = (\lambda_1, \ldots, \lambda_n)$ of t of length n such that $\lambda_i = \mathrm{wt_L}(e_i)$. Note that the maximal part size of λ is then restricted by M.

A weak composition π of a positive integer v of length n is said to *fit* into a weak composition λ of a positive integer t of the same length n, if the part sizes of π are upper bounded by the part sizes of λ, i.e., for every $i \in \{1, \ldots, n\}$ it holds $\pi_i \leq \lambda_i$. Owing to the aim of decomposing the Lee weight, let us denote by $C(v, t, \lambda, n, p^s)$ the set of all weak compositions of v of length n fitting into the weak decomposition λ of π of length n, where additionally the part sizes of λ are at most M. For a given composition λ of t, let m denote the maximal part size, i.e.,

$$m = \max\{\lambda_i \,|\, i \in \{1, \ldots, n\}\}\,.$$

7.3.1 Bounded Minimum Distance Decoding

Consider here the scenario where there exists a unique solution to the Lee syndrome decoding problem. This is, we introduce an error of weight t, where t is bounded by the Gilbert-Varshamov bound 3.4.5. Normalizing the weight t by the length n of the error vector e, the Gilbert-Varshamov bound in the Lee metric directly implies that $t/n < \frac{M}{2}$. Hence, as n grows large, zero is the most likely Lee weight to occure in e, followed by elements of Lee weight 1 and so forth. The maximum Lee weight $M = \lfloor p^s/2 \rfloor$ in $(\mathbb{Z}/p^s\mathbb{Z})^n$ is the least likely. Defining a threshold Lee weight $r \in [0, M]$, we assume that with high probability all entries of e of Lee weight larger than r lie outside the information set. Note that this assumption is justified by Theorem 5.1.6. Thus, using the partial Gaussian elimination algorithms, we are left with finding a smaller error vector, which only takes values in $\{0, \pm 1, \ldots, \pm r\}$. This will make a huge difference for algorithms such as the Lee-metric BJMM [130], where the list sizes are the main factor in the cost and these can now be immensely reduced.

Let us consider a random instance of the Lee syndrome decoding problem given by

$$H \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k)\times n}\,,\ s \in (\mathbb{Z}/p^s\mathbb{Z})^{n-k} \quad \text{and} \quad t \in \mathbb{N} \text{ with } t/n < M/2.$$

The framework takes as input $(H, s, t, r, \mathcal{S})$, where $\mathcal{S}$ denotes a solver for the smaller instance in the space $\{0, \pm 1, \ldots, \pm r\}$, which instead of outputting a list of possible solutions for the smaller instance immediately checks whether the smaller solution at hand leads to a solution of the original instance. The framework on $(H, s, t, r, \mathcal{S})$ works as follows:

1. For some $0 \leq \ell \leq n - k$, we bring the parity-check matrix into partial systematic form by multiplying H with some invertible $U \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k)\times(n-k)}$ and adapting the syndrome accordingly to $s' = sU^\top$. For simplicity, assume that we have an information set in the last k positions. Thus, the Lee syndrome decoding problem becomes

$$\begin{pmatrix} e_1 & e_2 \end{pmatrix} \begin{pmatrix} \mathrm{Id}_{n-k-\ell} & 0 \\ A^\top & B^\top \end{pmatrix} = \begin{pmatrix} s_1 & s_2 \end{pmatrix},$$

   with $A \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k-\ell)\times(k+\ell)}$, $B \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell\times(k+\ell)}$, $s_1 \in (\mathbb{Z}/p^s\mathbb{Z})^{n-k-\ell}$ and $s_2 \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell}$. Thus, we have to solve two parity-check equations:

$$\begin{aligned} e_1 + e_2 A^\top &= s_1, \\ e_2 B^\top &= s_2. \end{aligned} \tag{7.1}$$

   Here, we assume that e_2 has Lee weight v and e_1 has Lee weight $t - v$, for some positive integer $0 \leq v \leq t$.

2. We solve the smaller instance of the Lee syndrome decoding problem given by Equation (7.1) using algorithm $\mathcal{S}$. In particular, we find an error vector e_2 such that $e_2 B^\top = s_2$, $\mathrm{wt}_{\mathsf{L}}(e_2) = v$, and it has entries in $\{0, \pm 1, \ldots, \pm r\}$. Instead of storing a list of solutions e_2, $\mathcal{S}$ will immediately check whether $e_1 = s_1 - e_2 A^\top$ has the remaining Lee weight $t - v$. Clearly, v will also depend on the choice of r.

Solving the smaller instance can be achieved using various techniques, for example via Wagner's approach used in [40, 130] or via the representation technique used in [130]. However, we have to slightly adapt these techniques to make use of the assumption that the entries are restricted to $\{0, \pm 1, \ldots, \pm r\}$.

Recall, that we $\mathcal{S}^{(n)}_{v,p^s}(\underline{r})$ denote the Lee-sphere of weight $v \in \mathbb{N}$ centered at the origin with entries restricted to $\{0, \pm 1, \ldots, \pm r\}$, i.e.,

$$\mathcal{S}^{(n)}_{v,p^s}(\underline{r}) := \{x \in \{0, \pm 1, \ldots, \pm r\}^n | \, \mathrm{wt_L}\,(x) = v \, \}.$$

In the following lemma, we show that if e is a random vector of length n and Lee weight t which splits as (e_1, e_2) with $e_2 \in \mathcal{S}^{(k+\ell)}_{v,p^s}(\underline{r})$, then e_2 has a uniform distribution in $\mathcal{S}^{(k+\ell)}_{v,p^s}(\underline{r})$.

Lemma 7.3.2. *Let e be chosen uniformly at random in $\mathcal{S}^{(n)}_{t,p^s}$ such that $e = (e_1, e_2)$ with $e_2 \in \mathcal{S}^{(k+\ell)}_{v,p^s}(\underline{r})$. Then e_2 follows a uniform distribution in $\mathcal{S}^{(k+\ell)}_{v,p^s}(\underline{r})$, and henceforth e_1 follows a uniform distribution in $\mathcal{S}^{(n-k-\ell)}_{t-v,p^s}(\underline{r})$.*

Proof. For an arbitrary $e_2 \in \mathcal{S}^{(k+\ell)}_{v,p^s}(\underline{r})$, there are exactly $\left| \mathcal{S}^{(n-k-\ell)}_{t-v,p^s}(\underline{r}) \right|$ possible e that restrict to e_2 in their last $k+\ell$ coordinates. Therefore, if e is chosen uniformly at random, then each e_2 has an equal chance of being chosen from $\mathcal{S}^{(k+\ell)}_{v,p^s}(\underline{r})$. □

As a corollary, we see that this splitting of e comes with a probability of

$$P = \left| \mathcal{S}^{(k+\ell)}_{v,p^s}(\underline{r}) \right| \left| \mathcal{S}^{(n)}_{t-v,p^s} n-k-\ell \right| \left| \mathcal{S}^{(n)}_{t,p^s} \right|^{-1}.$$

Lee-BJMM Algorithm with Small Weights

Let us consider an adaption of the Lee-BJMM algorithm from [130], where two levels were the optimal choice and proved to remain the optimal choice also for this new algorithm. Although the smaller error vector e_2 now only has entries in $\{0, \pm 1, \ldots, \pm r\}$, to enable representation technique, we will assume that such a vector e_2 is built from the sum of two vectors $y_1 + y_2$, where ε many of their positions cancel out and thus are allowed to live in the whole ring $\mathbb{Z}/p^s\mathbb{Z}$. Let us denote these positions by $\mathcal{E}$. We also denote the symmetric group of size n by S_n.

Let us recall the high level idea of BJMM on two levels.
First, we split e_2 as

$$e_2 = y_1 + y_2 = (x_1^{(1)}, x_2^{(1)}) + (x_1^{(2)}, x_2^{(2)}).$$

Thus, for the syndrome equation to be satisfied, we want that

$$s_2 = e_2 B^\top = y_1 B^\top + y_2 B^\top.$$

Let us also split $B \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell \times (k+\ell)}$ into two matrices $B = \begin{pmatrix} B_1 & B_2 \end{pmatrix}$, where $B_i \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell \times (k+\ell)/2}$, for $i \in \{1, 2\}$. Then in a first merge to get $y_i = (x_1^{(i)}, x_2^{(i)})$ we want for $i = 1$, that they give the syndrome 0, i.e.,

$$x_1^{(1)} B_1^\top = -x_2^{(1)} B_2^\top,$$

and for $i = 2$ that they give the syndrome s_2, i.e.,

$$x_1^{(2)} B_1^\top = s_2 - x_2^{(2)} B_2^\top.$$

This choice is motivated by the fact that each partial syndrome is equally likely. Let us split $\mathcal{E}$ evenly into two disjoint index sets, i.e., $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$ such that $| \, \mathcal{E}_1 \, | = | \, \mathcal{E}_2 \, |$ and $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$. The base lists $\mathcal{B}_i$ for $i \in \{1, 2\}$ are then built as follows

$$\mathcal{B}_i = \left\{ \nu(x) \mid x_{\mathcal{E}_i^C} \in \{0, \ldots, \pm r\}^{(k+\ell-\varepsilon)/2}, x_{\mathcal{E}_i} \in (\mathbb{Z}/p^s\mathbb{Z})^{\varepsilon/2}, \mathrm{wt_L}(x_{\mathcal{E}_i^C}) = v/4, \nu \in S_{(k+\ell)/2} \right\}.$$

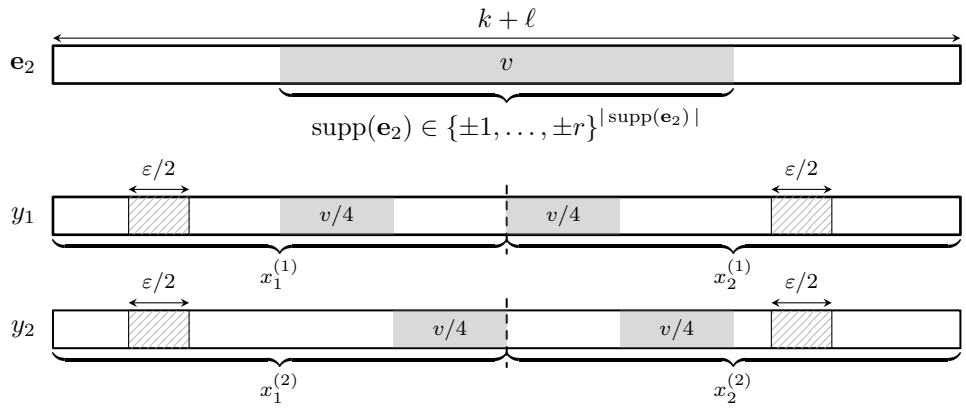

FIGURE 7.1: Illustration of two levels decomposition of the vector e_2 into y_1 and y_2, where $y_i = (x_1^{(i)}, x_2^{(i)})$ for $i = 1, 2$. The gray areas denote the support of the vectors and the values inside the area are the corresponding Lee weights.

For some positive integer $u \leq n$ and $x, y \in (\mathbb{Z}/p^s\mathbb{Z})^n$, we write $x =_u y$, to denote that $x = y$ in the last u positions. Let us define the following two sets.

$$\mathcal{L}_1 = \Big\{\mu(y) \mid y_{\mathcal{E}^C} \in \{0, \ldots, \pm r\}^{k+\ell-\varepsilon},\ y_{\mathcal{E}} \in (\mathbb{Z}/p^s\mathbb{Z})^{\varepsilon},\ \mathrm{wt_L}(y_{\mathcal{E}^C}) = v/2,\ yB^\top =_u 0,\ \mu \in S_{k+\ell}\Big\},$$

$$\mathcal{L}_2 = \Big\{\mu'(y) \mid y_{\mathcal{E}^C} \in \{0, \ldots, \pm r\}^{k+\ell-\varepsilon},\ y_{\mathcal{E}} \in (\mathbb{Z}/p^s\mathbb{Z})^{\varepsilon},\ \mathrm{wt_L}(y_{\mathcal{E}^C}) = v/2,\ yB^\top =_u s_2,\ \mu' \in S_{k+\ell}\Big\}.$$

Performing a concatenation merge (see Algorithm 2), we compute $y_i = (x_1^{(i)}, x_2^{(i)})$ for $(x_1^{(i)}, x_2^{(i)}) \in \mathcal{B}_1 \times \mathcal{B}_2$ on the syndromes 0 and s_2 and u positions. Hence, to get $y_1 \in \mathcal{L}_1$, we merge $y_1 = (x_1^{(1)}, x_2^{(1)})$, such that

$$x_1^{(1)} B_1^\top =_u -x_2^{(1)} B_2^\top,$$

and to get $y_2 \in \mathcal{L}_2$, we merge $y_2 = (x_1^{(2)}, x_2^{(2)})$, such that

$$x_1^{(2)} B_1^\top =_u s_2 - x_2^{(2)} B_2^\top.$$

We then merge $\mathcal{L}_1 \bowtie \mathcal{L}_2$ as shown in Algorithm 3 on the syndrome s_2 and ℓ positions, computing $e_2 = y_1 + y_2$, for $(y_1, y_2) \in \mathcal{L}_1 \times \mathcal{L}_2$ such that the positions $\mathcal{E}$ of y_1 and y_2 cancel out, i.e., $y_{1\mathcal{E}} + y_{2\mathcal{E}} = 0$ and $\mathrm{wt_L}(e_2) = v$.

Remark 7.3.3. Note that our base lists, as well as the lists $\mathcal{L}_i$ employ a permutation. Hence, it might happen that the $\mathcal{E}$ positions are not equal for y_1 and y_2, and these positions might not cancel out. However, the algorithm still succeeds, since we check within the merge, that $y_i \in \mathcal{L}_i$ have the correct weight v. The only implication for the workfactor is that the success probability in this case would even be larger, thus we are giving an upper bound on the cost.

Asymptotic Complexity Analysis

We now present the merging algorithms and their asymptotic costs. Since the cost depends on the sizes of the list $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{L}_1$ and $\mathcal{L}_1$ which are partly defined over the restricted spheres, we use the results presented in Section 7.2 and expected values of the quantities given in Lemma 7.2.1.

We fix the real numbers V, L, E, U with

$$0 \leq V \leq \min\{T, \varphi(r, t, n, p^s)\},\quad 0 \leq L \leq 1 - R,\quad 0 < E < R + L,$$

such that $0 \leq T - 2V \leq M(1 - R - L)$ and $0 < U < L$. Then we fix the internal algorithm parameters and v, ℓ, ε, u which we see as functions depending on n, such that

$$\lim_{n\to\infty} \frac{v}{n} = V, \ \lim_{n\to\infty} \frac{\ell}{n} = L, \ \lim_{n\to\infty} \frac{\varepsilon}{n} = E \ \text{ and } \ \lim_{n\to\infty} \frac{u}{n} = U.$$

Algorithm 2 Merge-concatenate

Require: The lists $\mathcal{B}_1, \mathcal{B}_2$, the positive integers $0 \leq u \leq \ell$, $B_1, B_2 \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell\times(k+\ell)/2}$ and $t \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell}$.
Ensure: $\mathcal{L} = \mathcal{B}_1 \boxplus_t \mathcal{B}_2$.
1: Lexicographically sort $\mathcal{B}_1$ according to the last u positions of $x_1 B_1^\top$ for $x_1 \in \mathcal{B}_1$. We also store the last u positions of $x_1 B_1^\top$ in the sorted list.
2: **for** $x_2 \in \mathcal{B}_2$ **do**
3: **for** $x_1 \in \mathcal{B}_1$ with $x_1 B_1^\top =_u t - x_2 B_2^\top$ **do**
4: $\mathcal{L} = \mathcal{L} \cup \{(x_1, x_2)\}$.
5: **end for**
6: **end for**
7: Return $\mathcal{L}$.

Lemma 7.3.4 ([130, Lemma 4.3]). *The asymptotics of the average cost of Algorithm 2 is*

$$\lim_{n\to\infty} \frac{1}{n} \max\left\{\log_{p^s}(|\mathcal{B}_1|), \log_{p^s}(|\mathcal{B}_2|), \log_{p^s}(|\mathcal{B}_1|) + \log_{p^s}(|\mathcal{B}_2|) - U\right\}.$$

From this we get the lists

$$\mathcal{L}_1 = \mathcal{B}_1 \boxplus_0 \mathcal{B}_2, \ \mathcal{L}_2 = \mathcal{B}_1 \boxplus_{s_2} \mathcal{B}_2.$$

The second merge should not only merge to the target vector s_2, it should also check the Lee weight of the merged vector $y_1 + y_2$ and also the Lee weight of the remaining error vector $e_1 = s_1 - (y_1 + y_2)A^\top$.

Algorithm 3 Last Merge

Require: The input lists $\mathcal{L}_1, \mathcal{L}_2$, the positive integers $0 \leq v \leq t, 0 \leq u \leq \ell$, $B \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell\times(k+\ell)}$, $s_2 \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell}$ and $s_1 \in (\mathbb{Z}/p^s\mathbb{Z})^{n-k-\ell}$, $A \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k-\ell)\times(k+\ell)}$.
Ensure: $e \in \mathcal{L}_1 \bowtie \mathcal{L}_2$.
1: Lexicographically sort $\mathcal{L}_1$ according to $y_1 B^\top$ for $y_1 \in \mathcal{L}_1$. We also store $y_1 B^\top$ in the sorted list.
2: **for** $y_2 \in \mathcal{L}_2$ **do**
3: **for** $y_1 \in \mathcal{L}_1$ with $y_1 B^\top = s_2 - y_2 B^\top$ **do**
4: **if** $\mathrm{wt}_{\llcorner} y_1 + y_2 = v$ and $\mathrm{wt}_{\llcorner} s_1 - (y_1 + y_2)A^\top = t - v$ **then**
5: Return $(s_1 - (y_1 + y_2)A^\top, y_1 + y_2)$.
6: **end if**
7: **end for**
8: **end for**

Corollary 7.3.5 ([130, Corollary 2]). *The asymptotic average cost of the last merge (Algorithm 3) is given by*

$$\lim_{n\to\infty} \frac{1}{n} \max\left\{\log_{p^s}(|\mathcal{L}_1|), \log_{p^s}(|\mathcal{L}_2|), \left(\log_{p^s}(|\mathcal{L}_1|) + \log_{p^s}(|\mathcal{L}_2|)\right) - (L - U)\right\}.$$

Note that the $L - U$ comes from the fact that the vectors already merge to s_2 on U positions due to the first merge. Also, it might happen that $y_1 + y_2$ results in a vector of Lee weight v, but the $\mathcal{E}$ positions did not cancel out, or the positions of low Lee weight are going above the threshold r. This will not be a problem for us, as this only results in a larger final list, which does not need to be stored and the success probability of the algorithm would then even be larger than P.

The way we choose u, is such that we ensure that there exists at least one representative $y_1 \in \mathcal{L}_1$ of the solution e_2, i.e., such that there exists $y_2 \in \mathcal{L}_2$ with $y_1 + y_2 = e_2$. Thus, we have to compute the expected total number of such representatives for a fixed e_2. From Lemma 7.3.2, we know that e_2 follows a uniform distribution in $\mathcal{S}_{v,p^s}^{(k+\ell)}(\underline{r})$.

Using the marginal distribution in (5.5) and (5.7), we can compute the expected Lee weight distribution for e_2. Let λ be the expected Lee weight composition of e_2, and σ be the expected support size of e_2. Also recall that for a weak composition λ of v, we denote by $C(v/2, v, \lambda, k+\ell, p^s)$ the number of weak compositions π of $v/2$ which fit into a composition λ of length $k+\ell$, i.e., the maximal part sizes are given by λ.

Lemma 7.3.6. *The expected number of representatives $(y_1, y_2) \in \mathcal{L}_1 \times \mathcal{L}_2$ for a fixed solution e_2 is at least given by*

$$C(v/2, v, \lambda, k+\ell, p^s)\binom{k+\ell-\sigma}{\varepsilon}(p^s-1)^{\varepsilon},$$

where λ is the expected Lee weight composition of e_2, and σ is the expected support size of e_2.

Proof. Consider the Lee weight composition of e_2 to be $\lambda = (\lambda_1, \ldots, \lambda_{k+\ell})$, which is such that $\lambda_i = \mathrm{wt}_{\mathsf{L}}((e_2)_i)$. Thus, $e_2 = (s_1\lambda_1, \ldots, s_{k+\ell}\lambda_{k+\ell})$, for $s_i \in \{1, -1\}$. Then, to get all possible representatives y_1, we need the number of weak compositions π of $v/2$ fitting into λ. In fact, for any $\pi = (\pi_1, \ldots, \pi_{k+\ell})$ fitting into λ, there exist exactly one eligible y_1 with $\mathrm{wt}_{\mathsf{L}}((y_1)_i) = \pi_i$ and $(y_1)_i = s_i\pi_i$. Note that the Lee weight composition of $y_2 \in \mathcal{L}_2$ is then

$$\mid \lambda - \pi \mid = (\mid \lambda_1 - \pi_1 \mid, \ldots, \mid \lambda_{k+\ell} - \pi_{k+\ell} \mid).$$

On the other hand, for any representative y_1, we cannot have $\pi_i = \mathrm{wt}_{\mathsf{L}}((y_1)_i) > \mathrm{wt}_{\mathsf{L}}((e_2)_i)$ and $(y_1)_i = -s_i\pi_i$ for any $i \in \{1, \ldots, \sigma\}$. In fact, let us assume we have A many positions in y_1 which are such that $\pi_i = \mathrm{wt}_{\mathsf{L}}((y_1)_i) > \mathrm{wt}_{\mathsf{L}}((e_2)_i) = \lambda_i$. Then due to the entry-wise additivity of the Lee weight, we have that y_2, with composition $\mid \lambda - \pi \mid$, has $\mathrm{wt}_{\mathsf{L}}(y_2) > v/2$: in the considered A positions we have that $\mathrm{wt}_{\mathsf{L}}((y_2)_j) = \pi_j - \lambda_j$ and the Lee weight of the remaining $\sigma - A$ positions is given by $\mathrm{wt}_{\mathsf{L}}((y_2)_j) = \lambda_j - \pi_j$, which if we sum over all positions gives

$$\begin{aligned}
\mathrm{wt}_{\mathsf{L}}(y_2) &= \sum_{j=1}^{A}(\pi_j - \lambda_j) + \sum_{j=A+1}^{k+\ell}(\lambda_j - \pi_j) \\
&= \sum_{j=1}^{A}(\pi_j - \lambda_j) + v - \sum_{j=1}^{A}\lambda_j - \left(v/2 - \sum_{j=1}^{A} -\pi_j\right) \\
&= v/2 + 2\left(\sum_{j=1}^{A}\pi_j - \lambda_j\right) \neq v/2.
\end{aligned}$$

It is easy to see, that for each fixed π, there exists only one representative y_1, which has in each position the same sign as e_2.

Recall that $C(v/2, v, \lambda, k+\ell, p^s)$ denotes the number of weak compositions π of $v/2$ which fit into λ. Now, since y_1 can take any non-zero value on the ε positions outside the support of e_2, we get the claim. Finally, the exact number of representations might even be larger than this, since a solution e_2 might also be formed from positions $\mathcal{E}$ which will not cancel out, as assumed for this computation. □

In order to ensure the existence of at least one representative $y_1 \in \mathcal{L}_1$ of e_2, we now choose

$$u = \left\lfloor \log_{p^s}\left(C(v/2, v, \lambda, k+\ell, p^s)\binom{k+\ell-\sigma}{\varepsilon}(p^s-1)^{\varepsilon}\right)\right\rfloor.$$

Thus, in the asymptotic cost we need to compute $U = \lim_{n\to\infty} u/n$. Again we use the saddle point technique for the asymptotic growth rate. Then, the asymptotics of $C(v, t, \lambda, n, p^s)$ is summarized in Lemma 7.3.7 below.

Lemma 7.3.7. *Let us consider a weak composition $\lambda = (\lambda_1, \ldots, \lambda_n)$ of t and the asymptotic relative Lee weight $T := \lim_{n\to\infty} t(n)/n$. In addition, let us consider a positive integer $v \leq t$ with $V := \lim_{n\to\infty} v(n)/n$. Let $m = \max\{\lambda_i \mid i \in \{1, \ldots, n\}\}$. If $0 \leq V < M$, then*

$$\lim_{n\to\infty} \frac{1}{n} \log_{p^s}(C(v, t, \lambda, n, p^s)) = \log_{p^s}(f(\rho)) - V \log_{p^s}(\rho),$$

where ρ is the unique real positive solution of

$$\sum_{i=1}^{m} c_i \frac{z + 2z^2 + \cdots + iz^i}{1 + z + \cdots + z^i} = V.$$

Proof. Note the generating function of the set of compositions $C(v, t, \lambda, n, p^s)$ is given by

$$\Phi(z) := \prod_{i=1}^{n} \left(\sum_{j=0}^{\lambda_i} z^j \right) = \prod_{i=1}^{m} \left(\sum_{j=0}^{i} z^j \right)^{c_i n},$$

where c_i corresponds to the multiplicity of i in the composition λ, i.e., there are $c_i n$ entries of $e \in (\mathbb{Z}/p^s\mathbb{Z})^n$ which have Lee weight i. Thus, $\Phi(z) = f(z)^n$, for

$$f(z) = \prod_{i=1}^{m} \left(\sum_{j=0}^{i} z^j \right)^{c_i}.$$

To get the asymptotics of $C(v, t, \lambda, n, p^s)$ we are interested in the coefficient of z^v in $\Phi(z)$. For this, let us define the asymptotic relative decomposition

$$V := \lim_{n\longrightarrow\infty} v(n)/n.$$

Now, using the saddle point technique of [60], we define $\Delta(f(z)) := \frac{zf'(z)}{f(z)}$. Let ρ be the unique positive real solution to $\Delta(f(z)) = V$, then

$$\lim_{n\to\infty} \frac{1}{n} \log_{p^s} \left(C(v, t, \lambda, n, p^s)\right) = \log_{p^s}(f(\rho)) - V \log_{p^s}(\rho).$$

□

Let us denote the asymptotics of the binomial coefficient by

$$\begin{aligned} H(F, G) &:= \lim_{n\to\infty} \frac{1}{n} \log_{p^s} \left(\binom{f(n)}{g(n)} \right) \\ &= F \log_{p^s}(F) - G \log_{p^s}(G) - (F - G) \log_{p^s}(F - G), \end{aligned}$$

where $f(n), g(n)$ are integer-valued functions such that $\lim_{n\to\infty} \frac{f(n)}{n} = F$ and $\lim_{n\to\infty} \frac{g(n)}{n} = G$. By Lemma 7.3.7, we have computed

$$\gamma(v/2) = \lim_{n'\to\infty} \frac{1}{n'} \log_{p^s} \left(C(v/2, v, \lambda, n', p^s)\right).$$

For us $n' = k + \ell$, which also tends to infinity for n going to infinity. Thus,

$$\lim_{n\to\infty} \frac{k+\ell}{n} \lim_{k+\ell\to\infty} \frac{1}{k+\ell} \log_{p^s} \left(C(v/2, v, \lambda, k+\ell, p^s)\right) = (R + L)\gamma(v/2).$$

Then,

$$U = (R + L)\gamma(v/2) + H(R + L - S, E) + E,$$

where $S = \lim_{n\to\infty} \sigma/n$.

Algorithm 4 Lee-BJMM with Small Balls

Require: $H \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k)\times n}$, $s \in (\mathbb{Z}/p^s\mathbb{Z})^{n-k}$, $t \in \mathbb{N}$, and the positive integers ℓ, v, ε satisfying $0 \leq \ell \leq n-k$, $0 \leq v \leq t$ and $0 \leq \varepsilon \leq k+\ell$.
Ensure: A vector $e \in (\mathbb{Z}/p^s\mathbb{Z})^n$ satisfying $s = eH^\top$ and $\mathrm{wt}_{\mathsf{L}}(e) = t$.

1: Choose an $n \times n$ permutation matrix P and an invertible matrix $U \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k)\times(n-k)}$ such that
$$UHP = \begin{pmatrix} \mathrm{Id}_{n-k-\ell} & A \\ 0 & B \end{pmatrix},$$
where $A \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k-\ell)\times(k+\ell)}$ and $B \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell\times(k+\ell)}$.
2: Compute
$$sU^\top = \begin{pmatrix} s_1 & s_2 \end{pmatrix},$$
where $s_1 \in (\mathbb{Z}/p^s\mathbb{Z})^{n-k-\ell}$ and $s_2 \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell}$.
3: Choose a set $\mathcal{E} \subset \{1, \ldots, k+\ell\}$ of size ε.
4: Build the lists $\mathcal{B}_1, \mathcal{B}_2$ as
$$\mathcal{B}_i = \{x \mid x_{\mathcal{E}^C} \in \{0, \pm 1, \ldots, \pm r\}^{(k+\ell-\varepsilon)/2}, x_{\mathcal{E}} \in (\mathbb{Z}/p^s\mathbb{Z})^{\varepsilon/2}, \mathrm{wt}_{\mathsf{L}}(x_{\mathcal{E}^C}) = v/4\}.$$
5: Compute $\mathcal{L}_1 = \mathcal{B}_1 \boxplus_0 \mathcal{B}_2$ and $\mathcal{L}_2 = \mathcal{B}_1 \boxplus_{s_2} \mathcal{B}_2$.
6: Compute $e \in \mathcal{L}_1 \bowtie \mathcal{L}_2$.
7: If this fails, return to Step 1.
8: Return $P^\top e$.

To ease the notation, we denote the asymptotics of the restricted Lee-metric sphere by

$$A^{(n)}_{t,p^s}(\underline{r}) \coloneqq \lim_{n\to\infty} \frac{1}{n} \log_{p^s}\left(\left|\mathcal{S}^{(n)}_{t(n),p^s}(\underline{r})\right|\right).$$

Recall that $A^{(n)}_{t,p^s}(\underline{r})$ is computed according to Corollary 7.2.2. Furthermore, let us denote by $W = \psi(r,t,n,p^s)/n$.

Theorem 7.3.8. *The asymptotic average time complexity of the Lee-metric BJMM algorithm on two levels is at most given by $I + C$, where*

$$I = A^{(n)}_{t,p^s}(M) - A^{(k+\ell)}_{t-v,p^s}(\underline{r}) - A^{(n-k-\ell)}_{t-v,p^s}(M),$$

is the expected number of iterations and $C = \max\{B, 2D - L + U, D\}$ is the expected cost of one iteration with

$$B = A^{((k+\ell-\varepsilon)/2)}_{v/4,p^s}(\underline{r}) + H((R+L)/2, E/2) + E/2,$$
$$D = A^{(k+\ell-\varepsilon)}_{v/2,p^s}(\underline{r}) + H(R+L, E) + E - U.$$

In addition, we have an expected memory of at most $\mathcal{M} = \max\{B, D\}$. On a capable quantum computer, the average time complexity is given by at most

$$I/2 + \max\left\{B, D, \frac{1}{2}(2D - L + U)\right\}.$$

Proof. For our base lists $\mathcal{B}_i$, we have that

$$|\,\mathcal{B}_i\,| = \left|\mathcal{S}^{((k+\ell-\varepsilon)/2)}_{v/4,p^s}(\underline{r})\right| \binom{(k+\ell)/2}{\varepsilon/2} (p^s-1)^{\varepsilon/2}.$$

Due to Lemma 7.2.2, the cost of the first merge is then given by

$$B = A^{((k+\ell-\varepsilon)/2)}_{v/4,p^s}(\underline{r}) + H((R+L)/2, E/2) + E/2.$$

For the second merge we also need to compute the asymptotic sizes of $\mathcal{L}_i$. First, we note that

$$|\mathcal{L}_i| = \frac{\left|\mathcal{S}^{(k+\ell-\varepsilon)}_{v/2,p^s}(\underline{r})\right| \binom{k+\ell}{\varepsilon}(p^s-1)^\varepsilon}{p^{su}}.$$

Thus,

$$D = \lim_{n\to\infty}\frac{1}{n}\log_{p^s}(|\mathcal{L}_i|) = A^{(k+\ell-\varepsilon)}_{v/2,p^s}(\underline{r}) + H(R+L,E) + E - U.$$

Using Corollary 7.3.5, the second merge costs asymptotically

$$2D - L + U = 2A^{(k+\ell-\varepsilon)}_{v/2,p^s}(\underline{r}) + 2H(R+L,E) + 2E - U - L.$$

We recall that the success probability of the algorithm is given by P, hence for $0 \le V \le \varphi(r,t,n,p^s)/n$, we get the following asymptotic number of iterations

$$A^{(n)}_{t,p^s}(M) - A^{(k+\ell)}_{t-v,p^s}(\underline{r}) - A^{(n-k-\ell)}_{t-v,p^s}(M).$$

The average memory required for the algorithm is given by $|\mathcal{B}_i|$ and $|\mathcal{L}_i|$, thus taking the asymptotics of these lists the claim follows. Finally, note that Grover's algorithm can be used to speed up on a capable quantum computer whenever a list L has to be searched. In particular, instead of $\mathcal{O}(|L|)$, Grover's algorithm only requires $\mathcal{O}(\sqrt{|L|})$ operations. Thus, this results asymptotically in $\lim_{n\to\infty}\frac{1}{2n}\log_{p^s}(|L|)$. In our classical asymptotic cost, every term stems from a searched list, except for B and D, which are intermediate lists that have to be stored in full. □

Observe that ℓ, v, r, ε are internal parameters, which can be chosen optimal, i.e., such that the algorithm achieves the minimal cost. Clearly, the choice for the threshold r will influence the possible choices for v.

Amortized Version

If we only consider p^{su} many vectors from the base lists $\mathcal{B}_i$, we could potentially reduce the cost and memory.

The algorithm is going to work exactly the same way, with the only difference that the base lists $\mathcal{B}'_i$ have size p^{su}. Thus, after using the merging Algorithm 2 on u positions we get lists $\mathcal{L}'_i$ of size p^{su} as well. Finally, we merge these lists using Algorithm 3 on ℓ positions. Note that the conditions on $U = \lim_{n\to\infty} u(n)/n$ are

$$L/3 \le U \le \min\{(R+L)\gamma(v/2) + H(R+L-S,E) + E, B, L\},$$

where B denotes the asymptotic size of the original base lists, i.e.,

$$B = A^{((k+\ell-\varepsilon)/2)}_{v/4,p^s}(\underline{r}) + H((R+L)/2, E/2) + E/2.$$

The condition $L/3 \le U$, comes from the size of the final list, i.e., the number of solutions for the smaller instance, which is $\frac{p^{s2u}}{p^{s(\ell-u)}} = p^{s(3u-\ell)}$. In order to have at least one solution, we require $3u \ge \ell$. Recall that $(R+L)\gamma(v/2) + H(R+L-S,E) + E$ denotes the asymptotic number of representations, thus the condition $U \le (R+L)\gamma(v/2) + H(R+L-S,E) + E$ is the same as for the original algorithm. The condition $U \le B$, as well as $U \le L$ are straightforward.

Note that in the amortized case, the success probability of splitting $e = (e_1, e_2)$ is not simply given by

$$P = \left|\mathcal{S}^{(k+\ell)}_{v,p^s}(\underline{r})\right| \left|\mathcal{S}^{(n-k-\ell)}_{t-v,p^s}\right| \left|\mathcal{S}^{(n)}_{t,p^s}\right|^{-1}$$

as in the non-amortized case, since our list of e_2 is by construction smaller. That is instead of all solutions to the smaller problem $\left|\mathcal{S}^{(k+\ell)}_{v,p^s}(\underline{r})\right| p^{-s\ell}$, we only consider Z many solutions to

the smaller problem. In other words, Z is the number of distinct e_2 in our last list. Similar to the approach of [40], we have a success probability of

$$P' = Zp^{s\ell} \left| \mathcal{S}_{t-v,p^s}^{(n-k-\ell)} \right| \left| \mathcal{S}_{t,p^s}^{(n)} \right|^{-1}.$$

In order to compute Z, let us denote by X the maximal amount of collisions of the last merge which would lead to an e_2 (that is with possible repetitions), by Y the total number of solutions to $e_2 B^\top = s_2$ with $e_2 \in \mathcal{S}_{v,p^s}^{(k+\ell)}(\underline{r})$, namely

$$Y = \left| \mathcal{S}_{v,p^s}^{(k+\ell)}(\underline{r}) \right| p^{-s\ell},$$

and finally by W the number of collisions that we are considering, that is

$$W = p^{s(3u-\ell)} = p^{s2u} p^{-s(\ell-u)}.$$

This leaves us with a combinatorial problem: having a basket with X balls having Y colors, if we pick W balls at random, how many colors are we going to see on average? This will determine the number of distinct tuples e_2 in the final list. This number is on average

$$Y \left(1 - \binom{X - X/Y}{W} \binom{X}{W}^{-1} \right),$$

which can be lowerbounded by W. In fact,

$$\begin{aligned} 1 - \binom{X - X/Y}{W} \binom{X}{W}^{-1} &= 1 - \frac{(X - X/Y + 1 - W) \cdots (X - W)}{(X - X/Y + 1) \cdots X} \\ &\geq 1 - (1 - W/X)^{X/Y} \sim W/Y. \end{aligned}$$

Hence, $Z \geq p^{s(3u-\ell)}$ and we get a success probability of at least

$$p^{s3u} \left| \mathcal{S}_{t-v,p^s}^{(n-k-\ell)} \right| \left| \mathcal{S}_{t,p^s}^{(n)} \right|^{-1}.$$

The asymptotic cost of the amortized version of Algorithm 4 is then given by $I' + \max\{U, 3U - L\}$, where I' is the expected number of iterations, i.e.,

$$I' \leq A_{t,p^s}^{(n)}(M) - 3U - A_{t-v,p^s}^{(n-k-\ell)}(M).$$

Hence, we can see that the restriction to the smaller balls does not influence the amortized version of BJMM, as the idea of amortizing is already to restrict the balls. The restriction only influences the conditions and thus the possible choices of U.

7.3.2 Decoding Beyond the Minimum Distance

There could be scenarios where one wants to decode more errors than the minimum Lee distance of the code at hand allows. In the classical case, i.e., in the Hamming metric, the cost can then be divided by the expected number of solutions N. This follows from the fact that for each of the N solutions we have a success probability P for one iteration to succeed. Assuming that the solutions are independent, this implies that to find one solution we expect the number of iterations to be $\frac{1}{PN}$.

In a scenario where we have $t > Mn/2$, the marginal distribution of $e \in (\mathbb{Z}/p^s\mathbb{Z})^n$ implies that $\pm M$ is the most likely entry of e, then the second most likely is $\pm(M-1)$ and so on, until the least likely entry is 0. In this case, we will reverse the previous algorithm and for some threshold Lee weight $0 \leq r \leq M$, we want the vector e_2 of Lee weight $t - \varphi(r-1, t, n, p^s) \leq v \leq t$ to live in $\{\pm r, \ldots, \pm M\}^{k+\ell}$. In order to construct such a vector, we will use a similar construction as before, where we exchange the set $\{0, \pm 1, \ldots, \pm r\}$ with $\{\pm r, \ldots, \pm M\}$. Thus, we similarly denote the n-dimensional Lee-sphere of weight $v \in \mathbb{N}$ centered at the origin with

entries restricted to $\{\pm r, \ldots, \pm M\}$ by

$$\mathcal{S}^{(n)}_{v,p^s}(\overline{r}) := \{x \in \{\pm r, \ldots, \pm M\}^n | \, \mathrm{wt_L}\,(x) = v\}.$$

Note that the success probability of such splitting is now given by

$$P = \left| \mathcal{S}^{(k+\ell)}_{v,p^s}(\overline{r}) \right| \left| \mathcal{S}^{(n-k-\ell)}_{t-v,p^s} \right| \left| \mathcal{S}^{(n)}_{t,p^s} \right|^{-1}.$$

Let us first illustrate the idea and then compute the sizes of the lists involved. Note,

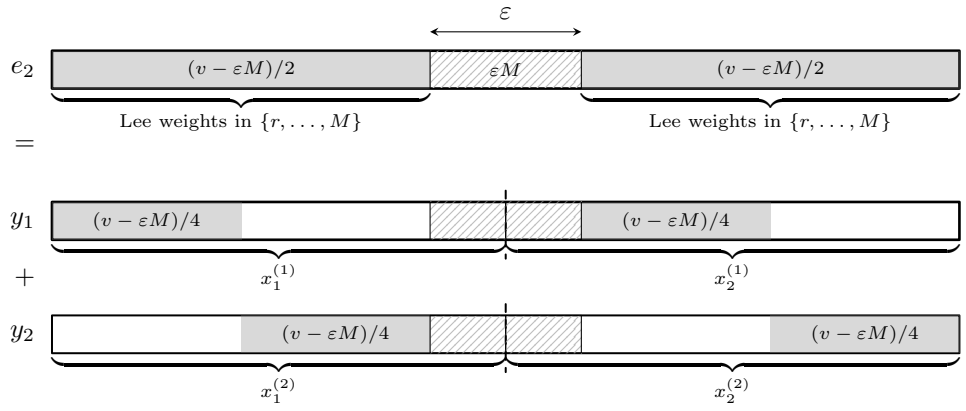

FIGURE 7.2: Illustration of two levels decomposition of the vector e_2 into y_1 and y_2, where $y_i = (x_1^{(i)}, x_2^{(i)})$ for $i = 1, 2$. The gray areas denote the support of the vectors and the values inside the area are the corresponding Lee weights. For $(y_1)_i$ and $(y_2)_i$ with $i \in \mathcal{E}$, we require $\mathrm{wt_L}\,(y_1)_i + (y_2)_i = M$.

that one of the main differences to the previous algorithm is that we require to partition the weights in order to guarantee that the large weight entries of y_1 will not be decreased after adding y_2. For this let us introduce the following set of indices $Z_1, Z_2, W_1, W_2, \mathcal{E}_1, \mathcal{E}_2$ satisfying

$$\begin{gathered} |\, Z_i \,| = |\, W_i \,| = (k + \ell - \varepsilon)/4, \\ Z_1 \cap Z_2 = W_1 \cap W_2 = \mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset, \text{ and} \\ Z_i \cap W_i \cap \mathcal{E}_i = \emptyset \text{ for } i \in \{1, 2\}. \end{gathered}$$

Let us denote their union by

$$\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2, Z = Z_1 \cup Z_2, W = W_2 \cup W_2.$$

For $i \in \{1, 2\}$, the base lists $\mathcal{B}_i$ are then given by

$$\begin{aligned} \mathcal{B}_i = \Big\{ \nu_i(x) \;\big|\; & x_{Z_i} \in \{0\}^{(k+\ell-\varepsilon)/4}, x_{W_i} \in \{\pm r, \ldots, \pm M\}^{(k+\ell-\varepsilon)/4}, x_{\mathcal{E}_i} \in (\mathbb{Z}/p^s\mathbb{Z})^{\varepsilon/2}, \\ & \mathrm{wt_L}(x_{W_i}) = (v - \varepsilon M)/4, \nu_i \in S_{(k+\ell)/2} \Big\}. \end{aligned}$$

All the base lists have the same size, which is given by

$$\binom{(k+\ell)/2}{\varepsilon/2} p^{s\varepsilon/2} \binom{(k+\ell-\varepsilon)/2}{(k+\ell-\varepsilon)/4} F^{(r)}\big((v - \varepsilon M)/4, (k + \ell - \varepsilon)/4, p^s\big).$$

Performing the concatenation merge of Algorithm 2, we build $\mathcal{L}_1$ and $\mathcal{L}_2$ from $\mathcal{B}_1$ and $\mathcal{B}_2$ as

$$\mathcal{L}_1 = \Big\{\mu_1(y_1) \mid \mu_1 \in S_{k+\ell}, y_1 B^\top =_u \mathbf{0}, (y_1)_Z \in \{0\}^{(k+\ell-\varepsilon)/2}, (y_1)_\mathcal{E} \in (\mathbb{Z}/p^s\mathbb{Z})^\varepsilon, \\ (y_1)_W \in \{\pm r, \dots, \pm M\}^{(k+\ell-\varepsilon)/2}, \mathrm{wt}_\mathsf{L}((y_1)_W) = (v-\varepsilon M)/2\Big\},$$

$$\mathcal{L}_2 = \Big\{\mu_2(y_2) \mid \mu_2 \in S_{k+\ell}, y_2 B^\top =_u \mathbf{s}_2, (y_2)_Z \in \{0\}^{(k+\ell-\varepsilon)/2}, (y_2)_\mathcal{E} \in (\mathbb{Z}/p^s\mathbb{Z})^\varepsilon, \\ (y_2)_W \in \{\pm r, \dots, \pm M\}^{(k+\ell-\varepsilon)/2}, \mathrm{wt}_\mathsf{L}((y_2)_W) = (v-\varepsilon M)/2\Big\}.$$

Both lists are of size

$$\binom{k+\ell}{\varepsilon} p^{s(\varepsilon-u)} \binom{k+\ell-\varepsilon}{(k+\ell-\varepsilon)/2} \left| \mathcal{S}^{((k+\ell-\varepsilon)/2)}_{(v-\varepsilon M)/2, p^s}(\overline{r}) \right|.$$

For this procedure to work, we also need the additional condition on v, r and ε, that

$$v \geq \varepsilon(M-r) + r(k+\ell).$$

Then, a final merge using Algorithm 3 will produce a final list of all smaller solutions of the smaller instance which does not require to be stored.

Lemma 7.3.9. *The number of representations $e_2 = y_1 + y_2$ for $(y_1, y_2) \in \mathcal{L}_1 \times \mathcal{L}_2$ is then given by at least*

$$R_B = \binom{k+\ell}{\varepsilon} \left(\sum_{i=0}^{\varepsilon} \binom{\varepsilon}{i} (M-r+1)^i r^{\varepsilon-i} \binom{\varepsilon'}{i} (M-r+1)^i \binom{\varepsilon'-i}{(\varepsilon'-i)/2} \right),$$

for $\varepsilon' = k+\ell-\varepsilon$.

Proof. To give a lower bound on the number of representations it is enough to give one construction.
The overall idea of this construction is to split the $\mathcal{E}_1$ positions of y_1 and $\mathcal{E}_2$ positions of y_2 into those parts where they overlap and those parts where they do not overlap. In the parts where $\mathcal{E}_1$ does not overlap with $\mathcal{E}_2$, we can only allow small Lee weights in y_1 such that, by adding large Lee weight entries of y_2, we can still reach the large Lee weight entries of e_2.

So let us consider a fixed $e_2 \in \mathcal{S}^{(k+\ell)}_{v,p^s}(\overline{r})$. As a first step we fix the $\mathcal{E}_1$ positions which gives $\binom{k+\ell}{\varepsilon}$. Then, within the $\mathcal{E}_1$ position we fix those of small Lee weight. This means for a fixed position we can assume that the entry in e_2 is a with $r \leq \mathrm{wt}_\mathsf{L}(a) \leq M$. Small Lee weights of y_1 now refer to the possible values of y_1 in this position such that a can be reached through large Lee weight entries of y_2. That is, for example if $a = r$, we allow in y_1 the entries $\{0, -1, \dots, r-M\}$, or if $a = M$ we allow in y_1 the entries $\{M-r, \dots, 0\}$. These allowed sets of small Lee weight always have size $M-r+1$, independently of the the value a. Thus, in $\mathcal{E}_1$ of size ε we choose i entries of small Lee weight, which give $\binom{\varepsilon}{i}(M-r+1)^i$ many choices. For the remaining $\varepsilon - i$ positions in $\mathcal{E}_1$ we have large Lee weights in y_1, which cannot reach the large Lee weight entries of e_2 through large Lee weight entries in y_2. Thus, they must come for the $\mathcal{E}_2$ positions. In these entries we have $r^{\varepsilon-i}$ possible choices. Note that out of the ε many positions of $\mathcal{E}_2$ we have only assigned $\varepsilon - i$ many. Hence, as a next step we choose of the remaining $k+\ell-\varepsilon$ positions the remaining i positions to have small Lee weight in y_2. Thus, the fixed large Lee weight entries of e_2 can be reached by adding these positions to large Lee weight entries of y_1. For this we have $\binom{k+\ell-\varepsilon-i}{i}(M-r+1)^i$ possibilities. As a final step we then partition the remaining positions to be either zero or of large Lee weight, i.e., $\binom{k+\ell-\varepsilon-i}{(k+\ell-\varepsilon-i)/2}$. □

Thus, we will need the additional condition $\varepsilon \leq (k+\ell)/2$, and we choose

$$u = \left\lfloor \log_{p^s}(R_B) \right\rfloor. \tag{7.2}$$

Asymptotic Complexity Analysis

At this point, let us discuss the asymptotic complexity of our ISD variant when decoding beyond the minimum Lee distance. Since we cannot take the asymptotics of an infinite sum, we need to bound the quantity in (7.2). In fact, setting $i = \varepsilon$ gives such lower bound.

$$R_B \geq \binom{k+\ell}{\varepsilon}(M-r+1)^{2\varepsilon}\binom{k+\ell-\varepsilon}{\varepsilon}\binom{k+\ell-2\varepsilon}{(k+\ell-2\varepsilon)/2}.$$

Then,

$$\begin{aligned} U = \lim_{n\to\infty} U(n)/n =& H(R+L,E) + 2E\log_{p^s}(M-r+1) \\ &+ H(R+L-E,E) + H(R+L-2E,(R+L-2E)/2). \end{aligned}$$

In addition, since we decode beyond the minimum distance, the Lee syndrome decoding problem has several solutions. Since the inputs have been chosen uniform at random, we can assume that these solutions are independent of each other. Thus, to find just one of all the expected

$$N = \frac{\left|\mathcal{S}^{(n)}_{t,p^s}\right|}{p^{s(n-k)}}$$

solutions we have an expected number of iterations given by $(NP)^{-1}$, instead of P^{-1}. Note that asymptotically this value is bounded by R, as

$$X = \lim_{n\to\infty}\frac{1}{n}\log_{p^s}\left(\left|\mathcal{S}^{(n)}_{t,p^s}\right| p^{-s(n-k)}\right) = A^{(n)}_{t,p^s}(M) - 1 + R \leq R.$$

Let us denote by $A^{(n)}_{t,p^s}(\overline{r}) = \lim_{n\to\infty}\frac{1}{n}\log_q\left(\left|\mathcal{S}^{(n)}_{t,p^s}(\overline{r})\right|\right)$. Using Corollary 7.2.3 we then deduce the asymptotic average time complexity of the restricted Lee-BJMM algorithm when decoding beyond the minimum Lee distance.

Corollary 7.3.10. *The asymptotic average time complexity of the Lee-metric BJMM algorithm on two levels for $t > Mn/2$ is given by at most $I + C$, where*

$$I = (1-R) - A^{(k+\ell)}_{v,p^s}(\overline{r}) - A^{(n-k-\ell)}_{t-v,p^s}(M)$$

is the expected number of iterations and $C = \max\{B, D, 2D - L + U\}$ is the cost of one iteration, where

$$\begin{aligned} B =& E/2 + H((R+L)/2, E/2) + H((R+L-E)/2, (R+L-E)/4) \\ &+ A^{((k+\ell-\varepsilon)/4)}_{(v-\varepsilon p^2/2)/4,p^s}(\overline{r}), \\ D =& E - U + H(R+L,E) + H(R+L-E,(R+L-E)/2) \\ &+ A^{((k+\ell-\varepsilon)/2)}_{(v-\varepsilon p^2/2)/2,p^s}(\overline{r}). \end{aligned}$$

In addition, we have an expected memory of at most $\mathcal{M} = \max\{B, D\}$. On a capable quantum computer, the average time complexity is given by at most $I/2 + \max\{B, D, \frac{1}{2}(2D - L + U)\}$.

Amortized Version

We consider again the amortized version of this algorithm, i.e., we only take p^{su} many vectors from the base lists $\mathcal{B}^{(1)}_i$, respectively $\mathcal{B}^{(2)}_i$.

The algorithm is going to work exactly the same way, similar to the amortized version for the first scenario. The asymptotic cost of the amortized version of Algorithm 4 is then given by $I' + \max\{U, 3U - L\}$, where I' is as before the expected number of iterations, i.e.,

$$I' \leq (1-R) - 3U - A^{(n-k-\ell)}_{t-v,p^s}(M).$$

7.4 Comparison to other Lee Metric ISD Algorithms

In this section we want to see how much cost reduction we were able to achieve by using this additional information on the error vector. For this we compare the new Lee-metric BJMM algorithm to the Lee-metric BJMM algorithm from [130] and to the algorithm using Wagner's approach in [40], which were until now the fastest algorithms to solve the Lee syndrome decoding problem. We denote by $e(R, p^s)$ the exponent of the asymptotic cost and compare $e(R^*, p^s)$ for $R^* = \arg \max_{0 \leq R \leq 1} (e(R, p^s))$.

In the first scenario, we only decode up to the Gilbert-Varshamov bound, i.e., we consider $\mathcal{B}^{(n)}_{d(n),p^s} = 1 - R$. Hence, we give an immediate relation between $T := \lim_{n\to\infty} d(n)/n$ and R, i.e., we are considering full-distance decoding.

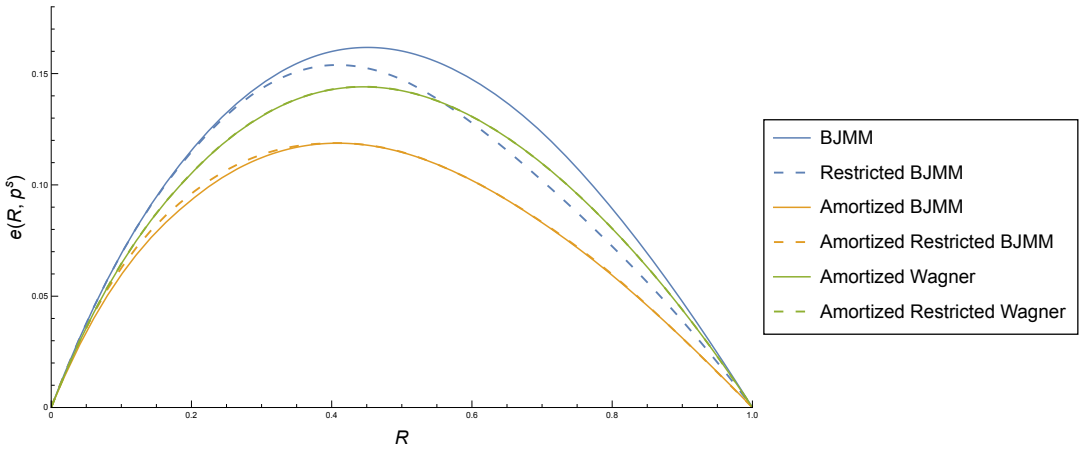

FIGURE 7.3: Comparison of asymptotic costs of full-distance decoding of different algorithms and their restricted versions, for $p = 47, s = 1$ and assuming the asymptotic Gilbert-Varshamov bound.

Algorithm	$e(R^*, p^s)$	R^*
Lee-BJMM	0.1618	0.451
Restricted Lee-BJMM for $r = 5$	0.1539	0.408
Amortized Lee-BJMM	0.1205	0.396
Amortized Restricted Lee-BJMM	0.1189	0.406
Amortized Lee-Wagner	0.1441	0.445
Amortized Restricted Lee-Wagner	0.1441	0.445

TABLE 7.1: Comparison of asymptotic costs for full-distance decoding for $p^s = 47$.

In the second scenario, where we have $N > 1$ solutions, one possible technique proposed in [40] is to fix a rate $R \in \{0.1, \ldots, 0.9\}$ and go through all $M/2 \leq T \leq M$, to see at which T the largest cost is attained for this fixed rate. However, this approach gives for the algorithm in [40] as well as for our algorithm always $T = M$. This is a very particular weight, where e will only have entries $\pm M$. The problem of decoding such instance is then a completely different one from the original problem and more like a binary syndrome decoding problem. As the algorithm in [40] and also our algorithm work for any large T, they will clearly not be suitable for this special scenario.

Another possible technique is the following: the asymptotic value for N is given by

$$X = A^{(n)}_{t,p^s}(M) - 1 + R \leq R,$$

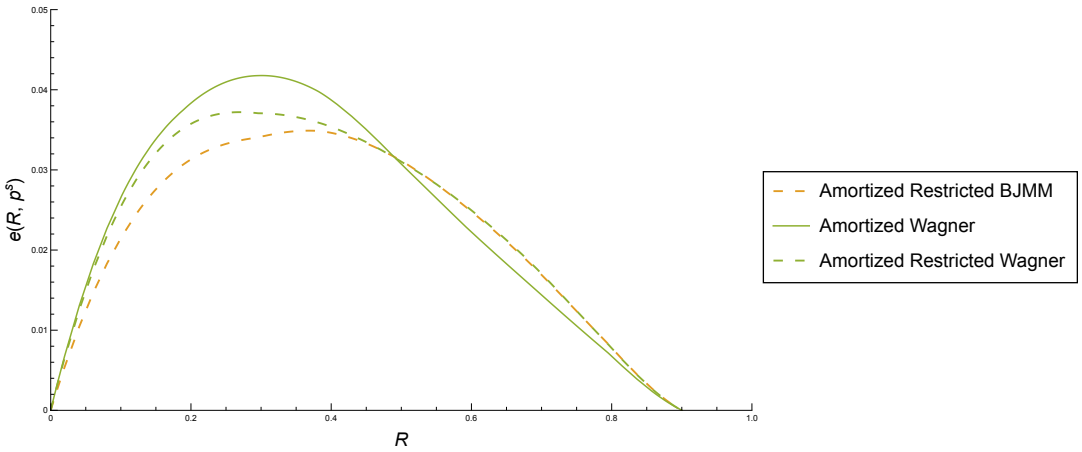

FIGURE 7.4: Comparison of asymptotic costs of decoding beyond the minimum distance of different algorithms and their restricted versions, for $q = 47$.

thus we can fix X to be a function in R, e.g. $X = R/2$. This will also directly lead to a $T = \lim_{n\to\infty} t(n)/n$, for which $A^{(n)}_{t,p^s}(M) = 1 - R/2$. If we had fixed X to be a constant independent of R instead, this would have obstructed the comparison for all rates smaller than this constant. To compare the asymptotic costs of several algorithms we then determine the rate for which the cost is maximal. Since there is no other non-amortized algorithm which considers the second case, we will only compare our amortized version with the algorithm provided in [40].

We observed that in the second case, where we decode beyond the minimum distance, ε is very small. Note that ε was introduced in [76] to increase the number of positions on which we can merge u. In our algorithm, however, u can be chosen very large, in fact, very close to ℓ, even for $\varepsilon = 0$. Thus, $\varepsilon > 0$ would only increase the size of the lists. We also want to note here that the program we are considering in Figure 7.4 takes the minimum of the cost of our algorithm and the cost of brute forcing. For this note that we fixed the number of solutions to be $p^{s(k/2)}$, thus going through all vectors e of weight t we expect to find a solution after $\left|\mathcal{S}^{(n)}_{t,p^s}\right| p^{-s(k/2)}$ many steps, that has an asymptotic cost of $A^{(n)}_{t,p^s}(M) - R/2 = 1 - R$. On the other hand, we might go through all solutions of the parity-check equations, which are p^{sk} many and expect to find a solution after $p^{s(k-k/2)}$ many steps, which has an asymptotic cost of $R/2$.

Algorithm	$e(R^*, q)$	R^*
Amortized Restricted Lee-BJMM	0.0349	0.368
Amortized Lee-Wagner	0.0418	0.301
Amortized Restricted Lee-Wagner	0.0372	0.270

TABLE 7.2: Comparison of asymptotic cost of different Lee metric ISD algorithms for $p = 47, s = 1$ beyond the minimum distance.

Remark 7.4.1. This approach can work for any metric and ambient space, as long as the distribution of the error vector allows us to solve the smaller instance in a smaller space. This might have an impact for the ring-learning with errors problem, since also there the error vector is drawn from a certain distribution, in this case the Gaussian.

7.5 Summary and Outlook

In this chapter, the marginal distribution of vectors of a given Lee weight has been used to improve the fastest information set decoding algorithm for the Lee metric known, i.e., the Lee-BJMM algorithm on two levels. The knowledge about the marginal distribution

allowed us to assume that, almost surely, the erroneous positions lie in a restricted Lee-sphere. That means that the entries in the erroneous positions have a Lee weight that is smaller (or larger) than a given threshold value $r \in [0, \lfloor p^s/2 \rfloor]$ for bounded minimum distance decoding (or decoding beyond the minimum distance). The restricted Lee-spheres then showed their impact on the size of the lists needed to store the possible candidates. As the complexity depends on the size of the lists, using the restricted Lee-spheres induced a slight reduction in the complexity of the Lee-BJMM algorithm. In the bounded minimum distance decoding, compared to an amortized version of the algorithm the novel Lee-BJMM variant does not show any improvement nor does it have a worse complexity. However, when decoding beyond the minimum Lee distance, the restricted-BJMM version presented outperforms the amortized Wagner algorithm especially for code rates up to $1/2$.

As mentioned, the Lee metric has a direct connection to the L^2-norm used in lattice-based cryptography. In the ring-learning with error problem in lattice-based cryptography the error vectors are drawn given a aussian distribution. Therefore, approaches used in the setting of lattices might be used in the Lee metric to further improve algorithms like information set decoding. Lattice-based cryptosystems are promising candidates in the standardization process of post-quantum cryptosystems imposted by the NIST. Hence, using lattice-based tools in the Lee metric could yield to Lee-metric code-based cryptosystems of the same or similar advantages as lattice-based schemes.

Chapter 8

Conclusions and Future Work

In this thesis we studied ring-linear codes in the Lee metric. In a first step, we focused on the algebraic aspects of codes over integer residue rings endowed with the Lee metric. More explicitly, we restricted to chain rings $\mathbb{Z}/p^s\mathbb{Z}$ and introduced the concept of generalized weights to the Lee metric. Adapting the existing theory in the Hamming metric to the Lee metric is not always straightforward. In fact, instead of defining the Lee-support of a vector as a set of indices, we represent the Lee-support in terms of a tuple storing the Lee weights of its entries. To extend the Lee-support from a vector to a code, we used three different approaches focussing on the maximum Lee weight, the minimum nonzero Lee weight and the maximum weight in a given column of a generator matrix, respectively. Eventually, we abandoned the classical idea of deriving a Singleton-like bound using generalized Lee weights and focused on the chain structure of the underlying chain ring $\mathbb{Z}/p^s\mathbb{Z}$. We introduced generalized Lee distances using the natural chain of inclusion of $\mathbb{Z}/p^s\mathbb{Z}$ together with more parameters on a generator matrix in systematic form. Finally, we derived a bound on the minimum Lee distance of a code which clearly outperforms all other bounds for most parameters.

One main motivation to study the Lee metric is its increasing interest in applications to code-based cryptography. Indeed, the NP-hardness of the underlying syndrome decoding problem and the ability to reduce the key size when using the Lee metric instead of the Hamming metric, makes Lee-metric codes a promising candidate for post-quantum cryptography. In code-based cryptosystems errors of a fixed weight are intentionally introduced to a codeword. In order to understand errors of fixed Lee weight, we introduced a block-wise channel model (the constant Lee-weight channel) adding an error vector of fixed weight to the transmitted message. We derived the marginal distribution of this channel model and used it to bound the size of n-dimensional spheres and balls of a given Lee-radius. The marginal distribution states the expected Lee-weight decomposition of vectors over $\mathbb{Z}/q\mathbb{Z}$ of given Lee weight in the limit of large block length. Additionally, we considered a discrete memoryless channel counterpart to the constant Lee-weight channel, referred to as the memoryless Lee channel. We showed that both channels coincide for growing block length. In a finite-length setting, we provided two algorithms to construct vectors of length n over $\mathbb{Z}/q\mathbb{Z}$ with given Lee weight t based on partitioning t. We noticed that the weight of such vectors can be increased or decreased by multiplying the vector component-wise with a suitable nonzero constant $a \in \mathbb{Z}/q\mathbb{Z}$. From a cryptographic viewpoint, reducing the Lee weight of a received word or even error vector would simplify the underlying syndrome decoding problem and could therefore lead to a possible reduction of the security level. We proved that the probability of this scenario is negligible as the length n grows large for any $\mathbb{Z}/q\mathbb{Z}$. For n constant, the same result applies to q being a prime number or a power of 2.

With the introduction of the two channel models, we studied the block error probability of both channels and the error-correction performance of regular low-density parity-check (LDPC) code families over the two channel models under belief propagation and symbol message-passing decoding. We derived the expected weight enumerator of a randomly chosen LDPC code in a regular LDPC ensemble using combinatorial tools and generating functions. This allowed us to understand the error-floor of the LDPC code family and enabled to derive bounds on the error-correction performance. As we restricted to parity-check matrices with nonzero entries lying in the set of units $(\mathbb{Z}/q\mathbb{Z})^\times$, we adapted the decoders accordingly. By means of density evolution and finite-length Monte Carlo simulations this restriction is visible when decoding using symbol message-passing. We showed, however, that belief propagation

and symbol message-passing outperform the Lee symbol flipping decoding algorithm designed for Lee-LDPC codes.

In a last step, we applied the result on the marginal distribution to the yet fastest known Lee-BJMM information set decoding algorithm on two levels. Given the marginal distribution of vectors of fixed Lee weight in the limit of large block length, we restricted the Lee weight of the erroneous positions to a subset of $\mathbb{Z}/q\mathbb{Z}$, optimizing a threshold value of the largest (respectively smallest) Lee weight an erroneous position achieves. This restriction leads to a reduction in the complexity and, hence, to an improvement of the Lee-BJMM variant.

The foundation of Lee-metric codes still shows gaps, when comparing it to other metrics used in coding theory. The study of novel techniques to derive bounds on Lee-metric code parameters is an important task. This thesis might inspire to move away from classical techniques and tools, and to focus more on the algebraic structure of an integer residue ring underlying a code in the Lee metric. Furthermore, the Lee metric is an interesting and promising candidate for code-based cryptography. Its strong connection to lattice-based cryptography might be used to improve code-based schemes or information set decoding in the Lee metric applying lattice-based techniques.

Bibliography

[1] C. Aguilar Melchor, N. Aragon, M. Bardet, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, A. Hauteville, A. Otmani, O. Ruatta, J.-P. Tillich, and G. Zémor. ROLLO- Rank-Ouroboros, LAKE & LOCKER. *NIST PQC Call for Proposals*, 2020.

[2] C. Aguilar Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, M. Bros, A. Couvreur, J.-C. Deneuville, P. Gaborit, A. Hauteville, and G. Zémor. Rank Quasi-Cyclic (RQC). *NIST PQC Call for Proposals*, 2020.

[3] C. Aguilar Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, G. Zémor, and J. Bos. HQC – Submission to the fourth round of the NIST post-quantum project, 2023. https://pqc-hqc.org/doc/hqc-specification_2023-04-30.pdf.

[4] M. R. Albrecht, D. J. Bernstein, T. Chou, C. Cid, J. Gilcher, T. Lange, V. Maram, I. von Maurich, R. Misoczki, R. Niederhagen, K. G. Paterson, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, J. Szefer, C. J. Tjhai, M. Tomlinson, and W. Wang. Classic McEliece – Submission to the fourth round of the NIST post-quantum project, 2023. https://classic.mceliece.org/nist/mceliece-20221023.pdf.

[5] T. L. Alderson and S. Huntemann. On maximum Lee distance codes. *SIAM Journal of Discrete Mathematics*, 2013.

[6] J. Antrobus and H. Gluesing-Luerssen. Maximal Ferrers diagram codes: constructions and genericity considerations. *IEEE Transactions on Information Theory*, 65(10):6204–6223, 2019.

[7] N. Aragon, P. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Guneysu, C. A. Melchor, R. Misoczki, E. Persichetti, N. Sendrier, J.-P. Tillich, G. Zemor, V. Vasseur, S. Ghosh, and J. Richter-Brokmann. BIKE – Submission to the fourth round of the NIST post-quantum project, 2022. https://bikesuite.org/files/v5.0/BIKE_Spec.2022.10.10.1.pdf.

[8] A. Ashikhmin, G. Kramer, and S. ten Brink. Extrinsic information transfer functions: Model and erasure channel properties. *IEEE Transaction of Information Theory*, 50(11):2657–2673, Nov. 2004.

[9] E. Assmus Jr and H. F. Mattson. Error-correcting codes: An axiomatic approach. *Information and Control*, 6(4):315–330, 1963.

[10] J. Astola. On the asymptotic behaviour of Lee-codes. *Discrete Applied Mathematics*, 8(1):13–23, 1984.

[11] M. Baldi, G. Cancellieri, F. Chiaraluce, E. Persichetti, and P. Santini. Using non-binary LDPC and MDPC codes in the McEliece cryptosystem. In *Proc. AEIT International Annual Conference*, Sept. 2019.

[12] M. Baldi, F. Chiaraluce, R. Garello, and F. Mininni. Quasi-cyclic low-density parity-check codes in the McEliece cryptosystem. In *2007 IEEE International Conference on Communications*, pages 951–956. IEEE, 2007.

[13] M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R. Perlner, D. Smith-Tone, J.-P. Tillich, and J. Verbel. Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 507–536. Springer, 2020.

[14] S. Barg. Some new NP-complete coding problems. *Problemy Peredachi Informatsii*, 30(3):23–28, 1994.

[15] J. Bariffi, H. Bartz, G. Liva, and J. Rosenthal. Analysis of Low-Density Parity-Check Codes over Finite Integer Rings for the Lee Channel. In *2022 IEEE Global Communications Conference*, pages 1–6, 2022.

[16] J. Bariffi, H. Bartz, G. Liva, and J. Rosenthal. On the properties of error patterns in the constant Lee weight channel. In *International Zurich Seminar on Information and Communication (IZS 2022), Zurich, Switzerland, March 2–4, 2022*, pages 44–48, 2022.

[17] J. Bariffi, H. Bartz, G. Liva, and J. Rosenthal. Error-Correction Performance of Regular Ring-Linear LDPC Codes over Lee Channels. *arXiv preprint arXiv:2312.14674*, 2023.

[18] J. Bariffi, K. Khathuria, and V. Weger. Information Set Decoding for Lee-Metric Codes using Restricted Balls. In *Code-Based Cryptography: 10th International Workshop, CBCrypto 2022 Trondheim, Norway, May 29–30, 2022 Revised Selected Papers*. Lecture Notes in Computer Science, Springer, 2022.

[19] J. Bariffi and V. Weger. Better bounds on the minimal Lee distance. *arXiv preprint arXiv:2307.06079*, 2023.

[20] H. Bartz and S. Puchinger. Decoding of Interleaved Linearized Reed-Solomon Codes with Applications to Network Coding, 2021.

[21] A. Becker, A. Joux, A. May, and A. Meurer. Decoding random binary linear codes in $2^{n/20}$: How 1+ 1= 0 improves information set decoding. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 520–536. Springer, 2012.

[22] E. Berlekamp, R. McEliece, and H. Van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[23] E. R. Berlekamp. Negacyclic codes for the Lee metric. Technical report, North Carolina State University. Dept. of Statistics, 1966.

[24] E. R. Berlekamp. *Algebraic coding theory*. McGraw Hill, 1968.

[25] D. J. Bernstein, J. Buchmann, and E. Dahmen. *Post-Quantum Cryptography*. Springer, 2009.

[26] D. J. Bernstein, T. Lange, and C. Peters. Smaller decoding exponents: ball-collision decoding. In *Annual Cryptology Conference*, pages 743–760. Springer, 2011.

[27] S. Bhattacharya and A. Banerjee. A method to find the volume of a sphere in the Lee metric, and its applications. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 872–876. IEEE, 2019.

[28] I. F. Blake. Codes over certain rings. *Information and Control*, 20(4):396–404, 1972.

[29] I. F. Blake. Codes over integer residue rings. *Information and Control*, 29(4):295–300, 1975.

[30] L. Boltzmann. Studien über das Gleichgewicht der lebendigen Kraft zwischen bewegten materiellen Punkten (Studies of the equilibrium and the life force between material points). 1868.

[31] D. Burshtein and G. Miller. Asymptotic enumeration methods for analyzing LDPC codes. *IEEE Transactions on Information Theory*, 50(6):1115–1131, 2004.

[32] E. Byrne, A.-L. Horlemann, K. Khathuria, and V. Weger. Density of free modules over finite chain rings. *Linear Algebra and its Applications*, 651:1–25, 2022.

[33] E. Byrne and A. Ravagnani. Partition-balanced families of codes and asymptotic enumeration in coding theory. *Journal of Combinatorial Theory, Series A*, 171:105169, 2020.

[34] E. Byrne and V. Weger. Bounds in the lee metric and optimal codes. *Finite Fields and Their Applications*, 87:102151, 2023.

[35] A. R. Calderbank and N. J. Sloane. Modular and p-adic cyclic codes. *Designs, Codes and Cryptography*, 6:21–35, 1995.

[36] A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.

[37] A. Canteaut and N. Sendrier. Cryptanalysis of the original McEliece cryptosystem. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 187–199. Springer, 1998.

[38] S. D. Cardell, M. Firer, and D. Napp. Generalized column distances. *IEEE Transactions on Information Theory*, 66(11):6863–6871, 2020.

[39] F. Chabaud. Asymptotic analysis of probabilistic algorithms for finding short codewords. In *Eurocode'92*, pages 175–183. Springer, 1993.

[40] A. Chailloux, T. Debris-Alazard, and S. Etinski. Classical and quantum algorithms for generic syndrome decoding problems and applications to the Lee metric. In *International Conference on Post-Quantum Cryptography*, pages 44–62. Springer, 2021.

[41] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. A. Perlner, and D. Smith-Tone. *Report on post-quantum cryptography*, volume 12. US Department of Commerce, National Institute of Standards and Technology, 2016.

[42] J. C.-Y. Chiang and J. K. Wolf. On channels and codes for the Lee metric. *Information and Control*, 19(2):159–173, 1971.

[43] G. Como and F. Fagnani. Average spectra and minimum distances of low-density parity-check codes over abelian groups. *SIAM Journal on Discrete Mathematics*, 23(1):19–53, 2009.

[44] T. Cover. Enumerative source encoding. *IEEE Transactions on Information Theory*, 19(1):73–77, 1973.

[45] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 2nd edition, 2006. chapter 15.

[46] I. Csiszár, P. C. Shields, et al. Information theory and statistics: A tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4):417–528, 2004.

[47] M. C. Davey and D. J. MacKay. Monte carlo simulations of infinite low density parity check codes over GF(q). In *Proc. of Int. Workshop on Optimal Codes and related Topics*, pages 9–15. Citeseer, 1998.

[48] P. Delsarte. Bilinear forms over a finite field, with applications to coding theory. *Journal of Combinatorial Theory, Series A*, 25(3):226–241, 1978.

[49] C. Di. Asymptotic and finite-length analysis of low-density parity-check codes. Technical report, EPFL, 2004.

[50] S. T. Dougherty. *Algebraic coding theory over finite commutative rings*. Springer, 2017.

[51] S. T. Dougherty, M. Gupta, and K. Shiromoto. On generalized weights for codes over finite rings. *preprint*, 2002.

[52] S. T. Dougherty and K. Shiromoto. MDR codes over $\mathbb{Z}_k$. *IEEE Transactions on Information Theory*, 46(1):265–269, 2000.

[53] I. I. Dumer. Two decoding algorithms for linear codes. *Problemy Peredachi Informatsii*, 25(1):24–32, 1989.

[54] M. Finiasz and N. Sendrier. Security bounds for the design of code-based cryptosystems. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 88–105. Springer, 2009.

[55] B. J. Frey and F. R. Kschischang. Probability propagation and iterative decoding. In *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, volume 34, pages 482–493. University of Illinois, 1996.

[56] E. M. Gabidulin. Theory of codes with maximum rank distance. *Problemy peredachi informatsii*, 21(1):3–16, 1985.

[57] R. Gabrys, H. M. Kiah, and O. Milenkovic. Asymmetric Lee distance codes for DNA-based storage. *IEEE Transactions on Information Theory*, 63(8):4982–4995, Aug. 2017.

[58] R. G. Gallager. *Low-Density Parity-Check Codes*. M.I.T. Press, Cambridge, MA, 1963.

[59] R. G. Gallager. *Information Theory and Reliable Communication*. Wiley, New York, 1968.

[60] D. Gardy and P. Solé. Saddle point techniques in asymptotic coding theory. In *Workshop on Algebraic Coding*, pages 75–81. Springer, 1991.

[61] E. N. Gilbert. A comparison of signalling alphabets. *The Bell System Technical Journal*, 31(3):504–522, 1952.

[62] H. Gluesing-Luerssen. On the sparseness of certain linear MRD codes. *Linear Algebra and its Applications*, 596:145–168, 2020.

[63] S. W. Golomb and L. R. Welch. Algebraic coding and the Lee metric. *Error Correcting Codes*, pages 175–194, 1968.

[64] E. Gorla and A. Ravagnani. Generalized weights of codes over rings and invariants of monomial ideals. *arXiv preprint arXiv:2201.05813*, 2022.

[65] E. Gorla and F. Salizzoni. Generalized column distances. *arXiv preprint arXiv:2212.12265*, 2022.

[66] E. Gorla and F. Salizzoni. Generalized weights of convolutional codes. *IEEE Transactions on Information Theory*, 2023.

[67] M. Greferath. An introduction to ring-linear coding theory. In *Gröbner Bases, Coding, and Cryptography*, pages 219–238. Springer, 2009.

[68] A. Gruica and A. Ravagnani. Common complements of linear subspaces and the sparseness of MRD codes. *SIAM Journal on Applied Algebra and Geometry*, 6(2):79–110, 2022.

[69] C. T. Gueye, J. B. Klamti, and S. Hirose. Generalization of BJMM-ISD using May-Ozerov nearest neighbor algorithm over an arbitrary finite field $\mathbb{F}_q$. In *Codes, Cryptology and Information Security*, pages 96–109. Springer International Publishing, 2017.

[70] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950.

[71] A. R. Hammons, P. V. Kumar, A. R. Calderbank, N. J. Sloane, and P. Solé. The Z_4-linearity of Kerdock, Preparata, Goethals, and related codes. *IEEE Transactions on Information Theory*, 40(2):301–319, 1994.

[72] W. K. Hayman. A generalisation of Stirling's formula. *Journal für die reine und angewandte Mathematik*, 1956.

[73] T. Helleseth, T. Kløve, and J. Mykkeltveit. The weight distribution of irreducible cyclic codes with block lengths $n_1((q^\ell - 1)/n)$. *SIAM Journal on Discrete Mathematics*, 18(2):179–211, 1977.

[74] S. Hirose. May-Ozerov Algorithm for Nearest-Neighbor Problem over $\mathbb{F}_q$ and Its Application to Information Set Decoding. In *International Conference for Information Technology and Communications*, pages 115–126. Springer, 2016.

[75] A.-L. Horlemann-Trautmann and V. Weger. Information set decoding in the Lee metric with applications to cryptography. *Advances in Mathematics of Communications*, 15(4):677–699, 2021.

[76] N. Howgrave-Graham and A. Joux. New generic algorithms for hard knapsacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 235–256. Springer, 2010.

[77] X.-Y. Hu, E. Eleftheriou, and D. Arnold. Regular and irregular progressive edge-growth Tanner graphs. *IEEE Transaction of Information Theory*, 51(1):386–398, Jan. 2005.

[78] C. Interlando, K. Khathuria, N. Rohrer, J. Rosenthal, and V. Weger. Generalization of the ball-collision algorithm. *Journal of Algebra Combinatorics Discrete Structures and Applications*, 7(2):195–207, 2018.

[79] S. Jukna. *Extremal combinatorics: with applications in computer science*. Springer Science & Business Media, 2011.

[80] Y. Komamiya. Application of logical mathematics to information theory. *3rd Japanese National Congress on Applied Math*, 437, 1953.

[81] F. Lázaro, A. G. i Amat, G. Liva, and B. Matuz. Symbol message passing decoding of nonbinary low-density parity-check codes. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–5. IEEE, 2019.

[82] G. Lechner, T. Pedersen, and G. Kramer. Analysis and design of binary message passing decoders. *IEEE Transactions on Communications*, 60(3):601–607, 2011.

[83] C. Lee. Some properties of nonbinary error-correcting codes. *IRE Transactions on Information Theory*, 4(2):77–82, 1958.

[84] P. J. Lee and E. F. Brickell. An Observation on the Security of McEliece's Public-Key Cryptosystem. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 275–280. Springer, 1988.

[85] J. S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transactions on Information Theory*, 34(5):1354–1359, Sept. 1988.

[86] H.-A. Loeliger. An upper bound on the volume of discrete spheres. *IEEE Transaction of Information Theory*, 40(6):2071–2073, 1994.

[87] P. Loidreau. Asymptotic behaviour of codes in rank metric over finite fields. *Designs, Codes and Cryptography*, 71:105–118, 2014.

[88] D. J. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–431, 1999.

[89] D. J. MacKay and R. M. Neal. Near Shannon limit performance of low density parity check codes. *Electronics Letters*, 33(6):457–458, 1997.

[90] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*, volume 16. Elsevier, 1977.

[91] J. L. Massey. *Notes on coding theory*. Waltham Research Center, 1969.

[92] A. May, A. Meurer, and E. Thomae. Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 107–124. Springer, 2011.

[93] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *Deep Space Network Progress Report*, 44:114–116, Jan. 1978.

[94] A. Meurer. *A coding-theoretic approach to cryptanalysis*. PhD thesis, Ruhr Universität Bochum, 2013.

[95] A. Neri, A.-L. Horlemann-Trautmann, T. Randrianarisoa, and J. Rosenthal. On the genericity of maximum rank distance and Gabidulin codes. *Designs, Codes and Cryptography*, 86(2):341–363, 2018.

[96] R. Niebuhr, E. Persichetti, P.-L. Cayrel, S. Bulygin, and J. Buchmann. On Lower Bounds for Information Set Decoding over $\mathbb{F}_q$ and on the Effect of Partial Knowledge. *IInternational Journal of Information and Coding Theory*, 4(1):47–78, 2017.

[97] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Prob. Contr. Inform. Theory*, 15(2):157–166, 1986.

[98] G. H. Norton and A. Sălăgean. On the structure of linear and cyclic codes over a finite chain ring. *Applicable Algebra in Engineering, Communication and Computing*, 10:489–506, 2000.

[99] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

[100] C. Peters. Information-set decoding for linear codes over $\mathbb{F}_q$. In *International Workshop on Post-Quantum Cryptography*, pages 81–94. Springer, 2010.

[101] M. S. Pinsker. Information and information stability of random variables and processes. *Holden-Day*, 1964.

[102] G. Poltyrev. Bounds on the decoding error probability of binary linear codes via their spectra. *IEEE Transactions on Information Theory*, 40(4):1284–1292, 1994.

[103] E. Prange. The use of coset equivalence in the analysis and decoding of group codes. *Electronics Research Directorate, Air Force Cambridge Research Center*, 1959.

[104] E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[105] S. Puchinger, J. Renner, and J. Rosenkilde. Generic decoding in the sum-rank metric. *arXiv preprint arXiv:2001.04812*, 2020.

[106] A. Ravagnani. Generalized weights: an anticode approach. *Journal of Pure and Applied Algebra*, 220(5):1946–1962, 2016.

[107] T. Richardson and R. Urbanke. The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding. *IEEE Transaction of Information Theory*, 47(2):599–618, Feb. 2001.

[108] S. Ritterhoff, G. Maringer, S. Bitzer, V. Weger, P. Karl, T. Schamberger, J. Schupp, and A. Wachter-Zeh. FuLeeca: A Lee-based signature scheme. In *CBCrypto 2023: Lecture Notes in Computer Science, Springer*, 2023.

[109] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[110] J. Rosenthal and R. Smarandache. Maximum distance separable convolutional codes. *Appl. Algebra Engrg. Comm. Comput.*, 10(1):15–32, 1999.

[111] R. M. Roth. Maximum-rank array codes and their application to crisscross error correction. *IEEE Transactions on Information Theory*, 37(2):328–336, 1991.

[112] R. M. Roth. Introduction to coding theory. *IET Communications*, 47, 2006.

[113] I. N. Sanov. On the probability of large deviations of random variables. *Selected Translations in Mathematical Statistics and Probability*, 1:213–244, 1961.

[114] P. Santini, M. Battaglioni, F. Chiaraluce, M. Baldi, and E. Persichetti. Low-Lee-Density Parity-Check Codes. In *2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[115] C. Satyanarayana. Lee metric codes over integer residue rings. *IEEE Transactions on Information Theory*, 25(2):250–254, 1979.

[116] B. Segre. Curve razionali normali e k-archi negli spazi finiti. *Annali di Matematica Pura ed Applicata*, 39(1):357–379, 1955.

[117] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[118] M. Shi, A. Alahmadi, and P. Solé. *Codes and rings: theory and practice*. Academic Press, 2017.

[119] K. Shiromoto. Singleton bounds for codes over finite rings. *Journal of Algebraic Combinatorics*, 12(1):95–99, 2000.

[120] P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM review*, 41(2):303–332, 1999.

[121] R. Singleton. Maximum distance q-nary codes. *IEEE Transactions on Information Theory*, 10(2):116–118, 1964.

[122] M. Sipser and D. A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[123] E. Spiegel. Codes over $\mathbb{Z}_m$. *Information and Control*, 35(1):48–51, 1977.

[124] D. Sridhara and T. E. Fuja. LDPC codes over rings for PSK modulation. *IEEE Transaction of Information Theory*, 51(9):3209–3220, Sept. 2005.

[125] J. Stern. A method for finding codewords of small weight. In *International Colloquium on Coding Theory and Applications*, pages 106–113. Springer, 1988.

[126] R. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981.

[127] J. H. Van Lint. *Introduction to Coding Theory*, volume 201. Springer, 1971.

[128] A. Vardy. The Intractability of Computing the Minimum Distance of a Code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, 1997.

[129] V. Weger, M. Battaglioni, P. Santini, F. Chiaraluce, M. Baldi, and E. Persichetti. Information set decoding of Lee-metric codes over finite rings. *arXiv preprint arXiv:2001.08425*, 2020.

[130] V. Weger, K. Khathuria, A.-L. Horlemann, M. Battaglioni, P. Santini, and E. Persichetti. On the hardness of the Lee syndrome decoding problem. *Advances in Mathematics of Communications*, 2022.

[131] V. K. Wei. Generalized Hamming weights for linear codes. *IEEE Transactions on Information Theory*, 37(5):1412–1418, 1991.

[132] N. Wiberg. *Codes and decoding on general graphs*. Department of Electrical Engineering, Linköping University Sweden, 1996.

[133] N. Wiberg, H.-A. Loeliger, and R. Kotter. Codes and iterative decoding on general graphs. *European Transactions on Telecommunications*, 6(5):513–525, 1995.

[134] H. S. Wilf. *generatingfunctionology*. CRC press, 2005.

[135] E. L. Wilmer, D. A. Levin, and Y. Peres. Markov chains and mixing times. *American Mathematical Society, Providence*, 2009.

[136] J. Wood. The structure of linear codes of constant weight. *Transactions of the American Mathematical Society*, 354(3):1007–1026, 2002.

[137] A. D. Wyner and R. L. Graham. An upper bound on minimum distance for a k-ary code. *Information and Control*, 13(1):46–52, 1968.

[138] K. Xie and J. Li. On accuracy of Gaussian assumption in iterative analysis for LDPC codes. In *2006 IEEE International Symposium on Information Theory*, pages 2398–2402. IEEE, 2006.

[139] M. Yadegari. The binomial theorem: a widespread concept in medieval islamic mathematics. *Historia Mathematica*, 7(4):401–406, 1980.