Information Set Decoding in the Lee Metric and the Local to Global Principle for Densities

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

von
Violetta Weger
von
Zürich, ZH

Promotionskommission
Prof. Dr. Joachim Rosenthal (Vorsitz)
Prof. Dr. Andrew Kresch
Prof. Dr. Anna-Lena Horlemann

Zürich, 2020

Für meine Eltern,
Jasmin und Ruedi

Abstract

This thesis consists of two parts: in the first part we focus on information set decoding algorithms and in the second part on the local to global principle for densities.

One of the main candidates for post-quantum cryptography is based on coding theory, more in detail, its security is based on the NP-complete problem of decoding a random linear code. Code-based cryptography first came up with the seminal work of McEliece in 1978. The fastest algorithm to solve this NP-complete problem, and thus to decode a random code, is information set decoding. These algorithms have exponential cost in the input size. Hence they are not considered as attacks on code-based cryptosystems but are rather used as a tool to determine the necessary size of public keys in order to achieve a given security level. The main disadvantage of code-based cryptography is its enormous public key size. Many researchers have tried to tackle this problem, by proposing different families of codes as secret key.

Recently, the community has changed its focus on a different direction: changing the underlying metric of the code. In fact, cryptosystems based on the rank metric achieve remarkably small key sizes. In this part of the thesis we follow this new path for code-based cryptography and provide different information set decoding algorithms in the Lee metric. The Lee metric is very promising as it can correct many more errors than the Hamming metric and in fact our theoretical comparisons confirm that the key size will decrease substantially.

In the second part of this thesis we focus on the natural density, which is the $\mathbb{Z}$-analogue of a uniform probability distribution. This topic belongs to theoretical number theory, and dates back to a question asked by Mertens and Césaro, namely: how likely is it that two randomly chosen integers are coprime? To compute the natural density there exist many different techniques. We will focus on the local to global principle that allows to compute natural densities by characterizing the target set locally, *i.e.*, over the p-adic integers. Using this method, we give an elegant proof that the density of coprime pairs is $\frac{1}{\zeta(2)}$, where ζ denotes the Riemann zeta function. In addition, the set of Eisenstein polynomials and the generalizations of the set of coprime pairs, namely the coprime m-tuples and the rectangular unimodular matrices, are considered.

Furthermore, seeing the natural density as a $\mathbb{Z}$-analogue of a uniform probability distribution the natural task arises to define the mean and the variance corresponding to this density. We use tools of analytic number theory to compute the mean and variance for the four sets of interest. The results all follow the same pattern, which only becomes visible when considering the local to global principle. Hence we are able to give an addendum to the local to global principle, which with few additional conditions guarantees the existence of the mean and the variance and allows to compute them directly in an elegant way.

Kurzfassung

Diese Doktorarbeit besteht aus zwei Teilen: im ersten Teil konzentrieren wir uns auf Decodieralgorithmen mittels Informations-Sets und im zweiten Teil auf das „lokal zu global“ Prinzip für Dichten.

Einer der Hauptkandidaten für Post-Quanten Kryptographie basiert auf Codierungstheorie, genauer gesagt, die Sicherheit basiert auf dem NP-vollständigen Problem des Decodierens eines zufälligen linearen Codes. Codierungs-basierte Kryptographie wurde mit der wegweisenden Arbeit von McEliece im Jahr 1978 eingeführt. Der schnellste Algorithmus, um dieses NP-vollständige Problem zu lösen und damit einen zufälligen Code zu decodieren, ist Decodierung via Informations-Sets. Da diese Algorithmen exponentielle Kosten haben, werden sie nicht als Angriffe auf Codierungs-basierte Kryptosysteme betrachtet, sondern werden vielmehr als Werkzeug verwendet, um die erforderliche Grösse des öffentlichen Schlüssels zu bestimmen, der nötig ist um ein gewisses Sicherheitslevel zu erreichen. Der grösste Nachteil der Codierungs-basierten Kryptographie ist die enorme Grösse dieses öffentlichen Schlüssels. Viele Forscher haben versucht, dieses Problem zu beheben, indem sie verschiedene Familien von Codes als geheimen Schlüssel vorgeschlagen haben. Erst vor kurzem hat sich der Fokus in eine andere Richtung gewendet: das Wechseln der zugrundeliegenden Metrik des Codes. In der Tat erzielen Kryptosysteme, die auf der Rangmetrik basieren, bemerkenswert kleine Schlüsselgrössen. In diesem Teil der Arbeit folgen wir diesem neuen Weg der Codierungs-basierten Kryptographie und präsentieren verschiedene Information-Set Dekodieralgorithmen in der Lee-Metrik. Die Lee-Metrik ist sehr vielversprechend, da sie viel mehr Fehler korrigieren kann als die Hamming-Metrik und in der Tat bestätigen unsere theoretischen Vergleiche, dass sich die Schlüsselgrösse erheblich verringert.

Im zweiten Teil dieser Arbeit konzentrieren wir uns auf die natürliche Dichte, die als Analogon einer gleichmässigen Wahrscheinlichkeitsverteilung über den ganzen Zahlen angesehen werden kann. Dieses Thema gehört zur theoretischen Zahlentheorie und geht auf eine Frage von Mertens und Césaro zurück, nämlich: Wie wahrscheinlich ist es, dass zwei zufällig gewählte ganze Zahlen teilerfremd sind? Um die natürliche Dichte zu berechnen, gibt es viele verschiedene Techniken. Wir werden uns auf das sogenannte „lokal zu global“ Prinzip konzentrieren, welches es ermöglicht, natürliche Dichten durch eine lokale Charakterisierung der Zielmenge zu berechnen. Mit Hilfe dieser Methode geben wir einen eleganten Beweis dafür, dass die Dichte von teilerfremden Paaren $\frac{1}{\zeta(2)}$ beträgt, wobei ζ die Riemannsche Zetafunktion bezeichnet. Zusätzlich werden wir die Menge der Eisenstein-Polynome und die Verallgemeinerungen der Menge der teilerfremden Paare, also die teilerfremden m-Tupel und die rechteckigen unimodularen Matrizen, betrachten. Wenn wir die natürliche Dichte als $\mathbb{Z}$-Analogon einer gleichmäßigen Wahrscheinlichkeitsverteilung betrachten, so stellt sich die natürliche Frage, den Mittelwert und die Varianz, die zu dieser Dichte gehören, zu definieren. Wir verwenden einige Werkzeuge der analytischen Zahlentheorie zur Berechnung des Mittelwerts und der Varianz für die vier Zielmengen. Die Ergebnisse folgen alle demselben Muster, das nur bei Betrachtung des „lokal zu global“ Prinzips sichtbar wird. Daher können wir ein Addendum zum „lokal zu global“ Prinzip erstellen, welches mit nur wenigen zusätzlichen Bedingungen die Existenz des Mittelwerts und der Varianz garantiert und es ermöglicht, diese auf eine elegante Art und Weise direkt zu berechnen.

Acknowledgments

This is going to be a rather long acknowledgment, as I have to thank many people and still I could not include all.

First of all, I am truly grateful to my advisor Joachim Rosenthal. I want to thank Joachim for the usual things: the opportunity of doing a PhD under his supervision, for always being available, for the great atmosphere and all his support. But also for the unusual things, as Joachim has always believed in me, more than I did, for encouraging and pushing me, as I have grown a lot with it.

I would like to thank Jens Zumbrägel, Sihem Mesnager and Camilla Hollanti for their careful review of this thesis.

I am extremely grateful to my research group, consisting of such fabulous people as Alessandro, Elif, Gianira, Julia, Karan, Niklas and Simran. I want to thank my academic older brother Alessandro for great discussions and his valuable insights. I want to thank my pirate comrades, Gianira and Karan, for discussions considering research, but more importantly for the private ones. I want to thank them for reading this thesis and all their advice. Especially, I am very grateful to Karan, for the many projects we have mastered together, for the relentless support and his patience with such a stubborn academic twin sister.

I would like to thank my co-authors, Anna-Lena, Paolo, Massimo and Severin, for the wonderful collaboration, for the great ideas, for the patience and belief in my skills. Special thanks to Severin for always being helpful and for proof reading this thesis.

I would like to thank the whole math institute, going to work rather felt like coming home, I could not have wished for better colleagues. I want to thank Genta for being the best Genta ever, Andres for all his jokes no one ever gets, Nicola for always having great theories and Raul for philosophizing with me. I want to thank Simone, Céline, Simon, Alberto, Benedetta, Jacopo, Ödül, Giovanni, Giulio and Marco for the great atmosphere and their support. I also want to thank Franz and Claudia for the great work experience and I also wish to thank the secret rulers of the math institute, Grit and Carsten, for always helping me out.

I cannot begin to express my gratitude to my family, who have always supported and believed in me. I want to thank Jasmin and Ruedi, for being the most loving and caring parents, I have dedicated this thesis to them. I want to thank my sister Roxane, my brother Julian and my sister in law Julia for their endless support, and I want to thank my little nephews Sohel, Lino and Nereo for being the greatest joy in my life.

I would like to thank my dear friends, starting with my dearest, Saskia. I have to thank her for being my soul mate in all these years, for all the love and support, I could not have done this without her. I would also like to thank the whole Team Awesome, Annika, David, Francesco, Julia, Louis, Maaike, Nadine and Valeria for all these years of being awesome. Many thanks also go to the two sunshines Thomas and Krishna and to Mata Hari, for being always open and all the good times.

Finally, I would like to express my deepest gratitude to Giacomo, for all his patience, his expertise and advice, for great collaborations and helpful insights. But more importantly, I want to thank Giacomo for all his love and support. I could not have done any of this without him.

Thank you.

Contents

Chapter 1

Preface

This thesis covers two aspects of algebra: in the first part we focus on applied algebra, namely cryptography, whereas in the second part, we consider a more theoretical topic, namely number theory.

Although, number theory can be used in cryptography, the topics we cover in this thesis do not live in this intersection.

We have put the two very different aspects together in one thesis for several reasons; for applied algebra studying the theoretical background is crucial and leads to a broader insight. On the other hand, from applications, such as cryptography, arise many interesting theoretical questions. Thus, studying both, applications and the theoretical background, builds a fruitful bridge, especially for cryptography.

Information Set Decoding On the applied algebra side we focus on information set decoding. These are algorithms that decode random linear codes. A code is a linear space over a finite field and to decode is comparable with finding the element in the code, that is closest to a given element in the full space. In order to measure how close two vectors are codes are equipped with a metric. To ensure that decoding is possible a code is usually endowed with some algebraic structure. However, if a code is chosen at random this problem is NP-complete.

Although these algorithms arose from a theoretical question they are currently of crucial importance in code-based cryptography.

In the era of capable quantum computers all currently used public key cryptosystems will be completely broken. With the threat coming from quantum computers cryptography is facing its probably most important task: the search for quantum-secure cryptosystems.

At the core of post-quantum cryptography lies the hope that cryptosystems whose security rely on NP-complete problems will survive attacks on quantum computers.

With the seminal work of McEliece the NP-complete problem of decoding a random linear code lies at the base of code-based cryptography.

The National Institute of Standards and Technology (NIST), responsible for standardizing cryptosystems, has initiated such a standardization process for post-quantum

cryptosystems and code-based cryptography is one of the main candidates.

In the McEliece cryptosystem a code with algebraic structure is used as private key. The public key consists of a scrambled version of this code. If the code is hidden well enough an adversary has to decode a random looking code. Thus, the adversary would use the fastest algorithm to solve this problem, namely information set decoding.

Information set decoding is based on an algorithm proposed by Prange in 1962 and many improvements have been suggested since. Information set decoding algorithms come with a cost that is exponential in the input size. Thus, such algorithms do not break a cryptosystems but rather determine the size of the public key for a fixed security level.

One of the main problems in code-based cryptography is the enormous size of the public key. Since the proposal of McEliece many researchers have tried to tackle this problem by proposing different families of codes as secret codes. Only recently, a new approach in code-based cryptography has gained a lot of attention: the change of metric. Since classical coding theory deals with the Hamming metric classical code-based cryptography does as well. However, the code-based cryptosystems that provide the smallest key sizes within the NIST submission round 2 are based on the rank metric.

In fact, it appears that decoding in the rank metric is even more costly than in the Hamming metric. The role of information set decoding in code-based cryptography has thus changed: it does not only determine the key sizes anymore but it also predicts whether a metric could be used for cryptographic purposes. As a key rule it usually holds that the more errors a code can correct, the harder becomes the information set decoding and in turn the public key can be chosen of lower size.

This new approach has opened many new directions for code-based cryptography, as there are many more metrics that one can explore. In this thesis we present information set decoding algorithms endowed with the Lee metric. The Lee metric is usually considered over finite rings instead of finite fields. The advantage of the Lee metric is that it can correct many more errors than the Hamming metric. Thus, the Lee metric promises to lower the key sizes for a fixed security level.

Comparing then the input sizes, which are needed for information set decoding algorithms to achieve a fixed workfactor, we have the first hints that the Lee metric can hold this promise.

Natural Density On the theoretical side of algebra we focus, in the second part of this thesis, on computing densities. Since there is no uniform probability distribution over the integers, the notion of natural density is introduced. The main question can be formulated as follows: for a subset $T \subseteq \mathbb{Z}^d$, how likely is it for a randomly chosen a to lie in T? The natural density answers this question by first bounding with a height H, *i.e.*, how many a with $\mid a \mid \leq H$ lie in T? Secondly, it divides this quantity by the size of the d-dimensional cube of height H, and lastly it lets H go to infinity. If this limits exists, this gives the density of T.

The introduction of the natural density dates back to questions asked by Mertens and Césaro in the 1870's, namely: how likely is it that two randomly chosen integers are coprime? Or equivalently: what is the density of coprime pairs? Surprisingly, the

answer to this question is $\frac{6}{\pi^2}$.

To compute the natural density there exist various techniques. In this thesis we focus on the local to global principle that allows to compute the density by characterizing the target set T locally, *i.e.*, over the p-adic integers.

The generalization of the question of Mertens and Césaro is to compute the density of coprime m-tuples, which has been solved in the 1970's by Nymann. An even further generalization is to compute the density of rectangular unimodular matrices. This recent result has been achieved in collaboration with Giacomo Micheli. Another set of interest is the set of Eisenstein polynomials, *i.e.*, all polynomials that satisfy the criterion of Eisenstein. This result is due to Dubickas for the monic case and due to Heyman and Shparlinski in the non-monic case, both results are rather recently.

Even though the result for rectangular unimodular matrices implies the results for coprime m-tuples and coprime pairs, we give all results and proofs for the sake of completeness and for a didactic reason: to understand the local to global principle it makes sense to apply it first on easier examples and then to increase the difficulty. This part of the thesis is thus on one hand a compendium of density results through the local to global principle, but more importantly it can be considered a learning device for this technique. This is also the reason for the rather large preliminary study on the natural density, presented in the second part of this thesis.

Seeing the natural density as a $\mathbb{Z}$-analogue of a uniform probability distribution, the natural question that arises is to define the analogue of the mean and variance. Using tools from analytic number theory, we hence compute the mean and the variance for the four sets of interest, namely the coprime pairs, the coprime m-tuples, the Eisenstein polynomials and the rectangular unimodular matrices. These computations all follow a common strategy and moreover the results follow a certain pattern. This pattern becomes visible when considering the local characterization within the local to global principle. We thus provide an addendum to the local to global principle, where with a few additional conditions we can guarantee the existence of the mean and the variance of a target set and even more, using the local characterization, the mean and variance can be computed directly.

We then apply the addendum to the four sets of interest again to show not only that the computation through the addendum provides a much shorter and more elegant proof, but also confirming the previous computations through analytic tools.

Summary The results within this thesis can be summarized as follows.

In Chapter 5 we present three information set decoding algorithms in the Lee metric, namely the algorithms by Prange, Lee-Brickell and Stern, together with their complexity analyses.

In more detail, in Section 5.2 we consider the ambient space $\mathbb{Z}/4\mathbb{Z}$, which presents a special case for the Lee metric. This section is built up on the article [55].

Section 5.3 contains results from [104], where we generalize these results to $\mathbb{Z}/p^s\mathbb{Z}$, for a prime number p and a positive integer s. We lay a special focus on the case $s = 1$, where the underlying structure is a finite field, these results can be found in [105].

Furthermore, in Section 5.3.4 we consider a more general structure of the parity-check matrix. This section is built upon [105], but is carried out in more detail.

In Chapter 6 we compare the information set decoding algorithms in the Lee metric with their counterparts in the Hamming metric and conclude that the Lee metric presents a promising alternative for cryptographic purposes, as it decreases the key sizes substantially.

In Chapter 10 we compute the densities of coprime pairs, coprime m-tuples, Eisenstein polynomials and rectangular unimodular matrices over the integers via the local to global principle. We also provide the idea and the results for the densities of the target sets over the algebraic integers. This section contains the article [78].

Chapter 11 contains results not yet submitted, where we use tools from analytic number theory to compute the mean and variance corresponding to the natural density of the four considered sets.

With these results it becomes evident that they all follow a certain pattern. This allows us to give in Chapter 12 an addendum to the local to global principle which allows to compute the mean and variance directly. In addition, we revisit the results from Chapter 11, which follow now as a corollary from this addendum. This chapter contains results from [79].

Part I

Information Set Decoding

Chapter 2

Introduction

Coding Theory In classical coding theory, the ambient space is a finite field $\mathbb{F}_q$ which is endowed with a metric and the subject of interest is a linear subspace of $\mathbb{F}_q^n$, for some positive integer n. Such a linear subspace is called a code $\mathcal{C}$ and the positive integer n is called its length. The dimension k of $\mathcal{C}$ as a linear vector subspace of $\mathbb{F}_q^n$ is called the dimension of $\mathcal{C}$. The elements that are inside $\mathcal{C}$ are called codewords.

The main aim of coding theory is that of communication through a noisy channel, *i.e.*, a sender wants to send a message $\mathbf{m}$ through a channel that possibly adds a random noise $\mathbf{e}$ to the message. Thus, the receiver obtains $\mathbf{m} + \mathbf{e}$ and in order to recover the correct message $\mathbf{m}$, the receiver has to remove the error $\mathbf{e}$. To ensure that this is possible the message $\mathbf{m}$ is encoded, *i.e.*, one sends instead the codeword $\mathbf{c} \in \mathcal{C}$ corresponding to the original message $\mathbf{m}$. The metric we have endowed with the code is able to tell us how far away the received word $\mathbf{c} + \mathbf{e}$ is from the original codeword $\mathbf{c}$. If this discrepancy is below a certain threshold, called the error correction capacity, a decoding algorithm is able to recover $\mathbf{c}$ and thus also the corresponding message $\mathbf{m}$.

From this reasoning one can identify the important aspects of a code; we want to have a large error correction capacity and we want to have an efficient decoding algorithm. Note that the metric plays a crucial rule in this topic, as it determines the distance between two vectors. The classical distance for codes is hence the number of entries in which two vectors differ. This metric is called the Hamming metric.

A linear code $\mathcal{C}$ is usually represented through either the generator matrix, which has the code as image, or the parity-check matrix, which has the code as kernel.

Public Key Cryptography Although cryptography is a comparatively young branch of mathematics, due to its various applications it contains many topics, such as signature schemes, identification schemes, private information retrieval, and many more.

In this thesis we will focus on public key cryptography (PKC). In this setting we have two parties, usually denoted by A and B, or by Alice and Bob. In PKC Bob wants to send a confidential message to Alice, thus they require some encryption. In this scenario Alice, the constructor of the cryptosystem, has a pair of keys; a public key and a secret key (or private key). As the names suggest, she publishes the public

key and keeps the secret key private. Bob can use the public key of Alice to encrypt his message. The encrypted message, called a cipher, is then sent to Alice. Alice can then use the secret key to decrypt the cipher and recover the message. Thus a PKC system consists of three steps: the key generation, the encryption and the decryption.

An eavesdropper, usually denoted by E, or Eve, can only see the cipher and the public key. In order to have a secure cryptosystem, we want that the decryption of the cipher, and thus also obtaining the private key, is infeasible for Eve.

Note that the scenario is not restricted to two parties only; anyone can use the public key to send confidential messages to the constructor.

The two main factors in public key cryptography are the public key size, *i.e.*, how large is the public key, and the security level. The security level is the workfactor needed by an adversary to decrypt the cipher or to obtain the private key. Other factors include the efficiency of the decryption process, the secret key size and the cipher text size.

Clearly, the security level and the public key size are relative to each other. In the sense that the larger one chooses a public key, the harder it should become for an adversary to attack and vice versa. Nevertheless, what we want in a PKC is a small public key size and a large security level.

The currently most used public key cryptosystems are RSA [94] and ECC (for an overview see [80]). The former is based on the hardness of integer factorization and the latter on the discrete logarithm problem over elliptic curves. These systems have been used since decades, *e.g.*, RSA was proposed in 1977, and possess very small key sizes for a fixed security level. Nevertheless, these cryptosystems cannot be used in near future. In fact, in the era of capable quantum computers all currently used public key cryptosystems (based on the hardness of integer factorization or the discrete logarithm problem) will be completely broken by Shor's algorithm [97].

Due to recent advances in building capable quantum computers by Google, IBM and others (see for example [8, 32]), the cryptographic community has to come up with cryptosystems that will survive attacks on quantum computers. This completely new branch of cryptography is called post-quantum cryptography (PQC).

In 2016 the National Institute of Standards and Technology (NIST) has initiated a standardization process for post-quantum cryptosystems. It is believed that systems whose security relies on NP-complete problems will be quantum-secure [36].

The main candidates for post-quantum cryptography are:

- Code-based cryptography (CBC), based on the NP-complete problem of decoding a random linear code. For an overview see [88].
- Lattice-based cryptography, based on the NP-complete problems of finding the shortest vector, respectively the closest vector in a lattice. For an overview see [89].
- Multivariate cryptography, based on the NP-complete problem of solving mutlivariate quadratic equations. For an overview see [41].

- Isogeny-based cryptography, based on finding the isogeny map between two super singular elliptic curves [60].

For an overview on post-quantum cryptography see [23].

In this thesis we will only focus on the first candidate, *i.e.*, CBC.

Code-Based Cryptography Code-based cryptography first came up in 1978 with the seminal work of McEliece [74] and is one of the most promising candidates for post-quantum cryptography.

In a nutshell, the McEliece cryptosystem works as follows. The private key is given by the generator matrix $\mathbf{G}$ of a linear code $\mathcal{C}$ that has a large error correction capacity t and an efficient decoding algorithm $\mathcal{D}$. The public key is a disguised version of the generator matrix, *i.e.*, $\varphi(\mathbf{G})$, where φ is a bijective isometry. The message $\mathbf{m}$ is encrypted by encoding it through the disguised generator matrix and adding an intentional error vector of weight at most t, hence the cipher is given by $\mathbf{c} = \mathbf{m}\varphi(\mathbf{G})+\mathbf{e}$. The owner of the private key can then invert the disguising function and since $\varphi^{-1}(e)$ still has weight at most t the owner can use the decoding algorithm to recover the message m, *i.e.*,

$$\mathbf{m} = \varphi(\mathcal{D}(\varphi^{-1}(\mathbf{m}\varphi(\mathbf{G}) + \mathbf{e}))).$$

Originally, McEliece proposed to use a binary Goppa code as secret code and to disguise the generator matrix $\mathbf{G}$ by computing $\mathbf{SGP}$, where $\mathbf{S}$ is an invertible matrix, thus only changing the basis of the code and $\mathbf{P}$ is a permutation matrix, thus giving a permutation equivalent code and in particular keeping the weight of the error vector invariant. An equivalent [68] cryptosystem was proposed by Niederreiter in [84], where one uses the parity-check matrix $\mathbf{H}$ instead of the generator matrix and the cipher is given by the syndrome of an error vector, *i.e.*, $\mathbf{He}^\top$.

Niederreiter originally proposed to use generalized Reed-Solomon (GRS) codes as secret codes. However, Sidelnikov and Shestakov showed in [99] that using GRS codes in the Niederreiter system directly (and hence also in the McEliece cryptosystem) is not safe.

The McEliece cryptosystem, as well as Niederreiter's cryptosystem, is usually denoting the framework of this system, *i.e.*, without restricting to a particular code. The interested reader can find these frameworks in the appendix, Section A.1.

Note that McEliece's original proposal of using a binary Goppa code remains unbroken until today and is even the prioritized system for standardization by NIST [5].

The main problem of the original McEliece system however remains: it has very large public key sizes. More in detail, the submission of Bernstein *et al.* [26] to use the original McEliece system requires for the NIST category 1, asking for a security level of 128 bits, a public key size of approximately 2 Megabits, and for the NIST category 5, asking for a security level of 256 bits, it requires a public key size of approximately 8 Megabits. One of the main reasons why the original McEliece cryptosystem suffers from such large key sizes is the low error correction capacity of Goppa codes.

Since the proposal of McEliece in 1978, there have been many attempts to tackle this problem, mainly by proposing different families of codes as secret codes. Since it seems that using a code with larger error correction capacity might lower the public key sizes in the McEliece framework, many researchers have tried to use generalized Reed-Solomon codes [11, 12, 13, 18, 27, 62, 61, 84]. Note that generalized Reed-Solomon codes are MDS codes and thus they can correct the maximal number of errors for fixed code length and dimension. Although they did reduce the key sizes and seemed very promising, their error correction capacity is a result of their algebraic structure, which is very easily distinguishable from random codes [37, 38, 99, 106].

Apart from the generalized Reed-Solomon codes, some of the famous families of codes that were considered are: non-binary Goppa codes [24], algebraic geometric codes [59], LDPC and MDPC codes [10, 82], Reed-Muller codes [98] and convolutional codes [69]. Most of them were unsuccessful in hiding the structure of the private code [39, 40, 64, 81, 86].

As stated before, candidates for post-quantum cryptography are based on NP-complete problems. In the case of CBC, this problem is called syndrome decoding problem.

Syndrome Decoding Problem In 1978, Berlekamp, McEliece and van Tilborg proved that decoding a random binary linear code is NP-complete by reducing it from the 3-dimensional matching problem, [20]. In 1994, this was further generalized to arbitrary finite fields by Barg in [16]. The problem can be stated as follows.

Problem 1 (Decoding Problem). Let $k < n$ be positive integers. Given a $k \times n$ generator matrix $\mathbf{G}$ of a code $\mathcal{C}$ over $\mathbb{F}_q$, a positive integer t, and a vector $\mathbf{y} \in \mathbb{F}_q^n$, which is such that $\mathbf{y} = \mathbf{mG} + \mathbf{e}$, for some $\mathbf{m} \in \mathbb{F}_q^k$ and $\mathbf{e} \in \mathbb{F}_q^n$ of Hamming weight t, find $\mathbf{e}$.

Observe that this is equivalent to the syndrome decoding problem (SDP), in the same way as the Niederreiter system is equivalent to the McEliece system [68].

Problem 2 (Syndrome Decoding Problem). Let $k \leq n$ be positive integers. Given an $(n-k) \times n$ parity-check matrix $\mathbf{H}$ of a code $\mathcal{C}$ over $\mathbb{F}_q$, a positive integer t, and a vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$, which is such that $\mathbf{s} = \mathbf{eH}^\top$, for some $\mathbf{e} \in \mathbb{F}_q^n$ of Hamming weight t, find $\mathbf{e}$.

In the McEliece system, or equivalently in the Niederreiter system, the secret code is usually endowed with a particular algebraic structure to guarantee the existence of an efficient decoding algorithm. However, if this code is hidden well enough by the disguising function, an adversary has to solve the NP-complete problem of decoding a random linear code.

An adversary would hence use the best generic decoding algorithm for random linear codes. Until today two main methods for decoding random linear codes have been proposed: information set decoding (ISD) and the generalized birthday algorithm (GBA). ISD algorithms are more efficient if the decoding problem has only a small number of solutions, whereas GBA is efficient when there are many solutions. Also other ideas such as statistical decoding [4], gradient decoding [9] and supercode decoding [9] have

been proposed but fail to outperform ISD algorithms.

It is important to remark that the problem on which the McEliece system is based upon is not exactly equivalent to the SDP. In the McEliece system the parameter t is usually bounded by the error correction capacity of the chosen code. Whereas in the SDP, the parameter t can be chosen to be any positive integer. Thus, we are in a more restricted regime than in the SDP. There have been attempts [58] to transform the McEliece system in such a way that the underlying problem is closer or even exactly equivalent to the SDP, the actual NP-complete problem. This proposal has been attacked shortly after in [66].

Nevertheless, it is clear that the ISD algorithms solving the SDP also solve the restricted SDP, *i.e.*, the underlying problem of the McEliece system.

ISD algorithms are an important aspect of CBC, since they are needed to find the key size achieving a given security level. ISD algorithms hence do not break a code-based cryptosystem but they determine the choice of secure parameters.

Overview of Algorithms The duality of the generator matrix and the parity-check matrix also gives rise to two equivalent formulations of ISD algorithms, one being through the generator matrix and one through the parity-check matrix.

In this thesis we will only focus on the parity-check formulations of ISD algorithms, even though some of the ISD algorithms were originally proposed in the generator matrix formulation. Note that the formulation through the parity-check matrix is a priori more intuitive, since it is enough to find a vector of a certain weight which has the same syndrome as the corrupted codeword.

All ISD algorithms are based on a decoding algorithm proposed by Prange [93] in 1962, hence before any code-based cryptosystem was proposed and before it was shown that the underlying problem is NP-complete. The structures of the improvements do not change much from the original: as a first step one chooses an information set, then Gaussian elimination brings the parity-check matrix in a standard form and assuming that the error vector has a certain weight distribution, going through smaller parts of the error vector and checking for the parity-check equations to hold, will reveal the error vector.

In an ISD algorithm we hence fix a weight distribution of the error vector and then go through all information sets, checking if we find the information set for which the error vector indeed achieves this weight distribution. Hence ISD algorithms are in general not deterministic, since there are instances for which there exists no information set accomplishing the wanted weight distribution. Thus, ISD algorithms should not be confused with brute-force algorithms, which by default are deterministic.

The brute-force algorithm solving the SDP is doing the exact opposite: it first fixes an information set and then goes through all weight distributions of the error vector. This is clearly deterministic, since for any choice of an information set the wanted er-

ror vector has to achieve some weight distribution. Since the brute-force algorithm is clearly slower than ISD algorithms, in the sense that it needs more binary operations, in practice we only consider ISD algorithms.

In the original algorithm by Prange [93], sometimes also referred to as plain ISD, the weight distribution that we assume on the error vector is such that there are no errors in the information set and hence all t errors are outside the information set. Even though the cost of one iteration of Prange's original ISD algorithm is very low, the algorithm still has a huge complexity due to the large number of iterations needed. Indeed, the assumption that no errors happen in the information set is not very likely and thus the success probability of one iteration is very low.

Many improvements have been suggested to Prange's simplest form of ISD (see for example [29, 31, 34, 43, 63, 67, 103]). They all focus on a more elaborate and more likely weight distribution of the error vector, which results in a higher cost of one iteration but less iterations have to be performed.

The improvements were splitting from an early time on into two directions: the first direction is following the splitting of Lee and Brickell [65] into the information set and the redundant set, *i.e.*, they ask for v errors in the information set and $t-v$ outside. The second direction is Dumer's splitting approach [43], which is asking for v errors in $k+\ell$ bits, which are containing an information set, and $t-v$ in the remaining $n-k-\ell$ bits. Clearly, the second direction includes the first direction by setting $\ell = 0$. Apart from improvements regarding the success probability, one can also improve the cost of one iteration: Canteaut and Chabaud [30] have provided a speed up for finding information sets. They show that the information set should not be taken at random after one unsuccessful iteration, but rather a part of the previous information set should be reused and therefore a part of the Gaussian elimination step is already performed.

Now, we focus on the first direction of improvements, which were first proposed for codes over the binary field and then generalized to arbitrary finite fields $\mathbb{F}_q$. In 1988, the same year as Lee and Brickell proposed their algorithm, Leon [67] generalized Lee-Brickell's algorithm by introducing a set of size ℓ outside the information set called zero-window, where no errors happen. Also in 1988, Stern [100] adapted the algorithm by Leon and proposed to partition the information set into two sets and ask for v errors in each part and $t-2v$ errors outside the information set (and outside the zero-window). These three algorithms that have been proposed in the same year appear to be independent from each other, as no paper cites the other. The generalization of both Lee-Brickell and Stern's algorithm to a general finite field $\mathbb{F}_q$ were performed by Peters [90] in 2010.

In 2011, Bernstein, Lange and Peters proposed the ball-collision algorithm [25], where they keep the partitioning of the information set but they reintroduce errors in the zero-window. In fact, they partition the zero-window into two sets and ask for w errors in both and hence for $t-2v-2w$ errors outside. This algorithm and its speed-up techniques were then generalized to $\mathbb{F}_q$ by Interlando, Khathuria, Rohrer, Rosenthal and Weger in [57]. In 2016, Hirose [53] generalized the nearest neighbor algorithm over $\mathbb{F}_q$ and applied it to the generalized Stern algorithm.

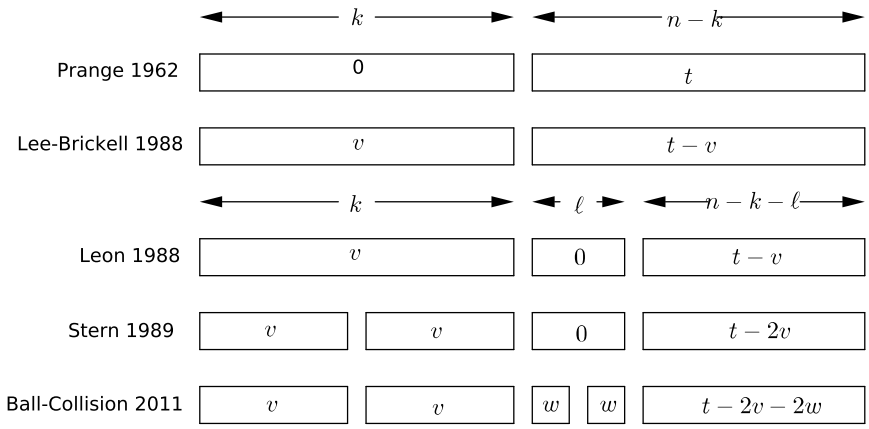

Figure 2.1: Overview of algorithms following the splitting of Lee-Brickell, adapted from [25].

An illustration for the algorithms that follow the improvements from the first direction is given in Figure 2.1. Note that for simplicity, we assume that the information set is in the first k positions and the zero-window is in the adjacent ℓ positions.

The second direction has resulted in many improvements, for example in 2009 Finiasz and Sendrier [45] have built two intersecting subsets of the $k+\ell$ bits, which contain an information set, and ask for v disjoint errors in both sets and $t-2v$ in the remaining $n-k-\ell$ bits. Niebuhr, Persichetti, Cayrel, Bulygin and Buchmann [83] in 2010 improved the performance of ISD algorithms over $\mathbb{F}_q$ based on the idea of Finiasz and Sendrier.

In 2011, May, Meurer and Thomae [72] proposed an improvement using the representation technique introduced by Howgrave-Graham and Joux [56]. To this algorithm Becker, Joux, May and Meurer [17] in 2012 introduced further improvements. This algorithm is usually referred to as BJMM algorithm.

In the same year Meurer in his dissertation [76] proposed a new generalized ISD algorithm based on these two papers. In 2015, May-Ozerov [71] used the nearest neighbor algorithm to improve the BJMM version of ISD. Later in 2017, the nearest neighbor algorithm over $\mathbb{F}_q$ was applied to the generalized BJMM algorithm by Gueye, Klamti and Hirose [51].

The second direction can be summarized as ISD algorithms, which do not use partitions but rather are allowing the sets to overlap. An illustration of the algorithms is given in Figure 2.2. Note that for simplicity we assume that the $k+\ell$ bits containing an information set are in the beginning. The overlapping sets are denoted by X_1 and X_2 and their intersection of size $2\alpha(k+\ell)$ is in blue. The amount of errors within the intersection is denoted by δ.

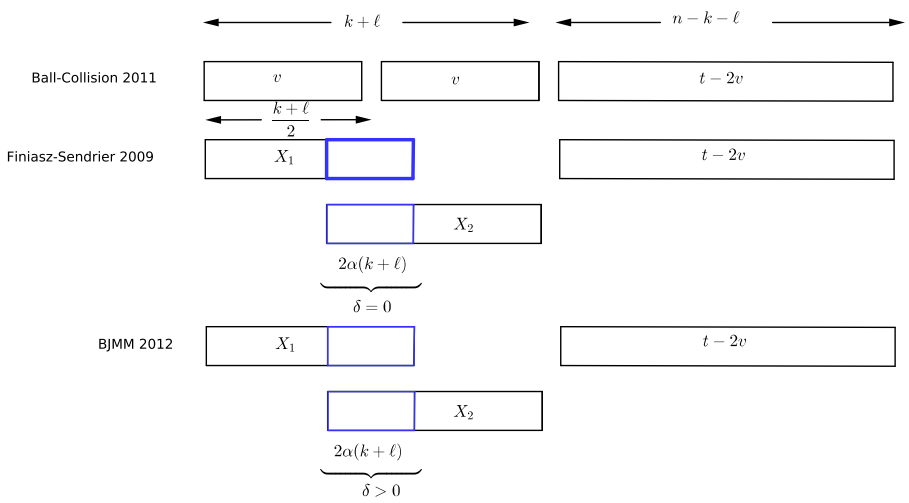

Figure 2.2: Overview of algorithms following the splitting of Dumer.

It is important to remark (see [76]) that the BJMM algorithm, even if having the smallest complexity, comes with a different cost: memory. In order to achieve a complexity of 128 bits, BJMM needs about 10^9 terabytes of memory. In fact, Meurer observed that if one restricts the memory to 2^{40} (which is a reasonable restriction), BJMM and the ball-collision algorithm are performing almost the same. We hence restrict ourselves to the first direction of improvements, where we use partitions.

In particular, we focus on Prange's original ISD algorithm, on Lee-Brickell's algorithm and on the most used ISD algorithm, which is Stern's algorithm. There are several reasons for this choice: on a classical computer one usually considers Stern's ISD algorithm. Although, there are some faster algorithms, *e.g.*, BJMM [17] and the ball collision algorithm [25], these cost improvements are only marginal, while the algorithm only gets more complex to understand.

On a capable quantum computer ISD algorithms are expected to have a square-root speed up [21], since Grover's search algorithm [49, 50] needs only $O(\sqrt{N})$ operations to find an element in a set of size N, instead of $O(N)$ many. Thus, intuitively, the search of an information set will become faster and thus the number of iterations needed in an ISD algorithm will decrease. All improvements on Prange's algorithm were focusing on a larger cost of one iteration, but in turn a smaller number of required iterations. Thus, on a capable quantum computer, it is believed that Prange's algorithm will be the fastest. Lee-Brickell's algorithm is only a slight modification of Prange's algorithm, nevertheless, due to this modification more instances can be solved.

Finally, we want to note here that Stern's algorithm already contains Lee-Brickell's algorithm. In fact, setting $\ell = 0$ and choosing the partition of the information set to be trivial, *i.e.*, one set is the full information set and the second is the empty set, Stern's

algorithm coincides with Lee-Brickell's algorithm.

In addition, Lee-Brickell's algorithm contains Prange's algorithm. One can see this by choosing in Lee-Brickell's algorithm $v = 0$.

Thus, it would be enough to provide only Stern's algorithm. Nevertheless, since the improvements are more complex to understand, we still provide all three algorithms separately, for the sake of clearness and completeness.

Change of Metric From the short historic overview above it becomes clear, that CBC constantly tried to find a balance between security and key sizes. This trade-off was assumed to be an unsolvable problem for CBC and its downfall, as other post-quantum candidates provide lower key sizes, for example lattice-based cryptosystems [54]. This changed only recently by the introduction of new metrics to CBC. In fact, in the NIST submission round 2 the two CBC systems achieving the lowest key sizes are endowed with the rank metric (see [2, 3]), instead of the Hamming metric as it is the case for classical coding theory.

The considered metric builds the fundament of coding theory. Thus, changing this metric changes also everything else. The reason for the low public key sizes in the rank metric is twofold: on one hand it seems that decoding a random linear rank metric code appears to be more difficult. In fact, while in the Hamming metric an ISD algorithm would go through vectors of a certain support size, in the rank metric it would go through subspaces of certain dimensions. Thus, changing the cost from the Newton binomial to the Gaussian binomial.

On the other hand, rank metric ISD algorithms (see [7, 35, 47, 87]) are very recent and have not been studied as thoroughly as in the Hamming metric.

This most likely also gives the reason why rank metric cryptosystems did not survive the third submission round of the NIST standardization process: recently an attack via the MinRank problem [15] was proposed on rank metric cryptosystems. This did not only increase the public key sizes for the cryptosystems but also showed that the metric and its application in cryptography is still very young and requires more thorough studies.

Although the rank metric cryptosystems have not been chosen as finalists in the standardization process, they have opened a new perspective in CBC that seems very fruitful. Namely, to study different kinds of metrics and their effects, when applied in CBC, on security levels and key sizes.

Following this idea, we want to introduce the Lee metric to cryptography. For this the study of ISD algorithms is crucial. In fact, if decoding a random linear code in certain metric comes with a high cost, then the considered metric presents a promising candidate for CBC and vice versa: improving ISD algorithms forces existing code-based cryptosystems to increase their key sizes accordingly.

2.1 Organization of this Part

The first part of the thesis is organized as follows. In Section 2.3 we introduce the notation that we use throughout this part and in Chapter 3 we cover the required preliminaries. More in detail, in Section 3.1 we recall the basic definitions of coding theory, in Section 3.2 we give a general structure of ISD algorithms and in Section 3.3 we introduce the techniques that we use for the ISD algorithms.

In Chapter 4 we introduce the three main algorithms that we will cover in this part of the thesis, namely Prange's, Lee-Brickell's and Stern's ISD algorithm. Although they have been originally introduced over the binary, we give them directly over $\mathbb{F}_q$ endowed with the Hamming metric and analyze their costs.

In Chapter 5 we change our focus to the Lee metric. Thus, in Section 5.1 we recall some definitions of ring-linear coding theory and properties of the Lee metric with a special focus on the quaternary case. In Section 5.1.1 we translate the techniques introduced in Section 3.3 to the Lee metric, whenever possible. In Section 5.2 we translate the ISD algorithms of interest, *i.e.*, Prange's, Lee-Brickell's and Stern's algorithm, to $\mathbb{Z}/4\mathbb{Z}$ endowed with the Lee metric.

In Section 5.3 we then generalize the ISD algorithms of interest to $\mathbb{Z}/p^s\mathbb{Z}$, where we also cover the special case of prime fields, *i.e.*, $s = 1$. Note that the structure of the algorithms over $\mathbb{Z}/p^s\mathbb{Z}$ is provided for a simplified systematic form. Thus, in Section 5.3.4 we introduce a technique to generalize these algorithms even further.

In Chapter 6 we compare the three ISD algorithms in the different ambient spaces and metrics that we have considered.

In Chapter 7 we draw some concluding remarks and state some open problems.

2.2 New Results in this Part

In this part of the thesis we will focus on three ISD algorithms, namely

1. Prange's ISD algorithm,
2. Lee-Brickell's ISD algorithm,
3. Stern's ISD algorithm.

The reason for this selection is the following: Prange's algorithm is the first ISD algorithm that has been proposed. It is also considered to be the fastest algorithm on a capable quantum computer [21]. On the other hand, Lee-Brickell's algorithm is a slight adaption of Prange's algorithm, though with the effect that more instances can be solved. Whereas Stern's algorithm is the ISD algorithm that is usually considered on a classical computer. Although there are faster algorithms, the improvements in the workfactor are only marginal and Stern's algorithm is not overly complex.

Note that the original algorithms by Prange [93], Lee-Brickell [65] and Stern [100] are proposed over the binary finite field endowed with the Hamming metric. In Chapter 4 we give the algorithms over $\mathbb{F}_q$. The generalization of Lee-Brickell's and Stern's algorithm to $\mathbb{F}_q$ was done by Peters in [90]. We also recall their algorithms over the binary in the appendix, Section A.2.

In Section 5.2 we translate the three algorithms, namely by Prange, Lee-Brickell and Stern to $\mathbb{Z}/4\mathbb{Z}$ endowed with the Lee metric. This section is based on the paper [55]:

Information set decoding in the Lee metric with applications to cryptography, by Anna-Lena Horlemann-Trautmann and Violetta Weger, in Advances in Mathematics of Communications, 10.3934/amc.2020089, 2019.

More in detail, Lee-Brickell's and Stern's algorithm over $\mathbb{Z}/4\mathbb{Z}$ endowed with the Lee metric were provided in [55] in collaboration with Anna-Lena Horlemann-Trautmann. Thus, the Lee metric analogue of Prange's algorithm over $\mathbb{Z}/4\mathbb{Z}$ is only available in this thesis.

In Section 5.3 we generalize the three ISD algorithms to $\mathbb{Z}/p^s\mathbb{Z}$ endowed with the Lee metric. This section is based on the papers [104, 105]:

Information set decoding of Lee-metric codes over finite rings, by Violetta Weger, Massimo Battaglioni, Paolo Santini, Franco Chiaraluce, Marco Baldi and Edoardo Persichetti, arXiv preprint arXiv:2001.08425, 2020.

On the Hardness of the Lee Syndrome Decoding Problem, by Violetta Weger, Massimo Battaglioni, Paolo Santini, Anna-Lena Horlemann-Trautmann and Edoardo Persichetti, arXiv preprint arXiv:2002.12785, 2020.

In more detail, Prange's algorithm and Stern's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric were provided in [104] in collaboration with Massimo Battaglioni, Paolo Santini, Franco Chiaraluce, Marco Baldi and Edoardo Persichetti.

Prange's algorithm and Stern's algorithm over $\mathbb{F}_p$ endowed with the Lee metric, as well as Lee-Brickell's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ and over $\mathbb{F}_p$ endowed with the Lee metric were provided in [105] in collaboration with Massimo Battaglioni, Paolo Santini, Anna-Lena Horlemann-Trautmann and Edoardo Persichetti.

The idea of the possible generalization was provided in [105], however without a complexity analysis or an example.

2.3 Notation

Let p be a prime number, q a prime power. We denote by $\mathbb{Z}/q\mathbb{Z}$ the ring of integers modulo q, and by $\mathbb{F}_q$ the finite field with q elements, as usual. For a ring $\mathcal{R}$, the multiplicative group is denoted by $\mathcal{R}^\times$.

Given an integer x, we denote its absolute value as $\mid x \mid$. We use capital letters to denote sets of integers. The cardinality of a set is denoted as $\mid V \mid$.

We use bold lower case, respectively upper case letters to denote (row-) vectors, respectively matrices. The identity matrix of size k is denoted by Id_k; the $k \times n$ zero

matrix is denoted as $\mathbf{0}_{k\times n}$, while $\mathbf{0}_n$ simply denotes the zero vector of length n. For a matrix $\mathbf{M}$, we denote by $\mathbf{M}^\top$ its transpose, by $\det(\mathbf{M})$ its determinant and by $\mathrm{rk}(\mathbf{M})$ its rank. For a set S, we denote by S^C its complement.

Given a vector $\mathbf{x}$ of length n and a set $S \subseteq \{1,\dots,n\}$, we denote by $\mathbf{x}_S$ the projection of $\mathbf{x}$ to the coordinates indexed by S. In the same way, $\mathbf{M}_S$ denotes the projection of the $k \times n$ matrix $\mathbf{M}$ to the columns indexed by S.

The support of a vector $\mathbf{a}$ of length n is defined as

$$\mathrm{Supp}(\mathbf{a}) := \{i \in \{1,\dots,n\} \mid a_i \neq 0\}.$$

For $S \subseteq \{1,\dots,n\}$ of size k, we denote by $\mathbb{F}_q^n(S)$ the vectors in $\mathbb{F}_q^n$ having support inside S. The projection of $\mathbf{x} \in \mathbb{F}_q^n(S)$ to $\mathbb{F}_q^k$ is then canonical and denoted by $\pi_S(\mathbf{x})$. On the other hand, we denote by $\sigma_S(\mathbf{x})$ the canonical embedding of a vector $\mathbf{x} \in \mathbb{F}_q^k$ into $\mathbb{F}_q^n(S)$.

Lastly, for a function f we denote by $\mathrm{im}(f)$ its image and by $\ker(f)$ its kernel. By a slight abuse of notation, we use the same also for matrices, *i.e.*, if $\mathbf{M} \in \mathbb{F}_q^{n\times m}$, then $\mathrm{im}(\mathbf{M}) = \{\mathbf{aM} \mid \mathbf{a} \in \mathbb{F}_q^n\}$, whereas $\ker(\mathbf{M}) = \{\mathbf{b} \in \mathbb{F}_q^m \mid \mathbf{bM}^\top = \mathbf{0}\}$.

Chapter 3

Preliminaries

3.1 Coding Theory

With the rationale given in Chapter 2 in mind, let us introduce coding theory properly.

Throughout the whole first part of this thesis, we will denote by $\mathbb{F}_q$ a finite field with q elements, where q is a prime power.

Definition 3.1.1 (Linear Code)**.** Let $1 \leq k \leq n$ be integers. Then, an $[n, k]$ *linear code* $\mathcal{C}$ over $\mathbb{F}_q$ is a k-dimensional linear subspace of $\mathbb{F}_q^n$.

As mentioned before, the parameter n is called the *length* of the code, and the elements in the code are called *codewords*. Classically, $\mathbb{F}_q$ is endowed with the Hamming metric.

Definition 3.1.2 (Hamming Distance)**.** Let n be a positive integer. For $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$, the *Hamming distance* between $\mathbf{x}$ and $\mathbf{y}$ is given by the number of positions in which they differ, *i.e.*,

$$d_H(\mathbf{x}, \mathbf{y}) = \mid \{i \in \{1, \ldots, n\} \mid x_i \neq y_i\} \mid .$$

Definition 3.1.3 (Hamming Weight)**.** Let n be a positive integer. For $\mathbf{x} \in \mathbb{F}_q^n$, the *Hamming weight* of $\mathbf{x}$ is given by the size of its support, *i.e.*,

$$\mathrm{wt}_H(\mathbf{x}) = \mid \{i \in \{1, \ldots, n\} \mid x_i \neq 0\} \mid .$$

Obviously, $\mathrm{wt}_H(\mathbf{x}) = d_H(\mathbf{x}, \mathbf{0})$ and $d_H(\mathbf{x}, \mathbf{y}) = \mathrm{wt}_H(\mathbf{x} - \mathbf{y})$. Thus, the Hamming distance is induced by the Hamming weight.

Another important parameter of a code is its minimum distance, *i.e.*, the minimal distance achieved by its distinct codewords.

Definition 3.1.4 (Minimum Distance)**.** Let $\mathcal{C}$ be a linear code over $\mathbb{F}_q$. The *minimum Hamming distance* of $\mathcal{C}$ is denoted by $d_H(\mathcal{C})$ and given by

$$d_H(\mathcal{C}) = \min\{d_H(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in \mathcal{C},\ \mathbf{x} \neq \mathbf{y}\}.$$

If there is no ambiguity, we will denote the minimum Hamming distance by d_H. Clearly, for a linear code $\mathcal{C}$ we have that

$$d_H(\mathcal{C}) = \min\{\mathrm{wt}_H(\mathbf{c}) \mid \mathbf{c} \in \mathcal{C},\ \mathbf{c} \neq \mathbf{0}\}.$$

In general, for an $\mathbf{x} \in \mathbb{F}_q^n$, we write $d_H(\mathbf{x}, \mathcal{C})$ for the minimal distance between $\mathbf{x}$ and a codeword in $\mathcal{C}$, *i.e.*,

$$d_H(\mathbf{x}, \mathcal{C}) = \min\{d_H(\mathbf{x}, \mathbf{c}) \mid \mathbf{c} \in \mathcal{C}\}.$$

Let us define the ball of radius r around a codeword $c \in \mathcal{C}$.

Definition 3.1.5. Let $k \leq n$ and r be positive integers, let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$, and let $\mathbf{c} \in \mathcal{C}$. Then the *Hamming ball* of radius r around $\mathbf{c}$ is defined as

$$B_H(\mathbf{c}, r, n, q) = \{\mathbf{x} \in \mathbb{F}_q^n \mid d_H(\mathbf{x}, \mathbf{c}) \leq r\}.$$

Note that since the code $\mathcal{C}$ is linear, the size of $B_H(\mathbf{c}, r, n, q)$ does not depend on the choice of $\mathbf{c}$. Thus, let us introduce

$$V_H(r, n, q) = \{\mathbf{x} \in \mathbb{F}_q^n \mid \mathrm{wt}_H(\mathbf{x}) \leq r\}.$$

The size of $V_H(r, n, q) = B_H(\mathbf{0}, r, n, q)$, the ball of radius r around the zero codeword in the Hamming distance, can be computed very easily:

$$\mid V_H(r, n, q) \mid = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i.$$

In fact, $\mathbf{x} \in V_H(r, n, q)$ is equivalent to $\mathrm{wt}_H(\mathbf{x}) \leq r$. Hence we go through all $i \leq r$, and compute the number of vectors in $\mathbb{F}_q^n$ of Hamming weight i, *i.e.*, we fix i positions, and for each position we have $q-1$ choices for entries in $\mathbb{F}_q^\times$.

The minimum distance of a code plays a crucial role in coding theory, as it directly implies how many errors can be corrected. For this let us elaborate on the concept of error correction. We say that a code can *correct* up to t errors, if for all $\mathbf{x} \in \mathbb{F}_q^n$ with $d_H(\mathbf{x}, \mathcal{C}) \leq t$, there exists exactly one $\mathbf{y} \in \mathcal{C}$, such that $d_H(\mathbf{x}, \mathbf{y}) \leq t$. Further, a *decoding algorithm* $\mathcal{D}$ is defined such that, for any $\mathbf{x} \in \mathbb{F}_q^n$, with $d_H(\mathbf{x}, \mathcal{C}) \leq t$, $\mathcal{D}(\mathbf{x})$ outputs $\mathbf{y} \in \mathcal{C}$ with $d_H(\mathbf{x}, \mathbf{y}) = d_H(\mathbf{x}, \mathcal{C})$, *i.e.*, the nearest codeword to $\mathbf{x}$.

Proposition 3.1.6 (Error Correction Capacity)**.** *Let $\mathcal{C}$ be a linear code over $\mathbb{F}_q$ of length n and of minimum distance d_H. Then, the code can correct up to $\left\lfloor \frac{d_H - 1}{2} \right\rfloor$ errors.*

For this choose r maximal such that all Hamming balls of radius r around the codewords of $\mathcal{C}$ do not intersect, *i.e.*, for all $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ with $\mathbf{x} \neq \mathbf{y}$ we have that $B_H(\mathbf{x}, r, n, q) \cap B_H(\mathbf{y}, r, n, q) = \emptyset$. Since the minimum distance between two distinct codewords is given by d_H, we must have that $r \leq \left\lfloor \frac{d_H - 1}{2} \right\rfloor$. In fact, let $\mathbf{x}, \mathbf{z} \in \mathcal{C}$ be two distinct codewords, and assume that we received a vector $\mathbf{y} \in \mathbb{F}_q^n$, with $\mathbf{y} \in B_H(\mathbf{x}, r, n, q) \cap B_H(\mathbf{z}, r, n, q)$. This implies that

$$d_H(\mathbf{x}, \mathbf{z}) \leq d_H(\mathbf{x}, \mathbf{y}) + d_H(\mathbf{y}, \mathbf{z}) \leq 2r \leq d_H - 1,$$

which contradicts that the minimum distance between two distinct codewords is d_H.

Thus, the error correction capacity is clearly the maximal radius such that all balls around the codewords do not intersect.

The Singleton bound provides an upper bound on the minimum distance of a code, by just using the length n and the dimension k.

Theorem 3.1.7 (Singleton Bound)**.** *Let $k \leq n$ be positive integers and let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$. Then,*

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

The proof is quite easy, since one can observe that by deleting $d_H - 1$ of the positions in all codewords, we have a code of length $n - d_H + 1$ and since the minimum distance between two distinct codewords of $\mathcal{C}$ was given by d_H, in the new smaller code we must still have all distinct codewords. Thus, the dimension of the new code remains k.

A code that achieves the Singleton bound is called a *maximum distance separable* (MDS) code. Such codes of course are of huge interest, since for fixed n and k they can correct the maximal amount of errors.

The representation of linear codes is usually done via two matrices, which we now introduce. On one hand, we have a matrix $\mathbf{G}$, which has the code $\mathcal{C}$ as image.

Definition 3.1.8 (Generator Matrix)**.** Let $k \leq n$ be positive integers and let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$. Then, a matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ which has the code as image, *i.e.*, such that

$$\mathcal{C} = \{\mathbf{x}\mathbf{G} \mid \mathbf{x} \in \mathbb{F}_q^k\},$$

is called a *generator matrix* of $\mathcal{C}$.

On the other hand, there exists a matrix $\mathbf{H}$, which has the code as kernel.

Definition 3.1.9 (Parity-Check Matrix)**.** Let $k \leq n$ be positive integers and let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$. Then, a matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ which has the code as kernel, *i.e.*, such that

$$\mathcal{C} = \{\mathbf{y} \in \mathbb{F}_q^n \mid \mathbf{H}\mathbf{y}^\top = \mathbf{0}\},$$

is called a *parity-check matrix* of $\mathcal{C}$.

Hence, we can look at the short exact sequence given by

$$\mathbf{0} \to \mathbb{F}_q^k \xrightarrow{\mathbf{G}} \mathbb{F}_q^n \xrightarrow{\mathbf{H}^\top} \mathbb{F}_q^{n-k} \to \mathbf{0},$$

with $\mathrm{im}(\mathbf{G}) = \mathcal{C} = \ker(\mathbf{H}^\top)$. For any $\mathbf{x} \in \mathbb{F}_q^n$, we call $\mathbf{x}\mathbf{H}^\top$ a *syndrome*. Clearly, if the syndrome corresponding to $\mathbf{x}$ is zero, then $\mathbf{x} \in \mathcal{C}$, and vice versa.

For $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$ let us denote by $\langle \mathbf{x}, \mathbf{y} \rangle$ the standard inner product, *i.e.*,

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i.$$

Then, we can define the dual of an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ as the dual subspace.

Definition 3.1.10 (Dual Code)**.** Let $k \leq n$ be positive integers and let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$. The *dual code* $\mathcal{C}^\perp$ is an $[n, n-k]$ linear code over $\mathbb{F}_q$, defined as

$$\mathcal{C}^\perp = \{\mathbf{x} \in \mathbb{F}_q^n \mid \langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{0} \ \forall \ \mathbf{y} \in \mathcal{C}\}.$$

Note that a parity-check matrix of $\mathcal{C}$ is in fact a generator matrix of $\mathcal{C}^\perp$.

Let us shortly recall the notation introduced in Section 2.3. For $\mathbf{x} \in \mathbb{F}_q^n$ and $S \subset \{1, \ldots, n\}$ we denote by $\mathbf{x}_S$ the vector consisting of the entries of $\mathbf{x}$ indexed by S. Similarly, we denote by $\mathcal{C}_S$ the code consisting of the codewords $\mathbf{c}_S$.

The bijective map induced by $\mathbf{G}$ is usually called *encoding map* $\mathcal{E}$:

$$\begin{aligned} \mathcal{E} : \mathbb{F}_q^k &\to \mathcal{C}, \\ \mathbf{x} &\mapsto \mathbf{x}\mathbf{G}. \end{aligned}$$

As we can see from the encoding map, the code is already completely defined by k positions, more formally, we want to introduce the notion of information set.

Definition 3.1.11 (Information Set)**.** Let $k \leq n$ be positive integers and let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$. Then, a set $I \subset \{1, \ldots, n\}$ of size k, with

$$| \mathcal{C} | = | \mathcal{C}_I |$$

is called an *information set* of $\mathcal{C}$.

Note that an $[n, k]$ linear code can have at most $\binom{n}{k}$ many information sets.

As a corollary, we have that $\mathbf{G}_I$ is an invertible matrix of size k and $\mathbf{H}_{I^C}$ is an invertible matrix of size $n - k$. In particular, we want to introduce the notion of systematic form.

Definition 3.1.12 (Systematic Form)**.** Let $k \leq n$ be positive integers and $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$. Then, for some permutation matrix $\mathbf{P}$ and some invertible matrix $\mathbf{U}$ the *systematic form* of the generator matrix $\mathbf{G}$ is

$$\mathbf{UGP} = \begin{pmatrix} \mathrm{Id}_k & \mathbf{A} \end{pmatrix},$$

where $\mathbf{A} \in \mathbb{F}_q^{k \times (n-k)}$. Whereas, for some permutation matrix $\mathbf{P}'$ and some invertible matrix $\mathbf{U}'$, the systematic form of the parity-check matrix is

$$\mathbf{U}'\mathbf{H}\mathbf{P}' = \begin{pmatrix} \mathbf{B} & \mathrm{Id}_{n-k} \end{pmatrix},$$

where $\mathbf{B} \in \mathbb{F}_q^{(n-k) \times k}$.

Since $\mathbf{G}\mathbf{H}^\top = \mathbf{0}$, we have that the matrices $\mathbf{A}$ and $\mathbf{B}$ are related. In fact, it holds that $\mathbf{B} = -\mathbf{A}^\top$.

As information sets are of particular interest in this part of the thesis, we provide here some examples.

Let $\mathcal{C}$ be the code generated by $\mathbf{G} \in \mathbb{F}_5^{2 \times 4}$, which is such that

$$\mathbf{G} = \begin{pmatrix} 1 & 3 & 2 & 3 \\ 0 & 4 & 4 & 3 \end{pmatrix}.$$

This is an extremal example, as all subsets of $\{1,\dots,4\}$ of size 2 are information sets. The systematic form corresponding to the information set $I = \{1,2\}$ is of the form

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

To give an example of a non-information set, let us look at the code $\mathcal{C}'$ generated by $\mathbf{G}' \in \mathbb{F}_5^{2\times 4}$, which is given by

$$\mathbf{G}' = \begin{pmatrix} 1 & 2 & 4 & 3 \\ 2 & 3 & 0 & 0 \end{pmatrix}.$$

Then, the set $\{3,4\}$ is clearly not an information set.

Let us recall here the syndrome decoding problem, since the main goal of this part of the thesis is to solve this problem via information set decoding algorithms.

Problem 3 (Syndrome Decoding Problem). Let $k \leq n$ be positive integers. Given an $(n-k)\times n$ parity-check matrix $\mathbf{H}$ of a code $\mathcal{C}$ over $\mathbb{F}_q$, a positive integer t, and a vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$, which is such that $\mathbf{s} = \mathbf{e}\mathbf{H}^\top$, for some $\mathbf{e} \in \mathbb{F}_q^n$ of Hamming weight t, find $\mathbf{e}$.

Note that a possible brute-force algorithm would be to go through all vectors $\mathbf{x} \in \mathbb{F}_q^n$ of Hamming weight t and to check whether $\mathbf{x}\mathbf{H}^\top = \mathbf{s}$ is satisfied. The cost of such an algorithm is given by

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

many additions and multiplications. However, as we will see in the next chapter, using information set decoding algorithms this cost can be greatly reduced.

Lastly, we want to state here the Gilbert-Varshamov bound in the Hamming metric which provides a sufficient conditions for the existence of linear codes.

Theorem 3.1.13 (Gilbert-Varshamov bound [95], Theorem 4.4)**.** *Let q be a prime power and let $k \leq n$ and d_H be positive integers, such that*

$$V_H(d_H - 2, n-1, q) < q^{n-k}. \tag{3.1.1}$$

Then, there exists a $[n,k]$ linear code over $\mathbb{F}_q$ with minimum Hamming distance at least d_H.

Observe that the classical Gilbert-Varshamov bound is a lower bound on the maximal cardinality of any code, not necessarily a linear code. However, the bound we are interested in is an upper bound, since we want the existence of a linear code.

3.2 Structure of Information Set Decoding

Information set decoding (ISD) algorithms, which first came up in 1962 by the seminal work of Prange [93], solve a difficult problem, namely the syndrome decoding problem, which in fact is proven to be NP-complete [20]. Thus, ISD algorithms are clearly not polynomial-time algorithms but rather exponential in the parameters of the instance,

i.e., in the code length n, the code dimension k and the target weight t.

As we have seen in the Chapter 2 the structure of the improvements on Prange's ISD algorithm do not differ much from the original. We hence give here a general structure of ISD algorithms. We are given a parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ of a code $\mathcal{C}$, a positive integer t and a syndrome $\mathbf{s} \in \mathbb{F}_q^{n-k}$, such that there exists a vector $\mathbf{e} \in \mathbb{F}_q^n$ of Hamming weight less than or equal to t with syndrome $\mathbf{s}$, *i.e.*, $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$. The aim of the algorithm is to find such a vector $\mathbf{e}$.

1. Find an information set $I \subset \{1, \dots, n\}$ of size k for $\mathcal{C}$.
2. Bring $\mathbf{H}$ into the systematic form corresponding to I, *i.e.*, find an invertible matrix $\mathbf{U} \in \mathbb{F}_q^{(n-k)\times(n-k)}$, such that $(\mathbf{U}\mathbf{H})_I = \mathbf{A}$, for some $\mathbf{A} \in \mathbb{F}_q^{(n-k)\times k}$ and $(\mathbf{U}\mathbf{H})_{I^C} = \mathrm{Id}_{n-k}$.
3. Go through all error vectors $\mathbf{e} \in \mathbb{F}_q^n$ having the assumed weight distribution (and in particular having Hamming weight t).
4. Check if the parity-check equations, *i.e.*, $\mathbf{e}\mathbf{H}^\top\mathbf{U}^\top = \mathbf{s}\mathbf{U}^\top$ are satisfied.
5. If they are verified, output $\mathbf{e}$, if not start over with a new selection of I.

The cost of an ISD algorithm is given by the cost of one iteration times how many times on average one has to repeat an iteration.

The average number of iterations required is given as the reciprocal of the success probability of one iteration. This probability is completely defined by the weight distribution we have assumed on the error vector.

3.3 Techniques

In this section we introduce the techniques that we will use throughout this chapter. Whenever it makes sense, we will introduce them directly over $\mathbb{F}_q$ equipped with the Hamming metric, else we will first introduce them for the Hamming metric over $\mathbb{F}_2$, where they were originally proposed, and then generalize them to $\mathbb{F}_q$. Some of these techniques can be used also in the Lee metric, for this we refer to Section 5.1.1. Most of these techniques were introduced in [25] over $\mathbb{F}_2$ and later generalized to $\mathbb{F}_q$ in [57]. We want to note here that the original algorithms considered in Chapter 4 did not apply these concepts to speed up their cost. Hence the cost we provide for the algorithms can be considered revised with the newest techniques.

First of all, we want to fix the cost that we consider throughout this thesis of one addition and one multiplication over $\mathbb{F}_q$, *i.e.*, we assume that one addition over $\mathbb{F}_q$ costs $\lceil \log_2(q) \rceil$ binary operations and one multiplication costs $\lceil \log_2(q) \rceil^2$ binary operations. The cost of the multiplication is clearly not using the fastest algorithm known but will be good enough for our purposes. Also for the cost of multiplying two matrices we will always stick to a broad estimate given by school book long multiplication, *i.e.*, multiplying $\mathbf{A}\mathbf{B}$, where $\mathbf{A} \in \mathbb{F}_q^{k\times n}$ and $\mathbf{B} \in \mathbb{F}_q^{n\times r}$ will cost $nkr\left(\lceil \log_2(q) \rceil + \lceil \log_2(q) \rceil^2\right)$ binary operations. However, over the binary the cost of a multiplication is the same as the cost of an addition, both only costing 1 bit.

Number of Iterations One of the main steps in computing the cost of an information set decoding algorithm was already explained in Section 3.2, namely the average number of iterations needed. This number depends on the success probability of one iteration. In turn, the success probability is completely given by the weight distribution of the error vector that we assume in the corresponding algorithm. In more detail, in one iteration we consider a fixed information set and thus the success probability of an iteration is given by the fraction of how many vectors there are with the assumed weight distribution, divided by how many vectors there are in general with this weight.

For example, we are looking for $\mathbf{e} \in \mathbb{F}_q^n$ of Hamming weight t, and we assume that the error vector has no errors inside an information set I, and thus all t errors appear in I^C of size $n-k$. Since there are $\binom{n-k}{t}(q-1)^t$ many vectors having support of size t in a set of size $n-k$ and the total number of vectors of support t in a set of size n is given by $\binom{n}{t}(q-1)^t$, we have that the success probability of one iteration is given by

$$\binom{n-k}{t}\binom{n}{t}^{-1},$$

and hence the number of iterations needed on average is given by

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

We can see clearly that the success probability, and thus also the number of iterations, does not depend on whether we are over $\mathbb{F}_2$ or $\mathbb{F}_q$. This will however change, when looking at a different metric than the Hamming metric.

Early Abort This technique that provides a speed up to some algorithms was introduced in [25] over $\mathbb{F}_2$ and then further generalized to $\mathbb{F}_q$ in [57].

In some of the algorithms we have to perform a computation and the algorithm will only proceed if the result of this computation satisfies a certain property. In our case, the property is that the weight of the resulting vector does not exceed a target weight.

We will thus compute one entry of the result and check the weight of this entry, before proceeding with the next entry. As soon as the weight of the full resulting vector would already be above the target weight, we can stop the computations, hence aborting early.

To provide an example also for this technique, assume that we have to compute $\mathbf{xA}$, for $\mathbf{x} \in \mathbb{F}_q^k$ of Hamming weight t and $\mathbf{A} \in \mathbb{F}_q^{k \times n}$. Usually computing $\mathbf{xA}$ would cost $nt\left(\lceil \log_2(q) \rceil^2 + \lceil \log_2(q) \rceil\right)$ binary operations.

However, assuming our algorithm only proceeds if $\mathrm{wt}_H(\mathbf{xA}) = w$, we can use the method of early abort, *i.e.*, computing one entry of the resulting vector and checking its weight simultaneously. For this we assume that the resulting vector is uniformly distributed. Since we are over $\mathbb{F}_q$, the probability that an entry adds to the weight of the full vector is given by $\frac{q-1}{q}$. Hence we can expect that after computing $\frac{q}{q-1}w$ entries the resulting vector should have reached weight w, and after computing $\frac{q}{q-1}(w+1)$ entries we should have exceeded the target weight w and can abort. Since computing

only one entry of the resulting vector costs $t\left(\lceil\log_2(q)\rceil^2+\lceil\log_2(q)\rceil\right)$ binary operations, the cost of this step is hence given by

$$\frac{q}{q-1}(w+1)t\left(\lceil\log_2(q)\rceil^2+\lceil\log_2(q)\rceil\right)$$

binary operations, instead of the previous

$$nt\left(\lceil\log_2(q)\rceil^2+\lceil\log_2(q)\rceil\right).$$

Clearly, this is a speed up, whenever $\frac{q}{q-1}(w+1)<n$.

Number of Collisions This concept was originally introduced in [25] for the ball collision algorithm over $\mathbb{F}_2$ and then generalized to $\mathbb{F}_q$ in [57]. In some algorithms we want to check if a certain condition is verified and only then we would proceed. This condition depends on two vectors $\mathbf{x}$ and $\mathbf{y}$ living in some sets. Hence the algorithm would go through all the vectors $\mathbf{x}$ and then through all the vectors $\mathbf{y}$ in their respective sets and check if the condition is satisfied for a fixed pair $(\mathbf{x},\mathbf{y})$. If this is the case, such a pair is called a *collision*. For all subsequent steps of the algorithm one would need to perform them for all the collisions, thus multiplying the cost of these steps with the size of the set of all $(\mathbf{x},\mathbf{y})$.

Instead, we can compute the average number of collisions we can expect. Let us also give an example for this technique; assume that we only proceed whenever

$$\mathbf{x}+\mathbf{y}=\mathbf{s},$$

for a fixed $\mathbf{s}\in\mathbb{F}_q^k$ and for all $\mathbf{x}\in\mathbb{F}_q^k$ of Hamming weight v and all $\mathbf{y}\in\mathbb{F}_q^k$ of Hamming weight w. To verify this condition we have to go through all possible $\mathbf{x}$ and $\mathbf{y}$, thus costing

$$\binom{k}{v}\binom{k}{w}(q-1)^{v+w}\min\{k,v+w\}\log_2(q)$$

binary operations. As a subsequent step one would compute for all such $(\mathbf{x},\mathbf{y})$ the vector $\mathbf{Ax}-\mathbf{By}$, for some fixed $\mathbf{A}\in\mathbb{F}_q^{n\times k}$ and $\mathbf{B}\in\mathbb{F}_q^{n\times k}$. Usually one would do this for all elements in $S=\{(\mathbf{x},\mathbf{y})\mid\mathbf{x},\mathbf{y}\in\mathbb{F}_q^k,\mathrm{wt}_H(\mathbf{x})=v,\mathrm{wt}_H(\mathbf{y})=w\}$, giving this step a cost of

$$\binom{k}{v}\binom{k}{w}(q-1)^{v+w}\min\{k,v+w\}n\left(\log_2(q)+\log_2(q)^2\right).$$

With the concept of average number of collisions we only have to perform the subsequent steps as many times as on average we expect a collision, *i.e.*, a pair $(\mathbf{x},\mathbf{y})$ such that $\mathbf{x}+\mathbf{y}=\mathbf{s}$. Assuming a uniform distribution, this amount is given by

$$\frac{\mid S\mid}{q^n}=\frac{\binom{k}{v}\binom{k}{w}(q-1)^{v+w}}{q^n}<\binom{k}{v}\binom{k}{w}(q-1)^{v+w-n}.$$

Thus computing $\mathbf{Ax}-\mathbf{By}$ for all $(\mathbf{x},\mathbf{y})\in S$ costs on average

$$\binom{k}{v}\binom{k}{w}(q-1)^{v+w-n}\min\{k,v+w\}n\left(\log_2(q)+\log_2(q)^2\right)$$

binary operations, which is clearly less than the previous cost.

Intermediate Sums This concept, like the previous, was introduced as well in [25] over $\mathbb{F}_2$ and generalized to $\mathbb{F}_q$ in [57]. We will first give the original idea over $\mathbb{F}_2$ and then show its generalization to $\mathbb{F}_q$.

In some algorithms we have to do a certain computation for all vectors in a certain set. The idea of intermediate sums is to do this computation in the easiest case and to use the resulting vector to compute the results for harder cases. This will become clear with an example.

Let $\mathbf{A} \in \mathbb{F}_2^{k \times n}$ and assume that we want to compute $\mathbf{xA}$ for all $\mathbf{x} \in \mathbb{F}_2^k$ of Hamming weight t. This would usually cost nt binary operations, for each $\mathbf{x}$. If we have to compute this for all $\mathbf{x} \in \mathbb{F}_2^k$ of Hamming weight t, then we would end up with a cost of

$$nt\binom{k}{t}$$

binary operations.

However, using the concept of intermediate sums helps to speed up this computation: we first compute $\mathbf{xA}$ for all $\mathbf{x} \in \mathbb{F}_2^k$ of Hamming weight 1, thus just outputting the rows of $\mathbf{A}$ which comes with no cost. As a next step, we compute $\mathbf{xA}$ for all $\mathbf{x} \in \mathbb{F}_2^k$ of Hamming weight 2, which is the same as adding two rows of $\mathbf{A}$, thus coming with a cost of $\binom{k}{2}n$ binary operations. As a next step, we compute $\mathbf{xA}$ for all $\mathbf{x} \in \mathbb{F}_2^k$ of Hamming weight 3, which is the same as adding one row of $\mathbf{A}$ to one of the already computed vectors from the previous step, thus this costs $\binom{k}{3}n$ binary operations. If we proceed in this way, until we compute $\mathbf{xA}$ for all $\mathbf{x} \in \mathbb{F}_2^k$ of Hamming weight t, this step costs

$$nL(k,t)$$

binary operations, where we define

$$L(k,t) = \sum_{i=2}^{t} \binom{k}{i}.$$

This is a speed up to the previous cost, since

$$n\sum_{i=2}^{t} \binom{k}{i} = n\left(\binom{k}{2} + \cdots + \binom{k}{t}\right) < nt\binom{k}{t}.$$

When generalizing this result to $\mathbb{F}_q$, to compute $\mathbf{xA}$ for all $\mathbf{x} \in \mathbb{F}_q^k$ of Hamming weight 1, does not come for free anymore but is rather given by computing $\mathbf{A} \cdot \lambda$ for all $\lambda \in \mathbb{F}_q^\times$, which costs $kn \lceil \log_2(q) \rceil^2$ binary operations. Further, if we want to compute $\mathbf{xA}$ for all $\mathbf{x} \in \mathbb{F}_q^k$ of Hamming weight 2, we have to add two multiples of rows of $\mathbf{A}$. While there are still $\binom{k}{2}$ many rows, we now have $(q-1)^2$ multiples. Thus, this step costs $\binom{k}{2}(q-1)^2 n \lceil \log_2(q) \rceil$ binary operations. Proceeding in this way, the cost of computing $\mathbf{xA}$ for all $\mathbf{x} \in \mathbb{F}_q^k$ of Hamming weight t, is given by

$$L_q(k,t) n \lceil \log_2(q) \rceil + kn \lceil \log_2(q) \rceil^2$$

binary operations, where we define

$$L_q(k,t) = \sum_{i=2}^{t} \binom{k}{i} (q-1)^i.$$

Instead of the previous cost of

$$\binom{k}{t}(q-1)^t nt \left(\lceil \log_2(q) \rceil^2 + \lceil \log_2(q) \rceil \right)$$

binary operations.

Chapter 4

Information Set Decoding in the Hamming Metric

In this chapter we provide the algorithms of Prange, Lee-Brickell and Stern over $\mathbb{F}_q$ endowed with the Hamming metric, we also emphasize the special case of $\mathbb{F}_2$, where the algorithms were originally proposed.

4.1 Prange's algorithm

Prange's algorithm [93] was introduced in 1962 and marks the first ISD algorithm over the binary. Sometimes it is referred to as plain ISD algorithm. Although there have been many improvements proposed on this plain ISD, Prange's algorithm should still be considered today. Indeed, on one hand, these improvements only give slight speed ups, and on the other hand, Prange's algorithm might be the fastest one on a capable quantum computer [22].

In Prange's algorithm we assume that there exists an information set I that is disjoint to the support of the error vector $\mathrm{Supp}(\mathbf{e})$, *i.e.*,

$$I \cap \mathrm{Supp}(\mathbf{e}) = \emptyset.$$

Of course, such an assumption comes with a probability whose reciprocal defines how many iterations are needed on average, if the algorithm ends. Observe that Prange's algorithm is indeed not deterministic, *i.e.*, there are instances which Prange's algorithm can not solve. For an easy example, one can just take an instance where $\mathrm{wt}_H(\mathbf{e}) = t > n - k = \mid I^C \mid$. For a more elaborate example, which also allows unique decoding, assume that we have a parity-check matrix, which is such that each information set includes the first position. Then an error vector with non-zero entry in the first position could never be found through Prange's algorithm.

More in detail, the structure of this algorithm is as follows: we first find an information set, then bring the parity-check matrix into systematic form according to this information set and when we apply the same row operations on the syndrome and its

weight matches the target weight, the modified syndrome is indeed the wanted error vector.

To illustrate the algorithm, let us assume that the information set is $I = \{1, \dots, k\}$, and let us denote by $J = I^C$. To bring the parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ into systematic form, we multiply by an invertible matrix $\mathbf{U} \in \mathbb{F}_q^{(n-k)\times(n-k)}$. Since we assume that no errors occur in the information set, we have that $\mathbf{e} = (\mathbf{0}_k, \mathbf{e}_J)$ with $\mathrm{wt}_H(\mathbf{e}_J) = t$. We are in the following situation:

$$\mathbf{e}\mathbf{H}^\top\mathbf{U}^\top = \begin{pmatrix}\mathbf{0}_k & \mathbf{e}_J\end{pmatrix}\begin{pmatrix}\mathbf{A}^\top \\ \mathrm{Id}_{n-k}\end{pmatrix} = \mathbf{s}\mathbf{U}^\top,$$

for $\mathbf{A} \in \mathbb{F}_q^{(n-k)\times k}$.

It follows that $\mathbf{e}_J = \mathbf{s}\mathbf{U}^\top$ and hence we are only left with checking the weight of $\mathbf{s}\mathbf{U}^\top$.

We will now give the algorithm of Prange in its full generality, *i.e.*, we are not restricting to the choice of I and J that we made before for illustrating the algorithm, thus we make use of the notation, such as $\mathbf{H}_I$, introduced in Section 2.3.

Algorithm 1 Prange's Algorithm over $\mathbb{F}_q$ in the Hamming metric

Input: $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $t \in \mathbb{N}$.
Output: $\mathbf{e} \in \mathbb{F}_q^n$ with $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$ and $\mathrm{wt}_H(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size k and define $J = I^C$.
2: Compute $\mathbf{U} \in \mathbb{F}_q^{(n-k)\times(n-k)}$, such that

$$(\mathbf{U}\mathbf{H})_I = \mathbf{A} \text{ and } (\mathbf{U}\mathbf{H})_J = \mathrm{Id}_{n-k},$$

where $\mathbf{A} \in \mathbb{F}_q^{(n-k)\times k}$.
3: Compute $\mathbf{s}' = \mathbf{s}\mathbf{U}^\top$.
4: **if** $\mathrm{wt}_H(\mathbf{s}') = t$ **then**
5: Return $\mathbf{e}$ such that $\mathbf{e}_I = \mathbf{0}_k$ and $\mathbf{e}_J = \mathbf{s}'$.
6: Start over with Step 1 and a new selection of I.

We now provide a complexity estimate of Prange's algorithm over $\mathbb{F}_q$ in the Hamming metric.

Theorem 4.1.1. *Prange's algorithm over $\mathbb{F}_q$ requires on average*

$$\binom{n-k}{t}^{-1}\binom{n}{t}(n-k)^2(n+1)(\lceil\log_2(q)\rceil + \lceil\log_2(q)\rceil^2)$$

binary operations.

Proof. One iteration of Algorithm 1 over $\mathbb{F}_q$ only consists in bringing $\mathbf{H}$ into systematic form and to apply the same row operations on the syndrome; thus, the cost can be

assumed equal to that of computing $\mathbf{U}\begin{pmatrix}\mathbf{H} & \mathbf{s}^\top\end{pmatrix}$, *i.e.*,

$$(n-k)^2(n+1)(\lceil \log_2(q) \rceil + \lceil \log_2(q) \rceil^2)$$

binary operations.

The success probability is given by having chosen the correct weight distribution of $\mathbf{e}$.

In this case, we require that no errors happen in the chosen information set, hence such a probability is given by

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

Then, the estimated overall cost of Prange's ISD algorithm over $\mathbb{F}_q$ is given as in the claim. □

Corollary 4.1.2. *Prange's algorithm over the binary endowed with the Hamming metric costs*

$$\binom{n-k}{t}^{-1}\binom{n}{t}(n-k)^2(n+1)$$

binary operations.

For the algorithm over $\mathbb{F}_2$ and its complexity analysis we refer the interested reader to the appendix, *i.e.*, Section A.2.

4.2 Lee-Brickell's algorithm

Lee-Brickell's algorithm [65] is an improvement to Prange's algorithm by reducing the number of iterations needed and increasing the cost of one iteration. This algorithm was originally proposed over the binary and then generalized to $\mathbb{F}_q$ by Peters in [90].

Different to Prange's algorithm, where no errors were allowed inside the information set, in Lee-Brickell's algorithm, we assume that there exists an information set, such that v errors happen inside the information set and $t-v$ outside. Similar to Prange's algorithm also the algorithm by Lee-Brickell is non-deterministic, as the parameter v is fixed as an input of the algorithm.

We give Lee-Brickell's algorithm over $\mathbb{F}_q$ in Algorithm 2. But first we elaborate on the structure of the algorithm. We first find an information set, then bring the parity-check matrix into systematic form according to this information set and go through all vectors of length k and Hamming weight v. We can now define the remaining error vector of length $n-k$, and check if it is has the remaining Hamming weight $t-v$.

As before, we will fix a choice of information set to illustrate the algorithm. Let us assume that the information set is $I = \{1, \ldots, k\}$, and hence $J = \{k+1, \ldots, n\}$. By multiplying with some $\mathbf{U}$ we can bring $\mathbf{H}$ into systematic form. Let us write the error vector as $\mathbf{e} = (\mathbf{e}_I, \mathbf{e}_J)$, with $\mathrm{wt}_H(\mathbf{e}_I) = v$ and $\mathrm{wt}_H(\mathbf{e}_J) = t - v$. We hence get that

$$\mathbf{e}\mathbf{H}^\top\mathbf{U}^\top = \begin{pmatrix}\mathbf{e}_I & \mathbf{e}_J\end{pmatrix}\begin{pmatrix}\mathbf{A}^\top \\ \mathrm{Id}_{n-k}\end{pmatrix} = \mathbf{s}\mathbf{U}^\top,$$

where $\mathbf{A} \in \mathbb{F}_q^{(n-k)\times k}$.

If we go through all $\mathbf{e}_I \in \mathbb{F}_q^k$ of Hamming weight v, it follows that $\mathbf{e}_J = \mathbf{s}\mathbf{U}^\top - \mathbf{e}_I\mathbf{A}^\top$. Hence we are only left with checking the weight of $\mathbf{e}_J$.

We will now give the algorithm of Lee-Brickell in its full generality, *i.e.*, without restricting to the choice of I and J.

Algorithm 2 Lee-Brickell's Algorithm over $\mathbb{F}_q$ in the Hamming metric

Input: $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $t \in \mathbb{N}$ and $v < \min\{t, k\}$.
Output: $\mathbf{e} \in \mathbb{F}_q^n$ with $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$ and $\mathrm{wt}_H(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size k and define $J = I^C$.
2: Compute $\mathbf{U} \in \mathbb{F}_q^{(n-k)\times(n-k)}$, such that

$$(\mathbf{U}\mathbf{H})_I = \mathbf{A} \quad \text{and } (\mathbf{U}\mathbf{H})_J = \mathrm{Id}_{n-k},$$

where $\mathbf{A} \in \mathbb{F}_q^{(n-k)\times k}$.
3: Compute $\mathbf{s}' = \mathbf{s}\mathbf{U}^\top$.
4: **for** $\mathbf{e}_I \in \mathbb{F}_q^k$ with $\mathrm{wt}_H(\mathbf{e}_I) = v$ **do**
5: **if** $\mathrm{wt}_H(\mathbf{s}' - \mathbf{e}_I\mathbf{A}^\top) = t - v$ **then**
6: Return $\mathbf{e}$ such that $\mathbf{e}_I = \mathbf{e}_I$ and $\mathbf{e}_J = \mathbf{s}' - \mathbf{e}_I A^\top$.
7: Start over with Step 1 and a new selection of I.

The following theorem provides a complexity estimate of Algorithm 2.

Theorem 4.2.1. *Lee-Brickell's algorithm over $\mathbb{F}_q$ requires on average*

$$\binom{k}{v}^{-1}\binom{n-k}{t-v}^{-1}\binom{n}{t}\left((n-k)^2(n+1) + \binom{k}{v}(q-1)^v \cdot \min\left\{n-k, \frac{q}{q-1}(t-v+1)\right\}v\right)\left(\lceil\log_2(q)\rceil + \lceil\log_2(q)\rceil^2\right)$$

binary operations.

Proof. Let us first consider the cost of one iteration. As a first step, we bring $\mathbf{H}$ into systematic form and to apply the same row operations on the syndrome. This can again be assumed to be equal to that of computing $\mathbf{U}\begin{pmatrix}\mathbf{H} & \mathbf{s}^\top\end{pmatrix}$, hence a broad estimate for the cost is

$$(n-k)^2(n+1)\left(\lceil\log_2(q)\rceil + \lceil\log_2(q)\rceil^2\right)$$

binary operations.

Then, we go through all $\mathbf{e}_I \in \mathbb{F}_q^k$ of Hamming weight v, and compute $\mathbf{s}' - \mathbf{e}_I\mathbf{A}^\top$. This would usually require

$$\binom{k}{v}(q-1)^v(n-k)v\left(\lceil\log_2(q)\rceil + \lceil\log_2(q)\rceil^2\right)$$

binary operations, since $\mathbf{e}_I$ has support of size v. However, the algorithm only proceeds if the Hamming weight of $\mathbf{s}' - \mathbf{e}_I\mathbf{A}^\top$ is $t - v$. Hence we can use the concept of early abort. Assuming that the resulting vector is uniformly distributed, we have that one entry of the resulting vector adds Hamming weight 1 to the weight of the full vector, with probability $\frac{q-1}{q}$. Thus, we have to compute on average $\frac{q}{q-1}(t - v + 1)$ many entries of the resulting vector before we can abort. Computing one entry of the vector $\mathbf{s}' - \mathbf{e}_I\mathbf{A}^\top$ costs

$$v\left(\lceil \log_2(q) \rceil + \lceil \log_2(q) \rceil^2\right)$$

binary operations. Therefore, by applying early abort, we get that this step costs on average

$$\binom{k}{v}(q-1)^v \frac{q}{q-1}(t-v+1)v\left(\lceil \log_2(q) \rceil + \lceil \log_2(q) \rceil^2\right)$$

binary operations.

Finally, we compute the average number of iterations needed. In this case, we require that v errors happen in the chosen information set and the remaining $t - v$ outside, hence the success probability is given by

$$\binom{k}{v}\binom{n-k}{t-v}\binom{n}{t}^{-1}.$$

Thus, by multiplying the cost of one iteration with the reciprocal of the success probability, we get the claim. □

The binary case is very similar to the q-ary one, hence we will only give here the cost. For the algorithm over the binary and the complexity analysis we refer to the appendix, *i.e.*, Section A.2.

Corollary 4.2.2. *The cost of Lee-Brickell's algorithm over the binary is given by*

$$\binom{k}{v}^{-1}\binom{n-k}{t-v}^{-1}\binom{n}{t}\left((n-k)^2(n+1) + \binom{k}{v}\min\{n-k, 2(t-v+1)\}v\right)$$

binary operations.

4.3 Stern's algorithm

Stern's algorithm [100] is one of the most used ISD algorithms, as it is considered one of the fastest algorithms on a classical computer. Stern's algorithm is an improvement on Lee-Brickell's and Prange's algorithm that decreases the number of iterations needed on average and in turn has a higher cost of one iteration. Also Stern's algorithm was generalized to $\mathbb{F}_q$ by Peters in [90].

In Stern's algorithm we use the idea of Lee-Brickell and allow errors inside the information set and in addition we partition the information set into two sets and ask for v errors in both of them. Further, we also use the idea of Leon [67] to have a *zero-window* of size ℓ outside the information set, where no errors happen.

Similar to Lee-Brickell's and Prange's algorithm also Stern's algorithm is non-deterministic.

We will give Stern's algorithm in Algorithm 3. But first we explain the algorithm and illustrate it for an easy choice of the information set and the zero-window.

We first find an information set, then bring the parity-check matrix into systematic form according to this information set. We partition the information set into two sets and define the sets S and T, where S takes care of all vectors living in one partition and T takes care of all vectors living in the other partition. We can now check whether two of such fixed vectors give us the wanted error vector.

To illustrate the algorithm, we assume that the information set is $I = \{1, \dots, k\}$ and that the zero-window is $Z = \{k+1, \dots, k+\ell\}$. Further, let us define $J = (I \cup Z)^C = \{k+\ell+1, \dots, n\}$. We again denote by $\mathbf{U}$ the matrix that brings the parity-check matrix into systematic form and write the error vector partitioned into the information set part I, the zero-window part Z and the remaining part J, as $\mathbf{e} = (\mathbf{e}_I, \mathbf{0}_\ell, \mathbf{e}_J)$, with $\mathrm{wt}_H(\mathbf{e}_I) = 2v$ and $\mathrm{wt}_H(\mathbf{e}_J) = t - 2v$. Thus, we get the following:

$$\mathbf{e}\mathbf{H}^\top\mathbf{U}^\top = \begin{pmatrix}\mathbf{e}_I & \mathbf{0}_\ell & \mathbf{e}_J\end{pmatrix}\begin{pmatrix}\mathbf{A}^\top & \mathbf{B}^\top \\ \mathrm{Id}_\ell & \mathbf{0}_{\ell\times(n-k-\ell)} \\ \mathbf{0}_{(n-k-\ell)\times\ell} & \mathrm{Id}_{n-k-\ell}\end{pmatrix} = \begin{pmatrix}\mathbf{s}_1 & \mathbf{s}_2\end{pmatrix} = \mathbf{s}\mathbf{U}^\top,$$

where $\mathbf{A} \in \mathbb{F}_q^{\ell\times k}$ and $\mathbf{B} \in \mathbb{F}_q^{(n-k-\ell)\times k}$.

From this we get the following two conditions

$$\mathbf{e}_I\mathbf{A}^\top = \mathbf{s}_1, \tag{4.3.1}$$

$$\mathbf{e}_I\mathbf{B}^\top + \mathbf{e}_J = \mathbf{s}_2. \tag{4.3.2}$$

We partition the information set I into the sets X and Y, for the sake of clarity, assume that k is even and $m = k/2$. Assume that $X = \{1, \dots, m\}$ and $Y = \{m+1, \dots, k\}$. Hence, we can write $\mathbf{e}_I = (\mathbf{e}_X, \mathbf{e}_Y)$, and Condition (4.3.1) becomes

$$\sigma_X(\mathbf{e}_X)\mathbf{A}^\top = \mathbf{s}_1 - \sigma_Y(\mathbf{e}_Y)\mathbf{A}^\top. \tag{4.3.3}$$

Observe that the σ_X is needed, as $\mathbf{e}_X$ has length m but we want to multiply it to $\mathbf{A}^\top \in \mathbb{F}_q^{k\times\ell}$. In the algorithm we will not use the embedding σ_X but rather $\mathbb{F}_q^k(X)$, thus $\mathbf{e}_X$ will have length k, but only support in X.

In the algorithm, we will define a set S that contains all vectors of the form $\sigma_X(\mathbf{e}_X)\mathbf{A}^\top$, *i.e.*, of the left side of (4.3.3) and a set T that contains all vectors of the form $\mathbf{s}_1 - \sigma_Y(\mathbf{e}_Y)\mathbf{A}^\top$, *i.e.*, of the right side of (4.3.3). Whenever a vector in S and a vector in T coincide, we call such a pair a collision.

For each collision we define $\mathbf{e}_J$ such that Condition (4.3.2) is satisfied, *i.e.*,

$$\mathbf{e}_J = \mathbf{s}_2 - \mathbf{e}_I\mathbf{B}^\top$$

and if the weight of $\mathbf{e}_J$ is the remaining $t - 2v$, we have found the wanted error vector.

Algorithm 3 Stern's Algorithm over $\mathbb{F}_q$ in the Hamming metric

Input: $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $t \in \mathbb{N}$, $k = m_1 + m_2, \ell < n-k$ and $v < \min\{m_1, m_2, \lfloor \frac{t}{2} \rfloor\}$.
Output: $\mathbf{e} \in \mathbb{F}_q^n$ with $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$ and $\mathrm{wt}_H(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size k and choose a zero-winow $Z \subset I^C$ of size ℓ, and define $J = (I \cup Z)^C$.
2: Partition I into X of size m_1 and Y of size $m_2 = k - m_1$.
3: Compute $\mathbf{U} \in \mathbb{F}_q^{(n-k)\times(n-k)}$, such that

$$(\mathbf{UH})_I = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix}, \ (\mathbf{UH})_Z = \begin{pmatrix} \mathrm{Id}_\ell \\ \mathbf{0}_{(n-k-\ell)\times\ell} \end{pmatrix} \text{ and } (\mathbf{UH})_J = \begin{pmatrix} \mathbf{0}_{\ell\times(n-k-\ell)} \\ \mathrm{Id}_{n-k-\ell} \end{pmatrix},$$

where $\mathbf{A} \in \mathbb{F}_q^{\ell\times k}$ and $\mathbf{B} \in \mathbb{F}_q^{(n-k-\ell)\times k}$.
4: Compute $\mathbf{s}\mathbf{U}^\top = \begin{pmatrix} \mathbf{s}_1 & \mathbf{s}_2 \end{pmatrix}$, where $\mathbf{s}_1 \in \mathbb{F}_q^\ell$ and $\mathbf{s}_2 \in \mathbb{F}_q^{n-k-\ell}$.
5: Compute the set S

$$S = \{(\mathbf{e}_X\mathbf{A}^\top, \mathbf{e}_X) \mid \mathbf{e}_X \in \mathbb{F}_q^k(X), \mathrm{wt}_H(\mathbf{e}_X) = v\}.$$

6: Compute the set T

$$T = \{(\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top, \mathbf{e}_Y) \mid \mathbf{e}_Y \in \mathbb{F}_q^k(Y), \mathrm{wt}_H(\mathbf{e}_Y) = v\}.$$

7: **for** $(\mathbf{a}, \mathbf{e}_X) \in S$ **do**
8: **for** $(\mathbf{a}, \mathbf{e}_Y) \in T$ **do**
9: **if** $\mathrm{wt}_H(\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top) = t - 2v$ **then**
10: Return $\mathbf{e}$ such that $\mathbf{e}_I = \mathbf{e}_X + \mathbf{e}_Y$, $\mathbf{e}_Z = \mathbf{0}_\ell$ and $\mathbf{e}_J = \mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$.
11: Start over with Step 1 and a new selection of I.

We will now give the algorithm of Stern in its full generality, *i.e.*, we are not restricting to the choice of I, J and Z, that we made before for illustrating the algorithm.

Algorithm 3 has a complexity estimate given as follows.

Theorem 4.3.1. *Stern's algorithm over $\mathbb{F}_q$ requires on average*

$$\begin{aligned}
&\binom{m_1}{v}^{-1}\binom{m_2}{v}^{-1}\binom{n-k-\ell}{t-2v}^{-1}\binom{n}{t}\\
&\cdot\left((n-k)^2(n+1)\left(\lceil\log_2(q)\rceil+\lceil\log_2(q)\rceil^2\right)+(m_1+m_2)\ell\lceil\log_2(q)\rceil^2\right.\\
&+\ell\left(L_q(m_1,v)+L_q(m_2,v)+\binom{m_2}{v}(q-1)^v\right)\lceil\log_2(q)\rceil\\
&+\frac{\binom{m_1}{v}\binom{m_2}{v}(q-1)^{2v}}{q^\ell}\min\left\{n-k-\ell,\frac{q}{q-1}(t-2v+1)\right\}\\
&\left.\cdot\,2v\left(\lceil\log_2(q)\rceil^2+\lceil\log_2(q)\rceil\right)\right)
\end{aligned}$$

binary operations.

Proof. Similar to the previous algorithms, as a first step we bring $\mathbf{H}$ into systematic form and apply the same row operations on the syndrome; a broad estimate for the cost is given by

$$(n-k)^2(n+1)\left(\lceil\log_2(q)\rceil+\lceil\log_2(q)\rceil^2\right)$$

binary operations.

To compute the set S, we can use the technique of intermediate sums. We want to compute $\mathbf{e}_X\mathbf{A}^\top$ for all $\mathbf{e}_X\in\mathbb{F}_q^k(X)$ of Hamming weight v. Using intermediate sums, this costs

$$L_q(m_1,v)\ell\lceil\log_2(q)\rceil+m_1\ell\lceil\log_2(q)\rceil^2$$

binary operations.

Similarly, we can build set T: we want to compute $\mathbf{s}_1-\mathbf{e}_Y\mathbf{A}^\top$, for all $\mathbf{e}_Y\in\mathbb{F}_q^k(Y)$ of Hamming weight v. Using intermediate sums, this costs

$$L_q(m_2,v)\ell\lceil\log_2(q)\rceil+m_2\ell\lceil\log_2(q)\rceil^2+\binom{m_2}{v}(q-1)^v\ell\lceil\log_2(q)\rceil$$

binary operations. Note, that the $L_q(m_2,v)\ell\lceil\log_2(q)\rceil+m_2\ell\lceil\log_2(q)\rceil^2$ part comes from computing $\mathbf{e}_Y\mathbf{A}^\top$, whereas the $\binom{m_2}{v}(q-1)^v\ell\lceil\log_2(q)\rceil$ part comes from subtracting from each of the vectors $\mathbf{e}_Y\mathbf{A}^\top$ the vector $\mathbf{s}_1$.

In the remaining steps of the algorithms we go through all $(\mathbf{a},\mathbf{e}_X)\in S$ and all $(\mathbf{a},\mathbf{e}_Y)\in T$, thus usually the cost of these steps should be multiplied by the size of $S\times T$. However, since the algorithm first checks for a collision, we can use instead of $\mid S\mid\mid T\mid$ the number of collisions we expect on average. More precisely: since S consists of all $\mathbf{e}_X\in\mathbb{F}_q^k(X)$ of Hamming weight v, S is of size $\binom{m_1}{v}(q-1)^v$ and similarly T is of size $\binom{m_2}{v}(q-1)^v$. The resulting vectors $\mathbf{e}_X\mathbf{A}^\top$, respectively, $\mathbf{s}_1-\mathbf{e}_Y\mathbf{A}^\top$ live in $\mathbb{F}_q^\ell$, and we assume that they are uniformly distributed. Hence, we have to check on average

$$\frac{\binom{m_1}{v}\binom{m_2}{v}(q-1)^{2v}}{q^\ell}$$

many collisions. For each collision we have to compute $\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$, which would usually require

$$(n-k-\ell)2v\left(\lceil \log_2(q)\rceil^2 + \lceil \log_2(q)\rceil\right)$$

binary operations. However, the algorithm only proceeds if the weight of $\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$ is $t-2v$, hence we can use the concept of early abort. Computing one entry of the vector $\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$ costs

$$2v\left(\lceil \log_2(q)\rceil^2 + \lceil \log_2(q)\rceil\right)$$

binary operations. Thus, by applying early abort, we get that this step costs on average

$$\frac{q}{q-1}(t-2v+1)2v\left(\lceil \log_2(q)\rceil^2 + \lceil \log_2(q)\rceil\right)$$

binary operations.

Finally, the success probability is given by having chosen the correct weight distribution of $\mathbf{e}$; this is exactly the same as over $\mathbb{F}_2$ and given by

$$\binom{m_1}{v}\binom{m_2}{v}\binom{n-k-\ell}{t-2v}\binom{n}{t}^{-1}.$$

Thus, we can conclude. □

Note that we usually set in Stern's algorithm the parameter $m_1 = \lfloor \frac{k}{2} \rfloor$. Hence assuming that k is even we get a nicer formula for the cost, being

$$\begin{aligned}
&\binom{k/2}{v}^{-2}\binom{n-k-\ell}{t-2v}^{-1}\binom{n}{t}\left(\left(\lceil \log_2(q)\rceil + \lceil \log_2(q)\rceil^2\right)\right.\\
&\cdot\left((n-k)^2(n+1) + \binom{k/2}{v}^2(q-1)^{2v-\ell}\min\left\{n-k-\ell, \frac{q}{q-1}(t-2v+1)\right\}2v\right)\\
&+ k\ell\lceil \log_2(q)\rceil^2 + \ell\left(2L_q(k/2,v) + \binom{k/2}{v}(q-1)^v\right)\lceil \log_2(q)\rceil\Big)
\end{aligned}$$

binary operations.

Corollary 4.3.2. *Stern's algorithm over the binary costs*

$$\begin{aligned}
&\binom{m_1}{v}^{-1}\binom{m_2}{v}^{-1}\binom{n-k-\ell}{t-2v}^{-1}\binom{n}{t}\\
&\qquad\cdot\left((n-k)^2(n+1) + \ell\left(L(m_1,v) + L(m_2,v) + \binom{m_2}{v}\right)\right.\\
&\qquad\left.+\frac{\binom{m_1}{v}\binom{m_2}{v}}{2^\ell}\min\{n-k-\ell, 2(t-2v+1)\}2v\right)
\end{aligned}$$

binary operations.

The only difference in the binary is the concept of intermediate sums, as explained in the Section 3.3. For the algorithm over the binary and its full complexity analysis we refer the reader to the appendix, *i.e.*, Section A.2.

Chapter 5

Information Set Decoding in the Lee Metric

In this chapter we change our focus to the Lee metric and to ring-linear codes. For this we will first recall the background on ring-linear coding theory, where we treat the special case of $\mathbb{Z}/4\mathbb{Z}$ separately. We then translate the techniques used for speed-ups in the complexity of ISD algorithms. Finally, we provide the three ISD algorithms (namely Prange, Lee-Brickell and Stern) in the quaternary case and later generalize this to the case $\mathbb{Z}/p^s\mathbb{Z}$ endowed with the Lee metric with a special emphasis on the case $s = 1$.

5.1 Preliminaries

Coding Theory in the Lee Metric In this section we want to introduce the Lee metric. For this we are going to change the ambient space, in which the codes live, from a finite field to a finite ring $\mathcal{R}$. In particular, we will focus on Galois rings $\mathbb{Z}/p^s\mathbb{Z}$, where p is a prime and s is a positive integer.

Let us assume that $\mathcal{R}$ is a finite commutative ring with identity, although the following definition stays true in the non-commutative case.

Note that the since over finite rings there is no notion of dimension, we need the definition of type.

Definition 5.1.1 ($\mathcal{R}$-Linear Code)**.** Let n be a positive integer. A submodule $\mathcal{C}$ of $\mathcal{R}^n$, with $\mid \mathcal{C} \mid = h$, is called an *$\mathcal{R}$-linear code* of length n and *type* h.

The most studied case of ring-linear coding theory is given by integer residue rings $\mathbb{Z}/m\mathbb{Z}$, for some positive integer m.

In this case a $\mathbb{Z}/m\mathbb{Z}$-linear code $\mathcal{C}$ of length n is an additive subgroup of $(\mathbb{Z}/m\mathbb{Z})^n$. We still have the notion of generator matrix, as the matrix which has the code as image, and of parity-check matrix, which has the code as kernel.

Also the definition of information set is still valid, *i.e.*, for a $\mathbb{Z}/m\mathbb{Z}$-linear code $\mathcal{C}$ of length n, we call $I \subset \{1, \ldots, n\}$ an information set, if $\mid \mathcal{C} \mid = \mid \mathcal{C}_I \mid$.

In traditional finite field coding theory we endow $\mathbb{F}_q$ with the Hamming metric to define the weight of a vector wt_H and the distance between two vectors d_H. In ring-linear coding theory over $\mathbb{Z}/m\mathbb{Z}$ we could use the Hamming metric as well, the Lee metric, the homogeneous metric, the Euclidean metric and so on, for an overview see [46]. If we use the Lee metric the corresponding codes are referred to as Lee metric codes.

The Lee weight of an element generalizes the Hamming weight over the binary field, but in contrast to the Hamming weight over $\mathbb{Z}/m\mathbb{Z}$, which only counts the number of nonzero entries, the Lee weight also takes into account the entry itself and its L_1-distance to zero.

Definition 5.1.2 (Lee Weight)**.** For $x \in \mathbb{Z}/m\mathbb{Z}$, the Lee weight of x is defined to be

$$\mathrm{wt}_L(x) = \min\{x, m - x\}.$$

Similarly, for $\mathbf{x} \in (\mathbb{Z}/m\mathbb{Z})^n$, the *Lee weight* of $\mathbf{x}$ is defined to be

$$\mathrm{wt}_L(\mathbf{x}) = \sum_{i=1}^{n} \mathrm{wt}_L(x_i).$$

The Lee weight induces a distance, called Lee distance.

Definition 5.1.3 (Lee Distance)**.** For $\mathbf{x}, \mathbf{y} \in (\mathbb{Z}/m\mathbb{Z})^n$, the *Lee distance* between $\mathbf{x}$ and $\mathbf{y}$ is given by

$$d_L(\mathbf{x}, \mathbf{y}) = \mathrm{wt}_L(\mathbf{x} - \mathbf{y}).$$

To illustrate the Lee metric, let us look at the toy example $m = 8$:

Example 5.1.4 ($\mathbb{Z}/8\mathbb{Z}$).

$$\begin{aligned} \mathrm{wt}_L(0) &= 0 \\ \mathrm{wt}_L(1) = \mathrm{wt}_L(7) &= 1 \\ \mathrm{wt}_L(2) = \mathrm{wt}_L(6) &= 2 \\ \mathrm{wt}_L(3) = \mathrm{wt}_L(5) &= 3 \\ \mathrm{wt}_L(4) &= 4 \end{aligned}$$

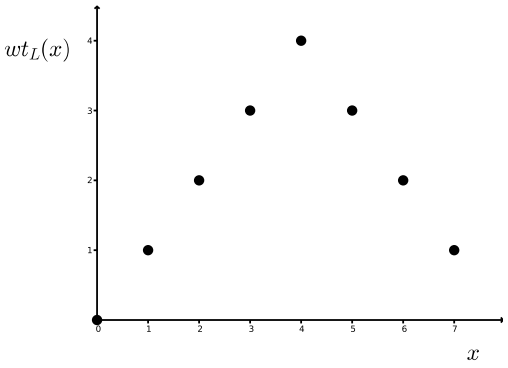

Figure 5.1: Lee weight in $\mathbb{Z}/8\mathbb{Z}$.

When representing $\mathbb{Z}/m\mathbb{Z}$ not as $\{0, \ldots, m-1\}$ but rather symmetrically, *i.e.*, $\{-\left\lfloor \frac{m-1}{2} \right\rfloor, \ldots, \left\lfloor \frac{m}{2} \right\rfloor\}$, the connection to the L_1-norm becomes evident:
$\mid x \mid = \mathrm{wt}_L(x)$, for all $x \in \{-\left\lfloor \frac{m-1}{2} \right\rfloor, \ldots, \left\lfloor \frac{m}{2} \right\rfloor\}$.

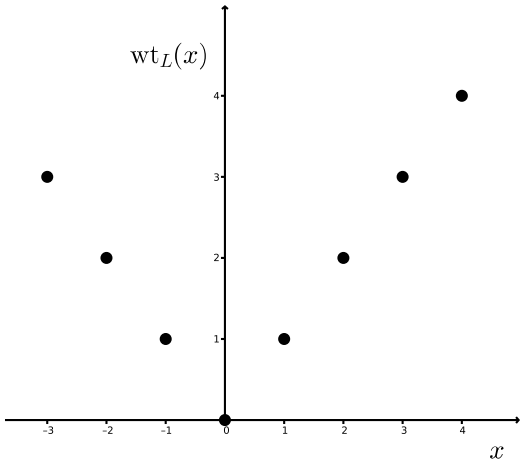

Figure 5.2: Lee weight in $\mathbb{Z}/8\mathbb{Z}$ represented symmetrically.

Let us now restrict to Galois rings $\mathbb{Z}/p^s\mathbb{Z}$. The type of a code $\mathcal{C}$ in $(\mathbb{Z}/p^s\mathbb{Z})^n$ is given by

$$\mid \mathcal{C} \mid = (p^s)^{k_1}(p^{s-1})^{k_2} \cdots p^{k_s}.$$

Thus, an information set is of size $K = \sum_{i=1}^{s} k_i$.

Observe, that the Lee weight of an element $x \in \mathbb{Z}/p^s\mathbb{Z}$ can be between 0 and $\left\lfloor \frac{p^s}{2} \right\rfloor$. Similarly, the Lee weight of an element $\mathbf{x} \in (\mathbb{Z}/p^s\mathbb{Z})^n$ can be between 0 and $\ell \left\lfloor \frac{p^s}{2} \right\rfloor$, where ℓ denotes the support size of $\mathbf{x}$, *i.e.*, the Hamming weight of $\mathbf{x}$.

Note that we usually distinguish between p^s being even or odd, since in the even case there exists exactly one element in $(\mathbb{Z}/p^s\mathbb{Z}) \setminus \{0\}$ having Lee weight $\frac{p^s}{2}$ and for all smaller weights there exist two such elements and in the odd case, for all possible Lee weights in $(\mathbb{Z}/p^s\mathbb{Z}) \setminus \{0\}$, there exist exactly two elements achieving this Lee weight.

We can also define the minimum Lee distance of a code.

Definition 5.1.5 (Minimum Lee Distance)**.** Let $k < n$ be positive integers and let $\mathcal{C}$ be a linear code over $\mathbb{Z}/p^s\mathbb{Z}$. Then, the *minimum Lee distance* of $\mathcal{C}$ is defined as

$$d_L(\mathcal{C}) = \min\{d_L(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}, \mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}\}.$$

Since the Lee distance is induced by the Lee weight, the minimum Lee distance of a code is also given by

$$d_L(\mathcal{C}) = \min\{\mathrm{wt}_L(\mathbf{x}) \mid \mathbf{x} \in \mathcal{C}, \mathbf{x} \neq \mathbf{0}\}.$$

In addition, the following proposition gives the systematic form for the generator matrix and the parity-check matrix.

Proposition 5.1.6 (Systematic Form)**.** *Let $\mathcal{C}$ be a linear code over $\mathbb{Z}/p^s\mathbb{Z}$ of length n and type $| \mathcal{C} | = (p^s)^{k_1}(p^{s-1})^{k_2} \cdots p^{k_s}$. Then $\mathcal{C}$ is permutation equivalent to a code having the following systematic generator matrix $\mathbf{G}$ of size $K \times n$*

$$\begin{pmatrix} \mathrm{Id}_{k_1} & \mathbf{A}_{1,2} & \mathbf{A}_{1,3} & \cdots & \mathbf{A}_{1,s} & \mathbf{A}_{1,s+1} \\ \mathbf{0}_{k_2 \times k_1} & p\,\mathrm{Id}_{k_2} & p\mathbf{A}_{2,3} & \cdots & p\mathbf{A}_{2,s} & p\mathbf{A}_{2,s+1} \\ \mathbf{0}_{k_3 \times k_1} & \mathbf{0}_{k_3 \times k_2} & p^2\,\mathrm{Id}_{k_3} & \cdots & p^2\mathbf{A}_{3,s} & p^2\mathbf{A}_{3,s+1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \mathbf{0}_{k_s \times k_1} & \mathbf{0}_{k_s \times k_2} & \mathbf{0}_{k_s \times k_3} & \cdots & p^{s-1}\,\mathrm{Id}_{k_s} & p^{s-1}\mathbf{A}_{s,s+1} \end{pmatrix}, \tag{5.1.1}$$

where $\mathbf{A}_{i,s+1} \in (\mathbb{Z}/p^{s+1-i}\mathbb{Z})^{k_i \times (n-K)}$ and $\mathbf{A}_{i,j} \in (\mathbb{Z}/p^{s+1-i}\mathbb{Z})^{k_i \times k_j}$ for $j \leq s$.

And $\mathcal{C}$ is permutation equivalent to a code having the following systematic parity-check matrix $\mathbf{H}$ of size $(n-k_1) \times n$

$$\begin{pmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} & \cdots & \mathbf{B}_{1,s-1} & \mathbf{B}_{1,s} & \mathrm{Id}_{n-K} \\ p\mathbf{B}_{2,1} & p\mathbf{B}_{2,2} & \cdots & p\mathbf{B}_{2,s-1} & p\,\mathrm{Id}_{k_s} & \mathbf{0}_{k_s \times (n-K)} \\ p^2\mathbf{B}_{3,1} & p^2\mathbf{B}_{3,2} & \cdots & p^2\,\mathrm{Id}_{k_{s-1}} & \mathbf{0}_{k_{s-1} \times k_s} & \mathbf{0}_{k_{s-1} \times (n-K)} \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ p^{s-1}\mathbf{B}_{s,1} & p^{s-1}\,\mathrm{Id}_{k_2} & \cdots & \mathbf{0}_{k_2 \times k_{s-1}} & \mathbf{0}_{k_2 \times k_s} & \mathbf{0}_{k_2 \times (n-K)} \end{pmatrix}, \tag{5.1.2}$$

where $\mathbf{B}_{1,j} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K) \times k_j}$ and $\mathbf{B}_{i,j} \in (\mathbb{Z}/p^{s+1-i}\mathbb{Z})^{k_{s-i+2} \times k_j}$ for $i > 1$.

The systematic form of the parity-check matrix (5.1.2) plays a crucial role in ISD algorithms.

For the Lee metric there exists an analogue to the Singleton bound.

Theorem 5.1.7 (Singleton Bound in the Lee metric, [96], Theorem 2)**.** *Let $\mathcal{C}$ be a Lee metric code over $\mathbb{Z}/p^s\mathbb{Z}$ of length n, let $k = \log_{p^s}(| \mathcal{C} |)$ and minimum Lee distance d_L. Then it holds that*

$$d_L \leq \left\lfloor \frac{p^s}{2} \right\rfloor (n - \lceil k \rceil + 1)\,.$$

Other important concepts of coding theory for ISD include the number of vectors of a given length and a given weight, and the average weight of an element. Let us start with the latter. In the following lemma, resulting from a Plotkin-type bound in the Lee metric, we compute the average Lee weight of an element in $\mathbb{Z}/p^s\mathbb{Z}$.

Lemma 5.1.8 ([95], Problem 10.15)**.** *Let x be chosen randomly in $\mathbb{Z}/p^s\mathbb{Z}$. The expected Lee weight of x is*

$$\mu_{p^s} = \begin{cases} \frac{p^s}{4} & \text{if } p = 2, \\ \frac{p^{2s}-1}{4p^s} & \text{if } p \neq 2. \end{cases}$$

Proof. If $p = 2$ and hence p^s is even, then summing up all weights in $\mathbb{Z}/p^s\mathbb{Z}$ gives

$$2 \sum_{i=1}^{\frac{p^s-2}{2}} i + \frac{p^s}{2} = \frac{(p^s-2)p^s}{4} + \frac{p^s}{2} = \frac{p^{2s}}{4}.$$

If $p \neq 2$ and hence p^s is odd, then we get

$$2\sum_{i=1}^{\frac{p^s-1}{2}} i = \frac{(p^s-1)(p^s+1)}{4} = \frac{p^{2s}-1}{4}.$$

To get the average we divide both cases by p^s and get the desired formula. □

Next, we want to count the vectors in $(\mathbb{Z}/p^s\mathbb{Z})^n$ having Lee weight w, with

$$0 \leq w \leq n\left\lfloor \frac{p^s}{2} \right\rfloor,$$

i.e., we want to determine

$$F(n,w,p^s) :=\mid \{\mathbf{x} \in (\mathbb{Z}/p^s\mathbb{Z})^n \mid \ \mathrm{wt}_L(\mathbf{x}) = w\} \mid .$$

Up to our best knowledge, a closed and exact formula was first given in [105] by the author in collaboration with Massimo Battaglioni, Paolo Santini, Anna-Lena Horlemann-Trautmann and Edoardo Persichetti.

We will consider two cases: either p^s is even, or p^s is odd. Indeed, in the former case there exists only one element in $\mathbb{Z}/p^s\mathbb{Z}$ having Lee weight $\left\lfloor \frac{p^s}{2} \right\rfloor$, whereas in the latter case there exist two such elements. We will first count the vectors in $(\mathbb{Z}/p^s\mathbb{Z})^n$ having Lee weight w and a fixed size of support ℓ. For this, we introduce

$$f(n,\ell,w,p^s) :=\mid \{\mathbf{x} \in (\mathbb{Z}/p^s\mathbb{Z})^n \ \mid \ \mathrm{wt}_H(\mathbf{x}) = \ell, \ \mathrm{wt}_L(\mathbf{x}) = w\} \mid .$$

For $n,\ell,w \in \mathbb{N}$, with $1 \leq \ell, \mu \leq w$, let us denote by $C(w,\ell,\mu)$ the number of compositions of w into ℓ parts, such that each part is at most of size μ, which can be computed as

$$C(w,\ell,\mu) = \sum_{j=0}^{\min\{\ell, \lfloor \frac{w-\ell}{\mu} \rfloor\}} (-1)^j \binom{\ell}{j} \binom{w-j\mu-1}{\ell-1},$$

as shown in [1]. Let us assume that $C(0,0,\mu) = 1$.

Proposition 5.1.9 ([105], Proposition 8)**.** *Let $n,\ell,w \in \mathbb{N}$ such that $1 \leq w \leq n\left\lfloor \frac{p^s}{2} \right\rfloor$ and $\ell \leq \min\{n,w\}$. Then,*

- *if p^s is even:*

$$f(n,\ell,w,p^s) = \sum_{k=0}^{\min\{\ell, \lfloor \frac{2w}{p^s} \rfloor\}} \binom{n-k}{\ell-k} 2^{\ell-k} \binom{n}{k} C\left(w - k\frac{p^s}{2}, \ell-k, \frac{p^s}{2}-1\right),$$

- *if p^s is odd:*

$$f(n,\ell,w,p^s) = \binom{n}{\ell} 2^\ell C\left(w,\ell,\left\lfloor \frac{p^s}{2} \right\rfloor\right).$$

Proof. When p^s is odd, there are two different elements in $\mathbb{Z}/p^s\mathbb{Z}$ having the same Lee weight $1 \leq v \leq \left\lfloor \frac{p^s}{2} \right\rfloor$, namely v and $p^s - v$. Therefore, in order to compute the number of vectors in $(\mathbb{Z}/p^s\mathbb{Z})^n$ with Lee weight w and support size ℓ, it is sufficient to consider all the compositions of w into ℓ parts with no parts larger than $\left\lfloor \frac{p^s}{2} \right\rfloor$, which are $C\left(w, \ell, \left\lfloor \frac{p^s}{2} \right\rfloor\right)$, all their possible dispositions in the n entries, which are $\binom{n}{\ell}$ and the fact that there are 2 choices with the same Lee weight for any part, yielding the factor 2^ℓ.

In the case where p^s is even, instead, there exists only one element in $\mathbb{Z}/p^s\mathbb{Z}$ having Lee weight $\frac{p^s}{2}$. So, we must separately consider all the compositions of w into ℓ parts containing the element $\frac{p^s}{2}$ exactly $0, 1, \ldots, \min\{\ell, \frac{2w}{p^s}\}$ times and no elements larger than $\frac{p^s}{2}$. For each case we have to take into account all the possible dispositions and consider that there are two choices with the same Lee weight for each part smaller than $\frac{p^s}{2}$, and only one choice for each part equal to $\frac{p^s}{2}$. Let us suppose that $\frac{p^s}{2}$ appears k times in a given composition of w into ℓ parts, which can happen in $\binom{n}{k}$ ways. The remaining weight $w - k\frac{p^s}{2}$ can be composed into $\ell - k$ parts in $C\left(w - k\frac{p^s}{2}, \ell - k, \frac{p^s}{2} - 1\right)$ different ways. The $\ell - k$ parts can be disposed in the remaining $n - k$ positions in $\binom{n-k}{\ell-k}$ ways and for each part (all the remaining parts are smaller than $\frac{p^s}{2}$) there are 2 choices with the same Lee weight, yielding a factor $2^{\ell-k}$. The total amount is then obtained by summing all these separate contributions.

□

Finally, to get the amount of vectors in $(\mathbb{Z}/p^s\mathbb{Z})^n$ having Lee weight w, we only have to sum all $f(n, \ell, w, p^s)$ from $\ell = 1$ to $\ell = \min\{n, w\}$.

Corollary 5.1.10 ([105], Corollary 9)**.** *Let $n \in \mathbb{N}$, and let $1 \leq w \leq n\left\lfloor \frac{p^s}{2} \right\rfloor$. Then,*

$$F(n, w, p^s) = \sum_{\ell=1}^{\min\{n,w\}} f(n, \ell, w, p^s). \tag{5.1.3}$$

Then, the size of a Lee ball is easily seen to be

$$\mid B_L(w, n, p^s) \mid = \sum_{i=0}^{w} F(n, i, p^s).$$

Lastly, we want to give the Gilbert-Varshamov bound also in the Lee metric, for prime finite fields.

Theorem 5.1.11 (Gilbert-Varshamov bound [95], Theorem 10.12)**.** *Let p be an odd prime and let $k < n$ and d_L be positive integers, such that*

$$\frac{p^{n-k+1} - 1}{p - 1} > \frac{\mid V_L(d_L - 1, n, p) \mid -1}{2}. \tag{5.1.4}$$

Then there exists a $[n, k]$ linear Lee code over $\mathbb{F}_p$ with minimum Lee distance at least d_L.

In the case of $s > 1$, we have the following Lee analogue of the Gilbert-Varshamov bound.

Theorem 5.1.12 (Lee Metric Gilbert-Varshamov Bound, [19], Theorem 13.73)**.** *Let n and d_L be positive integers. There exists a linear Lee code $\mathcal{C}$ of length n and minimum Lee distance d_L over $\mathbb{Z}/p^s\mathbb{Z}$, such that*

1. *if $p \neq 2$, then*

$$|\mathcal{C}| < \frac{(p^s)^n}{((|B_L(d-1,n,p^s)|-1)/2+1)(p^s-1)}.$$

2. *if $p = 2$, then*

$$|\mathcal{C}| < \frac{(p^s)^n}{(|B_L(d-1,n,p^s)|-1)(p^s-1)}.$$

Note that the classical Gilbert-Varshamov bound gives a lower bound on the code size and implies the existence of any (not necessarily linear) code. The bound given above is hence a more restricted version which gives a sufficient condition for the existence of a linear code.

The Quaternary Case Similar to the binary case in the Hamming metric, the case $\mathbb{Z}/4\mathbb{Z}$ is a special case for the Lee metric, hence we will treat this case here separately.

Definition 5.1.13 (Quaternary Code)**.** We say that $\mathcal{C}$ is a *quaternary code* of length n, if $\mathcal{C}$ is an additive subgroup of $(\mathbb{Z}/4\mathbb{Z})^n$.

The reason why the quaternary case is special, is due to the Gray map, which gives a connection between traditional finite field theory and $\mathbb{Z}/4\mathbb{Z}$-linear coding theory.

Definition 5.1.14 (Gray Isometry)**.** The *Gray map* is an isometry between $\mathbb{Z}/4\mathbb{Z}$ equipped with the Lee weight and $\mathbb{F}_2^2$ equipped with the Hamming weight and is defined as follows:

$$\begin{aligned}\phi : (\mathbb{Z}/4\mathbb{Z}, \mathrm{wt}_L) &\to (\mathbb{F}_2^2, \mathrm{wt}_H)\\ 0 &\mapsto (0,0),\\ 1 &\mapsto (0,1),\\ 2 &\mapsto (1,1),\\ 3 &\mapsto (1,0).\end{aligned}$$

The Gray map can be extended componentwise to

$$\overline{\phi} : ((\mathbb{Z}/4\mathbb{Z})^n, \mathrm{wt}_L) \to (\mathbb{F}_2^{2n}, \mathrm{wt}_H).$$

Note, however, that the Gray map does not preserve linearity, *i.e.*, the image of a quaternary linear code is generally not linear over $\mathbb{F}_2$.

In the quaternary case, the type of a code $\mathcal{C} \subset (\mathbb{Z}/4\mathbb{Z})^n$ is given by

$$|\mathcal{C}| = 4^{k_1}2^{k_2}.$$

The following proposition defines the quaternary systematic form of the generator matrix and the parity-check matrix.

Proposition 5.1.15 (Quaternary Systematic Form)**.** *Let $\mathcal{C}$ be a quaternary linear code of length n and type $4^{k_1}2^{k_2}$. Then, $\mathcal{C}$ is permutation equivalent to a code having the $(k_1+k_2)\times n$ generator matrix*

$$\mathbf{G} = \begin{pmatrix} \mathrm{Id}_{k_1} & \mathbf{A} & \mathbf{B} \\ \mathbf{0}_{k_2\times k_1} & 2\,\mathrm{Id}_{k_2} & 2\mathbf{C} \end{pmatrix}, \tag{5.1.5}$$

where $\mathbf{A}\in\mathbb{F}_2^{k_1\times k_2}, \mathbf{B}\in(\mathbb{Z}/4\mathbb{Z})^{k_1\times(n-k_1-k_2)}$ and $\mathbf{C}\in\mathbb{F}_2^{k_2\times(n-k_1-k_2)}$. A parity-check matrix of $\mathcal{C}$ is the corresponding permutation of the $(n-k_1)\times n$ matrix

$$\mathbf{H} = \begin{pmatrix} \mathbf{D} & \mathbf{E} & \mathrm{Id}_{n-k_1-k_2} \\ 2\mathbf{F} & 2\,\mathrm{Id}_{k_2} & \mathbf{0}_{k_2\times(n-k_1-k_2)} \end{pmatrix}, \tag{5.1.6}$$

where $\mathbf{D}\in(\mathbb{Z}/4\mathbb{Z})^{(n-k_1-k_2)\times k_1}, \mathbf{E}\in\mathbb{F}_2^{(n-k_1-k_2)\times k_2}$ and $\mathbf{F}\in\mathbb{F}_2^{k_2\times k_1}$.

To compute the number of vectors in $(\mathbb{Z}/4\mathbb{Z})^n$ having Lee weight w, we have to sum over all choices of i entries having Lee weight 2, of course only until $\lceil\frac{w}{2}\rceil$. For the rest of the $n-i$ entries we are missing a Lee weight of $w-2i$. We will achieve this with entries of Lee weight 1, where for each of the $w-2i$ entries, there are two choices: either 1 or 3. We will introduce the following notation for the amount of these vectors:

$$c(n,w) = \sum_{i=0}^{\lceil\frac{w}{2}\rceil} \binom{n}{i}\binom{n-i}{w-2i}2^{w-2i}.$$

With the Gray isometry, we have that the number of vectors in $(\mathbb{Z}/4\mathbb{Z})^n$ having Lee weight w is the same as the number of vectors in $\mathbb{F}_2^{2n}$ having Hamming weight w, which is simply given by $\binom{2n}{w}$. In fact, one can also check that

$$c(n,w) = \binom{2n}{w}.$$

Thus, we can easily derive the Gilbert-Varshamov bound for quaternary codes endowed with the Lee metric.

Proposition 5.1.16 (Gilbert-Varshamov Bound, [19], Theorem 13.73)**.** *Let $k<n$ and d_L be positive integers. There exists a linear quaternary code $\mathcal{C}$ of length n and minimum Lee distance d_L, such that*

$$|\,\mathcal{C}\,| < \frac{4^n}{\left(\sum_{i=0}^{d_L-1}\binom{2n}{i}-1\right)3+1}.$$

The nice formula for the size of the Lee ball over the quaternary and the Gray isometry will in fact also influence the ISD algorithms that we will provide in this chapter. Hence it makes sense to keep the quaternary case separate also for the ISD algorithms.

Note that in [55] the author in collaboration with Anna-Lena Horlemann-Trautmann gave the adaption of the McEliece cryptosystem and the Niederreiter cryptosystem over

the quaternary case. This can be done almost in a straight-forward manner, except for the fact that some parts of the generator matrix $\mathbf{G}$ live in $\mathbb{F}_2$, while others live over $\mathbb{Z}/4\mathbb{Z}$, which implies that also the message should be split in the same manner. A full description of the frameworks over the quaternary can be found in the appendix, namely Section A.3.

Syndrome Decoding Problem in the Lee Metric Analogously to the SDP in the Hamming metric, we can define the Lee syndrome decoding problem, as follows.

Problem 4 (Lee - Syndrome Decoding Problem (L-SDP)). Let $p^s \geq 4, k, n$ be positive integers. Given $\mathbf{H} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k)\times n}$, $\mathbf{s} \in (\mathbb{Z}/p^s\mathbb{Z})^{n-k} \setminus \{\mathbf{0}_{n-k}\}$ and $t \in \mathbb{N}$, is there a vector $\mathbf{e} \in (\mathbb{Z}/p^s\mathbb{Z})^n$ such that $\mathrm{wt}_L(\mathbf{e}) \leq t$ and $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$?

This problem was proven to be NP-complete in [105] by the author in collaboration with Massimo Battaglioni, Paolo Santini, Anna-Lena Horlemann-Trautmann and Edoardo Persichetti.

Note that the restriction $p^s \geq 4$ is not a problem, since in the cases $p^s = 2, 3$ the Lee metric and the Hamming metric coincide.

5.1.1 Techniques

In order to present the adaption of the ISD algorithms, we have seen in Chapter 4, to the Lee metric, we first need to translate the techniques we have introduced in Section 3.3 to the Lee metric.

Note that over the Hamming metric the binary case was a special case, *e.g.* for the intermediate sums. In the Lee metric the case of $\mathbb{Z}/4\mathbb{Z}$ is similar to the binary, as it is the most studied ambient space for Lee metric codes. Therefore, most of the theory of Lee metric codes is known for $\mathbb{Z}/4\mathbb{Z}$. This is mainly due to the Gray isometry, which will also be a very helpful tool for the techniques we use in ISD. In turn, we get that for the ISD algorithms as well, the quaternary case should be treated separately, as some techniques only apply there and not over general Galois rings $\mathbb{Z}/p^s\mathbb{Z}$.

In this section we hence introduce the techniques that we will use throughout this chapter. We will first introduce the techniques for $\mathbb{Z}/4\mathbb{Z}$, then we will show how they work over $\mathbb{Z}/p^s\mathbb{Z}$, if they work at all, and if needed also treat the special case $s = 1$.

Let us again fix first the cost of operations we will use throughout this chapter. We assume that one addition over $\mathbb{Z}/p^s\mathbb{Z}$ costs $\lceil \log_2(p^s) \rceil$ binary operations and one multiplication costs $\lceil \log_2(p^s) \rceil^2$ binary operations. Again, we are not using the fastest algorithms known, which will give a broad estimate of the real cost. Over $\mathbb{Z}/4\mathbb{Z}$ we have the special case, that both the addition and the multiplication just cost 2 binary operations, *i.e.*, computing $a + b$ for $a, b \in \mathbb{Z}/4\mathbb{Z}$ costs $2 = \lceil \log_2(4) \rceil$ binary operations and also computing $a \cdot b$ for $a, b \in \mathbb{Z}/4\mathbb{Z}$ costs only 2 binary operations since it reduces to either $a \cdot 2 = a + a$ or $a \cdot 3 = -a$, since $b = 0$ and $b = 1$ come with no cost.

Number of Iterations As over the Hamming metric, also in the Lee metric, we have to compute the average number of iterations needed for an ISD algorithm. This is given by the reciprocal of the success probability of one iteration where we have fixed an information set.

The success probability, depending on the weight distribution of the error vector that we assume in this ISD algorithm, is given by how many vectors there are which satisfy the assumed weight distribution, divided by the total number of vectors of the given Lee weight.

As we have seen in the preliminaries, over $\mathbb{Z}/4\mathbb{Z}$, the number of vectors of length n and Lee weight $0 \leq t \leq 2n$ is simply given through the Gray isometry as

$$\binom{2n}{t}.$$

For the general case of $\mathbb{Z}/p^s\mathbb{Z}$ we do not have such a nice formula. Nevertheless, we can use the closed and exact formula given in Corollary 5.1.10: for $F(n, t, p^s)$.

For example, we are looking for an error vector of length n and Lee weight t. Assuming that the error vector has Lee weight w in k positions and the remaining Lee weight $t - w$ in the remaining $n - k$ positions, the success probability of such a weight distribution is given by

$$\binom{2k}{w}\binom{2(n-k)}{t-w}\binom{2n}{t}^{-1}$$

over $\mathbb{Z}/4\mathbb{Z}$ and by

$$F(k, w, p^s)F(n-k, t-w, p^s)F(n, t, p^s)^{-1}$$

over $\mathbb{Z}/p^s\mathbb{Z}$.

And in turn, the number of iterations needed on average would be given by

$$\binom{2k}{w}^{-1}\binom{2(n-k)}{t-w}^{-1}\binom{2n}{t}$$

over $\mathbb{Z}/4\mathbb{Z}$ and by

$$F(k, w, p^s)^{-1}F(n-k, t-w, p^s)^{-1}F(n, t, p^s)$$

over $\mathbb{Z}/p^s\mathbb{Z}$.

The case $s = 1$ does not need to be treated as a special case for the number of iterations, one can just set $s = 1$ in the formula over $\mathbb{Z}/p^s\mathbb{Z}$.

Early Abort The concept of early abort can easily be adapted to the case of the Lee metric. Over $\mathbb{Z}/4\mathbb{Z}$, there is one element having Lee weight 0, two elements having Lee weight 1 and one element having Lee weight 2, hence on average a random element in $\mathbb{Z}/4\mathbb{Z}$ has Lee weight 1.

We have seen in Lemma 5.1.8, that the average Lee weight of a random element in $\mathbb{Z}/p^s\mathbb{Z}$ is given by μ_{p^s}. It is enough to provide an example for the general case $\mathbb{Z}/p^s\mathbb{Z}$, since μ_4 is in fact 1.

We provide an example also for this technique: assume that we have to compute $\mathbf{xA}$, for $\mathbf{x} \in (\mathbb{Z}/p^s\mathbb{Z})^k$ of Lee weight t and $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{k\times n}$. Usually computing $\mathbf{xA}$ would cost at most $n \min\{t, k\}\left(\lceil \log_2(p^s)\rceil^2 + \lceil \log_2(p^s)\rceil\right)$ binary operations, since $\mathbf{x}$ has support at most $\min\{t, k\}$.

However, our algorithm only proceeds if $\mathrm{wt}_L(\mathbf{xA}) = w$, and thus after computing $\mu_{p^s}^{-1}(w+1)$ entries, we should have exceeded the target weight w and can abort. Since computing only one entry of the resulting vector costs

$$\min\{t,k\}\left(\lceil\log_2(p^s)\rceil^2 + \lceil\log_2(p^s)\rceil\right)$$

binary operations, the maximal cost of this step is given by

$$\mu_{p^s}^{-1}(w+1)\min\{t,k\}\left(\lceil\log_2(p^s)\rceil^2 + \lceil\log_2(p^s)\rceil\right)$$

binary operations, instead of the previous

$$n\min\{t,k\}\left(\lceil\log_2(p^s)\rceil^2 + \lceil\log_2(p^s)\rceil\right).$$

This only provides a speed up, if $\mu_{p^s}^{-1}(w+1) < n$, hence in the algorithms we will usually have as a cost

$$\min\{\mu_{p^s}^{-1}(w+1), n\}\min\{t,k\}\left(\lceil\log_2(p^s)\rceil^2 + \lceil\log_2(p^s)\rceil\right)$$

binary operations.

Number of Collisions Also the concept of the average number of collisions can be adapted to the Lee metric.

The condition that the resulting vectors are uniformly distributed is easily verified for the case $s = 1$, but might need justification over $\mathbb{Z}/p^s\mathbb{Z}$. Assume that the systematic form of the parity-check matrix is given by the following $(n-k_1)\times n$ matrix

$$\mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathrm{Id}_{n-K} \\ p\mathbf{B} & \mathbf{0}_{(K-k_1)\times(n-K)} \end{pmatrix},$$

where $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K)\times K}$ and $\mathbf{B} \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{(K-k_1)\times K}$. Clearly, in an ISD algorithm we assume that the matrix $\mathbf{H}$ was completely random, but know that we have $K-k_1$ rows that are divisible by p. We can hence assume that $\mathbf{A}$, as well as $\mathbf{B}$, are random. Thus $\mathbf{e}_I\mathbf{A}^\top$ should remain random, as it is adding and subtracting random rows of $\mathbf{A}^\top$.

Let us also give an example for this technique; assume that we only proceed in the algorithm if

$$\mathbf{x} + \mathbf{y} = \mathbf{s},$$

for a fixed $\mathbf{s} \in (\mathbb{Z}/p^s\mathbb{Z})^k$ and for all $\mathbf{x} \in (\mathbb{Z}/p^s\mathbb{Z})^k$ of Lee weight v and all $\mathbf{y} \in (\mathbb{Z}/p^s\mathbb{Z})^k$ of Lee weight w. To verify this condition we have to go through all $(\mathbf{x},\mathbf{y}) \in S = \{(\mathbf{x},\mathbf{y}) \in (\mathbb{Z}/p^s\mathbb{Z})^k \mid \mathrm{wt}_L(\mathbf{x}) = v, \mathrm{wt}_L(\mathbf{y}) = w\}$. Thus giving a cost of

$$F(k,v,p^s)F(k,w,p^s)\min\{k, v+w\}\lceil\log_2(p^s)\rceil$$

binary operations over $\mathbb{Z}/p^s\mathbb{Z}$ and in the special case where $p^s = 4$ this costs

$$\binom{2k}{v}\binom{2k}{w}2\min\{k, v+w\}$$

binary operations. Assume that as a subsequent step we have to compute for all such $(\mathbf{x}, \mathbf{y}) \in S$ the vector $\mathbf{Ax} - \mathbf{By}$, for fixed $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{n\times k}$ and $\mathbf{B} \in (\mathbb{Z}/p^s\mathbb{Z})^{n\times k}$. Usually this would cost

$$F(k, v, p^s)F(k, w, p^s)(\min\{k, v\} + \min\{k, w\})n\left(\lceil \log_2(p^s)\rceil + \lceil \log_2(p^s)\rceil^2\right)$$

binary operations. However, using the concept of average number of collisions, we only have to go through

$$\frac{|\ S\ |}{(p^s)^n} = F(k, v, p^s)F(k, w, p^s)p^{-sn}$$

many collisions on average. Thus this step would require

$$F(k, v, p^s)F(k, w, p^s)p^{-sn}(\min\{k, v\} + \min\{k, w\})n\left(\lceil \log_2(p^s)\rceil + \lceil \log_2(p^s)\rceil^2\right)$$

binary operations over $\mathbb{Z}/p^s\mathbb{Z}$ and

$$\binom{2k}{v}\binom{2k}{w}4^{-n}2n(\min\{k, v\} + \min\{k, w\})$$

binary operations over $\mathbb{Z}/4\mathbb{Z}$.

Intermediate Sums Unfortunately, we only know how to adapt the concept of intermediate sums in the quaternary case. To adapt this concept also to the case $\mathbb{Z}/p^s\mathbb{Z}$ remains an open problem. Hence, we will now restrict to $\mathbb{Z}/4\mathbb{Z}$.

Let $\mathbf{A} \in (\mathbb{Z}/4\mathbb{Z})^{k\times n}$, and assume that we want to compute $\mathbf{xA}$ for all $\mathbf{x} \in (\mathbb{Z}/4\mathbb{Z})^k$ of Lee weight t. This would usually cost $2n\min\{t, k\}$ binary operations, for each $\mathbf{x}$. If we have to compute this for all $\mathbf{x} \in (\mathbb{Z}/4\mathbb{Z})^k$ of Lee weight t then we would end up with a cost of

$$2n\min\{t, k\}\binom{2k}{t}$$

binary operations.

However, we can first compute $\mathbf{xA}$ for all $\mathbf{x} \in (\mathbb{Z}/4\mathbb{Z})^k$ of Lee weight 1, thus just outputting the rows of $\mathbf{A}$ and $-\mathbf{A}$, which comes with no cost, for this we get $2k$ resulting vectors.

As a next step, we compute $\mathbf{xA}$ for all $\mathbf{x} \in (\mathbb{Z}/4\mathbb{Z})^k$ of Lee weight 2. If the support size of $\mathbf{x}$ is 2, then this is the same as adding two distinct rows of $\pm\mathbf{A}$, thus coming with a cost of less than $\binom{2k}{2}2n$ binary operations. If the support size of $\mathbf{x}$ is 1, then we just have to add a row of $\mathbf{A}$ to itself, costing $k2n$ binary operations.

As a next step, we compute $\mathbf{xA}$ for all $\mathbf{x} \in (\mathbb{Z}/4\mathbb{Z})^k$ of Lee weight 3. Again, if the support size of $\mathbf{x}$ is 3, then this is the same as adding a new row of $\pm\mathbf{A}$ to $\mathbf{yA}$, for $\mathbf{y}$ having Lee weight 2 and support size 2, which we already computed, hence this comes with a cost of less than $\binom{2k}{3}2n$ binary operations. If the support size of $\mathbf{x}$ is 2, then we have to add $\mathbf{yA}$ for $\mathbf{y}$ having Lee weight 2 and support size 1 (which we already computed) to a row of $\pm\mathbf{A}$. Hence this costs $\binom{2k}{3}2n$ binary operations, as well.

If we proceed in this way, until we compute $\mathbf{xA}$ for all $\mathbf{x} \in (\mathbb{Z}/4\mathbb{Z})^k$ of Hamming weight t, the cost of this step can be bounded from above by

$$2n\overline{L}_4(k,t)$$

binary operations, where we define

$$\overline{L}_4(k,t) = \sum_{i=2}^{t} \binom{2k}{i}.$$

Over the general case $\mathbb{Z}/p^s\mathbb{Z}$, keeping track of all possible support sizes, hence all partitions of $j \leq t$, is too cumbersome. Thus, over $\mathbb{Z}/p^s\mathbb{Z}$ we will use the cost

$$F(n,t,p^s)\min\{k,t\}n\left(\lceil \log_2(p^s)\rceil + \lceil \log_2(p^s)\rceil^2\right)$$

binary operations for this step.

5.2 Information Set Decoding over $\mathbb{Z}_4$ in the Lee Metric

Since the case $\mathbb{Z}/4\mathbb{Z}$ is a special one for the Lee metric, we will first focus on this case, where we also have more tools available.

Information set decoding over $\mathbb{Z}/4\mathbb{Z}$ equipped with the Lee metric was studied in [55] in collaboration with Anna-Lena Horlemann-Trautmann.

5.2.1 Prange's algorithm

In this section we formulate Prange's algorithm over $\mathbb{Z}/4\mathbb{Z}$ equipped with the Lee metric.

Recall that in Prange's algorithm we assume that there exists an information set I that is disjoint to the support of the error vector $\mathrm{Supp}(\mathbf{e})$.

The structure of Prange's algorithm over $\mathbb{Z}/4\mathbb{Z}$ is similar to the original algorithm of Prange. A crucial difference is the systematic form of the parity-check matrix over $\mathbb{Z}/4\mathbb{Z}$, hence we will illustrate the algorithm again for the choice $I = \{1,\dots,k_1+k_2\}$, and let us denote by $J = I^C$. To bring the parity-check matrix $\mathbf{H} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1)\times n}$ into systematic form, we multiply $\mathbf{H}$ by an invertible matrix $\mathbf{U} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1)\times(n-k_1)}$. Since we assume that no errors occur in the information set, we have that $\mathbf{e} = (\mathbf{0}_{k_1+k_2}, \mathbf{e}_J)$ with $\mathrm{wt}_L(\mathbf{e}_J) = t$. We are in the following situation

$$\begin{aligned}\mathbf{eH}^\top\mathbf{U}^\top &= \begin{pmatrix}\mathbf{0}_{k_1+k_2} & \mathbf{e}_J\end{pmatrix}\begin{pmatrix}\mathbf{A}^\top & 2\mathbf{C}^\top \\ \mathrm{Id}_{n-k_1-k_2} & \mathbf{0}_{(n-k_1-k_2)\times k_2}\end{pmatrix} \\ &= \begin{pmatrix}\mathbf{s}_1 & 2\mathbf{s}_2\end{pmatrix} = \mathbf{sU}^\top,\end{aligned}$$

where $\mathbf{A} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1-k_2)\times(k_1+k_2)}$, $\mathbf{C} \in (\mathbb{Z}/2\mathbb{Z})^{k_2\times(k_1+k_2)}$ and $\mathbf{s}_1 \in (\mathbb{Z}/4\mathbb{Z})^{n-k_1-k_2}$, $\mathbf{s}_2 \in (\mathbb{Z}/2\mathbb{Z})^{k_2}$.

It follows that $\mathbf{e}_J = \mathbf{s}_1$ and $\mathbf{0}_{k_2} = \mathbf{s}_2$, and hence, we are only left with checking that the Lee weight of $\mathbf{s}_1$ is t and that the Lee weight of $\mathbf{s}_2$ is zero.

We will now give the algorithm of Prange in its full generality, *i.e.*, we are not restricting to the choice of I and J, that we made before to illustrate the algorithm.

Algorithm 4 Prange's Algorithm over $\mathbb{Z}/4\mathbb{Z}$ in the Lee metric

Input: $\mathbf{H} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k)\times n}$, $\mathbf{s} \in (\mathbb{Z}/4\mathbb{Z})^{n-k_1}$, $t \in \mathbb{N}$.
Output: $\mathbf{e} \in (\mathbb{Z}/4\mathbb{Z})^n$ with $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$ and $\mathrm{wt}_L(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size $k_1 + k_2$ and define $J = I^C$.
2: Compute $\mathbf{U} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1)\times(n-k_1)}$, such that

$$(\mathbf{U}\mathbf{H})_I = \begin{pmatrix} \mathbf{A} \\ 2\mathbf{C} \end{pmatrix} \quad \text{and } (\mathbf{U}\mathbf{H})_J = \begin{pmatrix} \mathrm{Id}_{n-k_1-k_2} \\ \mathbf{0}_{k_2\times(n-k_1-k_2)} \end{pmatrix},$$

where $\mathbf{A} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1-k_2)\times(k_1+k_2)}$, $\mathbf{C} \in (\mathbb{Z}/2\mathbb{Z})^{k_2\times(k_1+k_2)}$.
3: Compute $\mathbf{s}\mathbf{U}^\top = \begin{pmatrix} \mathbf{s}_1 & 2\mathbf{s}_2 \end{pmatrix}$, where $\mathbf{s}_1 \in (\mathbb{Z}/4\mathbb{Z})^{n-k_1-k_2}$ and $\mathbf{s}_2 \in (\mathbb{Z}/2\mathbb{Z})^{k_2}$.
4: **if** $\mathrm{wt}_L(\mathbf{s}_1) = t$ and $\mathrm{wt}_L(\mathbf{s}_2) = 0$ **then**
5: Return $\mathbf{e}$, such that $\mathbf{e}_I = \mathbf{0}_{k_1+k_2}$ and $\mathbf{e}_J = \mathbf{s}_1$.
6: Start over with Step 1 and a new selection of I.

We now provide a complexity estimate of Prange's algorithm over $\mathbb{Z}/4\mathbb{Z}$ in the Lee metric.

Theorem 5.2.1. *Prange's algorithm over $\mathbb{Z}/4\mathbb{Z}$ equipped with the Lee metric requires on average*

$$\binom{2(n-k_1-k_2)}{t}^{-1}\binom{2n}{t}2(n-k_1)^2(n+1)$$

binary operations.

Proof. One iteration of Algorithm 4 over $\mathbb{Z}/4\mathbb{Z}$ only consists in bringing $\mathbf{H}$ into systematic form and to apply the same row operations on the syndrome; thus, the cost can be assumed equal to that of computing $\mathbf{U}\begin{pmatrix}\mathbf{H} & \mathbf{s}^\top\end{pmatrix}$, *i.e.*,

$$2(n-k_1)^2(n+1)$$

binary operations.

The success probability is given by having chosen the correct weight distribution of $\mathbf{e}$. In this case, this is given by the amount of vectors which have Lee weight t in the redundant set of size $n-k_1-k_2$, divided by the amount of vectors of length n and Lee weight t, *i.e.*,

$$\binom{2(n-k_1-k_2)}{t}\binom{2n}{t}^{-1}.$$

Then, the estimated overall cost of Prange's ISD algorithm over $\mathbb{Z}/4\mathbb{Z}$ is given as in the claim. □

5.2.2 Lee-Brickell's algorithm

In this section we want to formulate the algorithm of Lee-Brickell in $\mathbb{Z}/4\mathbb{Z}$ equipped with the Lee metric. Such a formulation was provided in [55] in collaboration with Anna-Lena Horlemann-Trautmann. In the following we will use a slightly modified version of the cost, in order to match the rest of the algorithms.

For this recall that in Lee-Brickell's algorithm we assume that there exists an information set such that v errors happen inside the information set and $t-v$ outside.

The idea of the algorithm over $\mathbb{Z}/4\mathbb{Z}$ coincides with the original idea over the binary, nevertheless, due to the more complicated systematic form, we will illustrate the structure for the choice $I = \{1, \dots, k_1 + k_2\}$, and $J = I^C$. As usual, we bring the parity-check matrix $\mathbf{H} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1)\times n}$ into systematic form by multiplying by an invertible matrix $\mathbf{U} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1)\times(n-k_1)}$. We can write the error vector partitioned into the information set part I and the non-information set part J as $\mathbf{e} = (\mathbf{e}_I, \mathbf{e}_J)$, with $\mathrm{wt}_L(\mathbf{e}_I) = v$ and $\mathrm{wt}_L(\mathbf{e}_J) = t - v$. Hence, we get

$$\begin{aligned}\mathbf{e}\mathbf{H}^\top\mathbf{U}^\top &= \begin{pmatrix}\mathbf{e}_I & \mathbf{e}_J\end{pmatrix}\begin{pmatrix}\mathbf{A}^\top & 2\mathbf{C}^\top \\ \mathrm{Id}_{n-k_1-k_2} & \mathbf{0}_{(n-k_1-k_2)\times k_2}\end{pmatrix} \\ &= \begin{pmatrix}\mathbf{s}_1 & 2\mathbf{s}_2\end{pmatrix} = \mathbf{s}\mathbf{U}^\top,\end{aligned}$$

where $\mathbf{A} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1-k_2)\times(k_1+k_2)}$, $\mathbf{C} \in (\mathbb{Z}/2\mathbb{Z})^{k_2\times(k_1+k_2)}$ and $\mathbf{s}_1 \in (\mathbb{Z}/4\mathbb{Z})^{n-k_1-k_2}, \mathbf{s}_2 \in (\mathbb{Z}/2\mathbb{Z})^{k_2}$.

From this we get the following two conditions

$$\mathbf{e}_I\mathbf{A}^\top + \mathbf{e}_J = \mathbf{s}_1, \tag{5.2.1}$$

$$2\mathbf{e}_I\mathbf{C}^\top = 2\mathbf{s}_2. \tag{5.2.2}$$

For Condition (5.2.2) to be satisfied we go through all $\mathbf{e}_I$ of length $k_1 + k_2$ and Lee weight v and check if $2\mathbf{e}_I\mathbf{C}^\top = 2\mathbf{s}_2$. For Condition (5.2.1) to be verified we define $\mathbf{e}_J = \mathbf{s}_1 - \mathbf{e}_I\mathbf{A}^\top$, hence we are only left with checking the Lee weight of $\mathbf{e}_J$.

We will now give the algorithm of Lee-Brickell in its full generality.

In the following theorem we provide a complexity analysis of Lee-Brickell's algorithm over $\mathbb{Z}/4\mathbb{Z}$ equipped with the Lee metric.

Theorem 5.2.2. *Lee-Brickell's algorithm over $\mathbb{Z}/4\mathbb{Z}$ in the Lee metric requires on average*

$$\begin{aligned}&\binom{2(k_1+k_2)}{v}^{-1}\binom{2(n-k_1-k_2)}{t-v}^{-1}\binom{2n}{t}(2(n-k_1)^2(n+1) \\ &+ \overline{L}_4(k_1+k_2, v)2k_2 + \binom{2(k_1+k_2)}{v}\min\{t-v+1, n-k_1-k_2\} \\ &\cdot 2\min\{v, k_1+k_2\})\end{aligned}$$

binary operations.

Algorithm 5 Lee-Brickell's Algorithm over $\mathbb{Z}/4\mathbb{Z}$ in the Lee metric

Input: $\mathbf{H} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1)\times n}$, $\mathbf{s} \in (\mathbb{Z}/4\mathbb{Z})^{n-k_1}$, $t \in \mathbb{N}$, $v < \min\{t, 2(k_1+k_2)\}$.
Output: $\mathbf{e} \in (\mathbb{Z}/4\mathbb{Z})^n$ with $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$ and $\mathrm{wt}_L(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size $k_1 + k_2$ and define $J = I^C$.
2: Compute $\mathbf{U} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1)\times(n-k_1)}$, such that

$$(\mathbf{U}\mathbf{H})_I = \begin{pmatrix} \mathbf{A} \\ 2\mathbf{C} \end{pmatrix} \quad \text{and } (\mathbf{U}\mathbf{H})_J = \begin{pmatrix} \mathrm{Id}_{n-k_1-k_2} \\ \mathbf{0}_{k_2\times(n-k_1-k_2)} \end{pmatrix},$$

where $\mathbf{A} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1-k_2)\times(k_1+k_2)}$ and $\mathbf{C} \in (\mathbb{Z}/2\mathbb{Z})^{k_2\times(k_1+k_2)}$.
3: Compute $\mathbf{s}\mathbf{U}^\top = \begin{pmatrix} \mathbf{s}_1 & 2\mathbf{s}_2 \end{pmatrix}$, where $\mathbf{s}_1 \in (\mathbb{Z}/4\mathbb{Z})^{n-k_1-k_2}$ and $\mathbf{s}_2 \in (\mathbb{Z}/2\mathbb{Z})^{k_2}$.
4: **for** $\mathbf{e}_I \in (\mathbb{Z}/4\mathbb{Z})^{k_1+k_2}$ with $\mathrm{wt}_L(\mathbf{e}_I) = v$ **do**
5: **if** $\mathbf{e}_I\mathbf{C}^\top = \mathbf{s}_2$ **then**
6: **if** $\mathrm{wt}_L(\mathbf{s}_1 - \mathbf{e}_I\mathbf{A}^\top) = t - v$ **then**
7: Return $\mathbf{e}$, such that $\mathbf{e}_I = \mathbf{e}_I$ and $\mathbf{e}_J = \mathbf{s}_1 - \mathbf{e}_I A$.
8: Start over with Step 1 and a new selection of I.

Proof. As usual, the first step of the algorithm is to bring the parity-check matrix into systematic form by multiplying by $\mathbf{U}$ and computing $\mathbf{s}\mathbf{U}^\top$. A broad estimate of this cost is given by

$$2(n-k_1)^2(n+1)$$

binary operations.

In the next step, we go through all $\mathbf{e}_I \in (\mathbb{Z}/4\mathbb{Z})^{k_1+k_2}$ of Lee weight v, and check if $\mathbf{e}_I\mathbf{C}^\top = \mathbf{s}_2$. This would usually require $2\min\{v, k_1+k_2\}k_2$ binary operations for each $\mathbf{e}_I$. Hence, in total this step would cost

$$\binom{2(k_1+k_2)}{v} 2\min\{v, k_1+k_2\}k_2$$

binary operations. However, we can use the concept of intermediate sums over $\mathbb{Z}/4\mathbb{Z}$ to compute $\mathbf{e}_I\mathbf{C}^\top$ for all $\mathbf{e}_I \in (\mathbb{Z}/4\mathbb{Z})^{k_1+k_2}$ of Lee weight v, which costs

$$\overline{L}_4(k_1+k_2, v)2k_2$$

binary operations, instead of the previous

$$\binom{2(k_1+k_2)}{v} 2\min\{v, k_1+k_2\}k_2$$

binary operations.

Finally, we compute $\mathbf{s}_1 - \mathbf{e}_I\mathbf{A}^\top$, for all $\mathbf{e}_I \in (\mathbb{Z}/4\mathbb{Z})^{k_1+k_2}$ of Lee weight v. This would usually require

$$\binom{2(k_1+k_2)}{v} 2\min\{v, k_1+k_2\}(n-k_1-k_2)$$

binary operations, since $\mathbf{e}_I$ has support size of at most $\min\{v, k_1 + k_2\}$. However, the algorithm only proceeds if the Lee weight of $\mathbf{s}_1 - \mathbf{e}_I\mathbf{A}^\top$ is $t - v$, hence we can use the concept of early abort. Since computing one entry of the vector $\mathbf{s}_1 - \mathbf{e}_I\mathbf{A}^\top$ costs $2\min\{v, k_1 + k_2\}$ binary operations, this step costs on average

$$\binom{2(k_1 + k_2)}{v}(t - v + 1)2\min\{v, k_1 + k_2\}$$

binary operations.

The success probability is given by

$$\binom{2(k_1 + k_2)}{v}\binom{2(n - k_1 - k_2)}{t - v}\binom{2n}{t}^{-1}.$$

By multiplying the number of iterations needed, *i.e.*, the reciprocal of the success probability, with the cost of one iteration, we get the claim. □

5.2.3 Stern's algorithm

In this section we want to formulate Stern's algorithm over $\mathbb{Z}/4\mathbb{Z}$ equipped with the Lee metric. Such an algorithm was provided in [55] in collaboration with Anna-Lena Horlemann-Trautmann. The cost we provide here will be slightly different to the one in the paper, in order to match the other costs.

Recall that in Stern's algorithm we have partitioned the information set into two sets and ask for Lee weight v in both of the sets. Further, we assume that there is a zero-window of size ℓ outside the information set, where no errors happen.

As for the previous algorithms, we will illustrate also Stern's algorithm for the choice of information set being $I = \{1, \dots, k_1 + k_2\}$, and the zero-window being $Z = \{k_1 + k_2 + 1, \dots, k_1 + k_2 + \ell\}$, and finally the remaining set $J = (I \cup Z)^C$.

Assume that $\mathbf{U} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1)\times(n-k_1)}$, is such that $\mathbf{UH}$ is in systematic form. We can write the error vector partitioned into the information set part I, the zero-window part Z and the remaining part J as $\mathbf{e} = (\mathbf{e}_I, \mathbf{0}_\ell, \mathbf{e}_J)$, with $\mathrm{wt}_L(\mathbf{e}_I) = 2v$ and $\mathrm{wt}_L(\mathbf{e}_J) = t - 2v$. Therefore, we can write

$$\begin{aligned}
&\mathbf{e}\mathbf{H}^\top\mathbf{U}^\top \\
&= \begin{pmatrix}\mathbf{e}_I & \mathbf{0}_\ell & \mathbf{e}_J\end{pmatrix}\begin{pmatrix}\mathbf{A}^\top & \mathbf{B}^\top & 2\mathbf{C}^\top \\ \mathrm{Id}_\ell & \mathbf{0}_{\ell\times(n-k_1-k_2-\ell)} & \mathbf{0}_{\ell\times k_2} \\ \mathbf{0}_{(n-k_1-k_2-\ell)\times\ell} & \mathrm{Id}_{n-k_1-k_2-\ell} & \mathbf{0}_{(n-k_1-k_2-\ell)\times k_2}\end{pmatrix} \\
&= \begin{pmatrix}\mathbf{s}_1 & \mathbf{s}_2 & 2\mathbf{s}_3\end{pmatrix} = \mathbf{s}\mathbf{U}^\top,
\end{aligned}$$

where $\mathbf{A} \in (\mathbb{Z}/4\mathbb{Z})^{\ell\times(k_1+k_2)}$, $\mathbf{B} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1-k_2-\ell)\times(k_1+k_2)}$, $\mathbf{C} \in (\mathbb{Z}/2\mathbb{Z})^{k_2\times(k_1+k_2)}$ and $\mathbf{s}_1 \in (\mathbb{Z}/4\mathbb{Z})^\ell, \mathbf{s}_2 \in (\mathbb{Z}/4\mathbb{Z})^{n-k_1-k_2-\ell}, \mathbf{s}_3 \in (\mathbb{Z}/2\mathbb{Z})^{k_2}$.

From this we get the following three conditions

$$\mathbf{e}_I\mathbf{A}^\top = \mathbf{s}_1, \tag{5.2.3}$$
$$\mathbf{e}_I\mathbf{B}^\top + \mathbf{e}_J = \mathbf{s}_2, \tag{5.2.4}$$
$$2\mathbf{e}_I\mathbf{C}^\top = 2\mathbf{s}_3. \tag{5.2.5}$$

We partition the information set I into the sets X and Y, for the sake of clarity, assume that k_1+k_2 is even, $m = (k_1+k_2)/2$, $X = \{1, \ldots, m\}$ and $Y = \{m+1, \ldots, k_1+k_2\}$. We can hence write $\mathbf{e}_I = (\mathbf{e}_X, \mathbf{e}_Y)$, and Condition (5.2.3) and Condition (5.2.5) become

$$\begin{aligned} \sigma_X(\mathbf{e}_X)\mathbf{A}^\top &= \mathbf{s}_1 - \sigma_Y(\mathbf{e}_Y)\mathbf{A}^\top, \\ \sigma_X(\mathbf{e}_X)\mathbf{C}^\top &= \mathbf{s}_3 - \sigma_Y(\mathbf{e}_Y)\mathbf{C}^\top. \end{aligned} \tag{5.2.6}$$

Observe that the σ_X is needed, as $\mathbf{e}_X$ has length m, but we want to multiply it to $\mathbf{A}^\top \in (\mathbb{Z}/4\mathbb{Z})^{(k_1+k_2)\times\ell}$, respectively to $\mathbf{C}^\top \in (\mathbb{Z}/2\mathbb{Z})^{(k_1+k_2)\times k_2}$. In the algorithm we will not use the embedding σ_X but rather $(\mathbb{Z}/4\mathbb{Z})^{k_1+k_2}(X)$, thus $\mathbf{e}_X$ will have length k_1+k_2, but only support in X.

In the algorithm we will define a set S that contains all vectors of the form $\sigma_X(\mathbf{e}_X)\mathbf{A}^\top$ and $\sigma_X(\mathbf{e}_X)\mathbf{C}^\top$, *i.e.*, of the left side of (5.2.6) and a set T that contains all vectors of the form $\mathbf{s}_1 - \sigma_Y(\mathbf{e}_Y)\mathbf{A}^\top$ and $\mathbf{s}_3 - \sigma_Y(\mathbf{e}_Y)\mathbf{C}^\top$, *i.e.*, of the right side of (5.2.6). Whenever a vector in S and a vector in T coincide, we call such a pair a collision.

For each collision we define $\mathbf{e}_J$ such that Condition (5.2.4) is satisfied, *i.e.*,

$$\mathbf{e}_J = \mathbf{s}_2 - \mathbf{e}_I\mathbf{B}^\top$$

and if the Lee weight of $\mathbf{e}_J$ is the remaining $t - 2v$, we have found the wanted error vector.

We will now give the algorithm of Stern in its full generality, *i.e.*, we are not restricting to the choice of I, J and Z, that we made before for illustrating the algorithm.

Algorithm 6 Stern's Algorithm over $\mathbb{Z}/4\mathbb{Z}$ in the Lee metric

Input: $\mathbf{H} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1)\times n}$, $\mathbf{s} \in (\mathbb{Z}/4\mathbb{Z})^{n-k_1}$, $t \in \mathbb{N}$,
$k_1 + k_2 = m_1 + m_2, \ell < n - k_1 - k_2$ and $v < \min\{2m_1, 2m_2, \lfloor \frac{t}{2} \rfloor\}$.
Output: $\mathbf{e} \in (\mathbb{Z}/4\mathbb{Z})^n$ with $\mathbf{eH}^\top = \mathbf{s}$ and $\mathrm{wt}_L(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size $k_1 + k_2$ and choose a zero-window $Z \subset I^C$ of size ℓ, and define $J = (I \cup Z)^C$.
2: Partition I into X of size m_1 and Y of size $m_2 = k_1 + k_2 - m_1$.
3: Compute $\mathbf{U} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1)\times(n-k_1)}$, such that

$$(\mathbf{UH})_I = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \\ 2\mathbf{C} \end{pmatrix}, \ (\mathbf{UH})_Z = \begin{pmatrix} \mathrm{Id}_\ell \\ \mathbf{0}_{(n-k_1-\ell)\times\ell} \\ \mathbf{0}_{k_2\times\ell} \end{pmatrix}, \ (\mathbf{UH})_J = \begin{pmatrix} \mathbf{0}_{\ell\times(n-k_1-k_2-\ell)} \\ \mathrm{Id}_{n-k_1-k_2-\ell} \\ \mathbf{0}_{k_2\times(n-k_1-k_2-\ell)} \end{pmatrix},$$

where $\mathbf{A} \in (\mathbb{Z}/4\mathbb{Z})^{\ell\times(k_1+k_2)}, \mathbf{B} \in (\mathbb{Z}/4\mathbb{Z})^{(n-k_1-k_2-\ell)\times(k_1+k_2)}$ and $\mathbf{C} \in (\mathbb{Z}/2\mathbb{Z})^{k_2\times(k_1+k_2)}$.
4: Compute $\mathbf{sU}^\top = \begin{pmatrix} \mathbf{s}_1 & \mathbf{s}_2 & 2\mathbf{s}_3 \end{pmatrix}$, where $\mathbf{s}_1 \in (\mathbb{Z}/4\mathbb{Z})^\ell, \mathbf{s}_2 \in (\mathbb{Z}/4\mathbb{Z})^{n-k_1-k_2-\ell}$ and $\mathbf{s}_3 \in (\mathbb{Z}/2\mathbb{Z})^{k_2}$.
5: Compute the set S

$$S = \{(\mathbf{e}_X\mathbf{A}^\top, \mathbf{e}_X\mathbf{C}^\top, \mathbf{e}_X) \mid \mathbf{e}_X \in (\mathbb{Z}/4\mathbb{Z})^{k_1+k_2}(X), \ \mathrm{wt}_L(\mathbf{e}_X) = v\}.$$

6: Compute the set T

$$T = \{(\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top, \mathbf{s}_3 - \mathbf{e}_Y\mathbf{C}^\top, \mathbf{e}_Y) \mid \mathbf{e}_Y \in (\mathbb{Z}/4\mathbb{Z})^{k_1+k_2}(Y), \ \mathrm{wt}_L(\mathbf{e}_Y) = v\}.$$

7: **for** $(\mathbf{a}, \mathbf{b}, \mathbf{e}_X) \in S$ **do**
8: **for** $(\mathbf{a}, \mathbf{b}, \mathbf{e}_Y) \in T$ **do**
9: **if** $\mathrm{wt}_L(\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top) = t - 2v$ **then**
10: Return $\mathbf{e}$, such that $\mathbf{e}_I = \mathbf{e}_X + \mathbf{e}_Y$, $\mathbf{e}_Z = \mathbf{0}_\ell$ and $\mathbf{e}_J = \mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$.
11: Start over with Step 1 and a new selection of I.

In the following, we give the complexity analysis of Algorithm 6.

Theorem 5.2.3. *Stern's algorithm over $\mathbb{Z}/4\mathbb{Z}$ equipped with the Lee metric requires on average*

$$\begin{aligned} &\binom{2m_1}{v}^{-1}\binom{2m_2}{v}^{-1}\binom{2(n-k_1-k_2-\ell)}{t-2v}^{-1}\binom{2n}{t} \\ &\quad\cdot\left(2(n-k_1)^2(n+1) + \left(\overline{L}_4(m_1, v) + \overline{L}_4(m_2, v) + \binom{2m_2}{v}\right)(2\ell + 2k_2)\right. \\ &\quad\left. + \frac{\binom{2m_1}{v}\binom{2m_2}{v}}{4^{\ell+k_2}} \min\{t - 2v + 1, n - k_1 - k_2 - \ell\} 2\min\{2v, k_1 + k_2\}\right) \end{aligned}$$

binary operations.

Proof. Let us first compute the cost of one iteration.

A broad estimate of bringing the parity-check matrix into systematic form and performing the same operations on the syndrome is given by

$$2(n-k_1)^2(n+1)$$

binary operations.

To compute the set S, we can use the technique of intermediate sums. We want to compute $\mathbf{e}_X\mathbf{A}^\top$ and $\mathbf{e}_X\mathbf{C}^\top$ for all $\mathbf{e}_X \in (\mathbb{Z}/4\mathbb{Z})^{k_1+k_2}(X)$ of Lee weight v. With intermediate sums, this costs

$$\overline{L}_4(m_1, v)(2\ell + 2k_2)$$

binary operations, where the part $\overline{L}_4(m_1, v)2\ell$ comes from computing $\mathbf{e}_X\mathbf{A}^\top$ and the part $\overline{L}_4(m_1, v)2k_2$ comes from computing $\mathbf{e}_X\mathbf{C}^\top$.

For the set T it works similarly: we want to compute $\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top$ and $\mathbf{s}_3 - \mathbf{e}_Y\mathbf{C}^\top$, for all $\mathbf{e}_Y \in (\mathbb{Z}/4\mathbb{Z})^{k_1+k_2}(Y)$ of Lee weight v. Using intermediate sums, this costs

$$\overline{L}_4(m_2, v)(2\ell + 2k_2) + \binom{2m_2}{v}(2\ell + 2k_2)$$

binary operations. The $\overline{L}_4(m_2, v)2\ell + \binom{2m_2}{v}2\ell$ part comes from computing $\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top$, whereas the $\overline{L}_4(m_2, v)2k_2 + \binom{2m_2}{v}2k_2$ part comes from computing $\mathbf{s}_3 - \mathbf{e}_Y\mathbf{C}^\top$.

The remaining steps of the algorithms are in a nested for-loop where we go through all elements in S and all elements in T, thus usually the cost of these steps should be multiplied by the size of $S \times T$. However, since the algorithm only continues if the first entry of the element in S coincides with the first entry of the element in T, thus asking for a collision, we can use instead the number of collisions we expect on average. Since In the next step, we want to check for collisions between S and T. Note that S is of size $\binom{2m_1}{v}$ and similarly T is of size $\binom{2m_2}{v}$. The resulting vectors $\mathbf{e}_X\mathbf{A}^\top$, respectively, $\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top$ live in $(\mathbb{Z}/4\mathbb{Z})^\ell$, whereas the second part of the resulting vectors $\mathbf{e}_X\mathbf{C}^\top$ and $\mathbf{s}_3 - \mathbf{e}_Y\mathbf{C}^\top$ live in $(\mathbb{Z}/4\mathbb{Z})^{k_2}$ and we assume that they are uniformly distributed. Hence, we have to check on average

$$\frac{\binom{2m_1}{v}\binom{2m_2}{v}}{4^{\ell+k_2}}$$

many collisions. For each collision we have to compute $\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$, which would usually require

$$(n - k_1 - k_2 - \ell)2\min\{2v, k_1 + k_2\}$$

binary operations. However, we can use the technique of early abort, since the algorithm only continues if the weight of $\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$ is $t - 2v$. Note that assuming the resulting vector is uniformly distributed, we have that one entry of the resulting vector adds Lee weight 1 to the weight of the full vector, with probability 1. Hence we have to compute on average $(t - 2v + 1)$ many entries of the resulting vector before we can abort. Computing one entry of the vector $\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$ costs $4v$ binary operations. Thus, by applying early abort, we get that this step costs on average

$$(t - 2v + 1)2\min\{2v, k_1 + k_2\}$$

binary operations.

The success probability is given by

$$\binom{2m_1}{v}\binom{2m_2}{v}\binom{2(n-k_1-k_2-\ell)}{t-2v}\binom{2n}{t}^{-1}.$$

Thus, we get the claimed complexity. □

5.3 Information Set Decoding over $\mathbb{Z}/p^s\mathbb{Z}$ in the Lee Metric

Let $\mathbf{H} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k_1)\times n}$ be a parity-check matrix of a linear code $\mathcal{C}$ over $(\mathbb{Z}/p^s\mathbb{Z})^n$ equipped with the Lee metric and of type

$$|\mathcal{C}| = (p^s)^{k_1}(p^{s-1})^{k_2}\cdots p^{k_s}$$

and hence having an information set of size $K = \sum_{i=1}^{s} k_i$.

Recall that the systematic form of the parity-check matrix $\mathbf{H}$ is given as

$$\mathbf{H} = \begin{pmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} & \cdots & \mathbf{B}_{1,s-1} & \mathbf{B}_{1,s} & \mathrm{Id}_{n-K} \\ p\mathbf{B}_{2,1} & p\mathbf{B}_{2,2} & \cdots & p\mathbf{B}_{2,s-1} & p\,\mathrm{Id}_{k_s} & \mathbf{0}_{k_s\times(n-K)} \\ p^2\mathbf{B}_{3,1} & p^2\mathbf{B}_{3,2} & \cdots & p^2\,\mathrm{Id}_{k_{s-1}} & \mathbf{0}_{k_{s-1}\times k_s} & \mathbf{0}_{k_{s-1}\times(n-K)} \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ p^{s-1}\mathbf{B}_{s,1} & p^{s-1}\,\mathrm{Id}_{k_2} & \cdots & \mathbf{0}_{k_2\times k_{s-1}} & \mathbf{0}_{k_2\times k_s} & \mathbf{0}_{k_2\times(n-K)} \end{pmatrix}, \tag{5.3.1}$$

where $\mathbf{B}_{1,j} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K)\times k_j}$ and $\mathbf{B}_{i,j} \in (\mathbb{Z}/p^{s+1-i}\mathbb{Z})^{k_{s+2-i}\times k_j}$ for $i > 1$.

For the following algorithms, however, we will only consider $\mathbf{H}$ to be in the following form:

$$\mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathrm{Id}_{n-K} \\ p\mathbf{B} & \mathbf{0}_{(K-k_1)\times(n-K)} \end{pmatrix}, \tag{5.3.2}$$

for $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K)\times K}$ and $\mathbf{B} \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{(K-k_1)\times K}$.

Hence we only partition into the rows of $\mathbf{H}$ which are divisible by p, and those which are not. This is exactly as the structure we have over $\mathbb{Z}/4\mathbb{Z}$, and hence we can directly adapt the algorithms from Section 5.2 to $\mathbb{Z}/p^s\mathbb{Z}$ without repeating their structure. The main difference will be the techniques that can only be applied over $\mathbb{Z}/4\mathbb{Z}$, and thus cannot be used anymore.

We will also cover the special case, where $s = 1$. Since there we are over a finite field, the structure of the algorithms are exactly the same as over $\mathbb{F}_q$ in the Hamming metric, see Chapter 4. Thus, we will not repeat the idea of these algorithms, but we will give the full algorithms and their costs, as the tools we have for the Hamming metric do not apply to the Lee metric as well.

We will then later see a generalization of the algorithms to the form (5.3.1), where we iteratively use the algorithms considering the easier form (5.3.2) and the algorithms over $\mathbb{F}_p$.

ISD algorithms over $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric were already studied in [104] using the systematic form (5.3.2). The special case of $s = 1$ was studied in [105]. Both of these papers are in collaboration with Massimo Battaglioni, Paolo Santini, Anna-Lena Horlemann-Trautmann, Edoardo Persichetti, Marco Baldi and Franco Chiaraluce.

5.3.1 Prange's algorithm

In this section we present Prange's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric. Such an algorithm was provided in [104] in collaboration with Massimo Battaglioni, Paolo Santini, Edoardo Persichetti, Marco Baldi and Franco Chiaraluce. The cost of the algorithm might differ from the one provided in the paper to match the techniques used in this chapter.

Recall that in Prange's algorithm, we assume that there exists an information set I, that is disjoint to the support of the error vector $\mathrm{Supp}(\mathbf{e})$.

The structure of the algorithm is exactly as for Prange's algorithm over $\mathbb{Z}/4\mathbb{Z}$, hence we will not repeat it and instead give directly the algorithm and its complexity.

Algorithm 7 Prange's Algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ in the Lee metric

Input: $\mathbf{H} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k_1)\times n}$, $\mathbf{s} \in (\mathbb{Z}/p^s\mathbb{Z})^{n-k_1}$, $t \in \mathbb{N}$.
Output: $\mathbf{e} \in (\mathbb{Z}/p^s\mathbb{Z})^n$ with $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$ and $\mathrm{wt}_L(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size K and define $J = I^C$.
2: Compute $\mathbf{U} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k_1)\times(n-k_1)}$, such that

$$(\mathbf{UH})_I = \begin{pmatrix} \mathbf{A} \\ p\mathbf{B} \end{pmatrix} \quad \text{and} \quad (\mathbf{UH})_J = \begin{pmatrix} \mathrm{Id}_{n-K} \\ \mathbf{0}_{(K-k_1)\times(n-K)} \end{pmatrix},$$

where $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K)\times K}, \mathbf{B} \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{(K-k_1)\times K}$.
3: Compute $\mathbf{s}\mathbf{U}^\top = \begin{pmatrix} \mathbf{s}_1 & p\mathbf{s}_2 \end{pmatrix}$, where $\mathbf{s}_1 \in (\mathbb{Z}/p^s\mathbb{Z})^{n-K}$ and $\mathbf{s}_2 \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{K-k_1}$.
4: **if** $\mathrm{wt}_L(\mathbf{s}_1) = t$ and $\mathrm{wt}_L(\mathbf{s}_2) = 0$ **then**
5: Return $\mathbf{e}$, such that $\mathbf{e}_I = \mathbf{0}_K$ and $\mathbf{e}_J = \mathbf{s}_1$.
6: Start over with Step 1 and a new selection of I.

The average complexity of Algorithm 7 can be computed as follows.

Theorem 5.3.1. *Prange's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ endowed with the Lee metric requires on average*

$$F(n-K,t,p^s)^{-1}F(n,t,p^s)(n-k_1)^2(n+1)\left(\lceil\log_2(p^s)\rceil + \lceil\log_2(p^s)\rceil^2\right)$$

binary operations.

Proof. As for the previous algorithms considering Prange's idea, we have that one iteration only consists in bringing $\mathbf{H}$ into systematic form thus a broad estimate is given by

$$(n-k_1)^2(n+1)\left(\lceil\log_2(p^s)\rceil + \lceil\log_2(p^s)\rceil^2\right)$$

binary operations.

The success probability is given by having chosen the correct weight distribution of $\mathbf{e}$. In this case, we require that no errors happen in the chosen information set, hence such a probability is given by

$$F(n-K,t,p^s)F(n,t,p^s)^{-1}.$$

Thus, we get the claim. □

The case $s=1$ Prange's algorithm over $\mathbb{F}_p$ equipped with the Lee metric was given in [105] in collaboration with Massimo Battaglioni, Paolo Santini, Anna-Lena Horlemann-Trautmann and Edoardo Persichetti.

For the special case, where $s=1$, we are over the finite field $\mathbb{F}_p$ and hence the systematic form of the parity-check matrix is given by

$$\mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathrm{Id}_{n-k} \end{pmatrix},$$

for $\mathbf{A} \in \mathbb{F}_p^{(n-k)\times k}$.

The structure of the algorithm is exactly as over $\mathbb{F}_p$ equipped with the Hamming metric, we hence refer to Section 4.1.

Algorithm 8 Prange's Algorithm over $\mathbb{F}_p$ in the Lee metric

Input: $\mathbf{H} \in \mathbb{F}_p^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_p^{n-k}$, $t \in \mathbb{N}$.
Output: $\mathbf{e} \in \mathbb{F}_p^n$ with $\mathbf{eH}^\top = \mathbf{s}$ and $\mathrm{wt}_L(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1,...,n\}$ of size k and define $J = I^C$.
2: Compute $\mathbf{U} \in \mathbb{F}_p^{(n-k)\times(n-k)}$, such that

$$(\mathbf{UH})_I = \mathbf{A} \text{ and } (\mathbf{UH})_J = \mathrm{Id}_{n-k},$$

where $\mathbf{A} \in \mathbb{F}_p^{(n-k)\times k}$.
3: Compute $\mathbf{s}' = \mathbf{sU}^\top$.
4: **if** $\mathrm{wt}_L(\mathbf{s}') = t$ **then**
5: Return $\mathbf{e}$, such that $\mathbf{e}_I = \mathbf{0}_k$ and $\mathbf{e}_J = \mathbf{s}'$.
6: Start over with Step 1 and a new selection of I.

The average complexity of Prange's algorithm over $\mathbb{F}_p$ endowed with the Lee metric is as follows.

Theorem 5.3.2. *Prange's algorithm over $\mathbb{F}_p$ equipped with the Lee metric requires on average*

$$F(n-k,t,p)^{-1}F(n,t,p)(n-k)^2(n+1)\left(\lceil \log_2(p)\rceil + \lceil \log_2(p)\rceil^2\right)$$

binary operations.

Proof. Bringing the parity-check matrix into systematic form and applying the same operations on the syndrome costs

$$(n-k)^2(n+1)\left(\lceil \log_2(p)\rceil + \lceil \log_2(p)\rceil^2\right)$$

binary operations.

The success probability is given by

$$F(n-k,t,p)F(n,t,p)^{-1}.$$

Hence, we get the claim. □

5.3.2 Lee-Brickell's algorithm

In this section we want to formulate the algorithm of Lee-Brickell in $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric. The structure and idea of Lee-Brickell's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ in the Lee metric was already given in [105] in collaboration with Massimo Battaglioni, Paolo Santini, Anna-Lena Horlemann-Trautmann and Edoardo Persichetti.

For this recall that in Lee-Brickell's algorithm we assume that there exists an information set such that v errors happen inside the information set and $t-v$ outside.

Since, the structure of the algorithm is exactly as for Lee-Brickell's algorithm over $\mathbb{Z}/4\mathbb{Z}$, we refer for an illustration to Section 5.2.2.

Algorithm 9 Lee-Brickell's Algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ in the Lee metric

Input: $\mathbf{H} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k_1)\times n}$, $\mathbf{s} \in (\mathbb{Z}/p^s\mathbb{Z})^{n-k_1}$, $t \in \mathbb{N}$, $v < \min\{K\lfloor \frac{p^s-1}{2}\rfloor, t\}$.
Output: $\mathbf{e} \in (\mathbb{Z}/p^s\mathbb{Z})^n$ with $\mathbf{eH}^\top = \mathbf{s}$ and $\mathrm{wt}_L(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1,...,n\}$ of size K and define $J = \{1,\dots,n\} \setminus I$.
2: Compute $\mathbf{U} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k_1)\times(n-k_1)}$, such that

$$(\mathbf{UH})_I = \begin{pmatrix} \mathbf{A} \\ p\mathbf{B} \end{pmatrix} \quad \text{and } (\mathbf{UH})_J = \begin{pmatrix} \mathrm{Id}_{n-K} \\ \mathbf{0}_{(K-k_1)\times(n-K)} \end{pmatrix},$$

where $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K)\times K}$ and $\mathbf{B} \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{(K-k_1)\times K}$.
3: Compute $\mathbf{sU}^\top = \begin{pmatrix} \mathbf{s}_1 & p\mathbf{s}_2 \end{pmatrix}$, where $\mathbf{s}_1 \in (\mathbb{Z}/p^s\mathbb{Z})^{n-K}$ and $\mathbf{s}_2 \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{K-k_1}$.
4: **for** $\mathbf{e}_I \in (\mathbb{Z}/p^s\mathbb{Z})^K$ with $\mathrm{wt}_L(\mathbf{e}_I) = v$ **do**
5: **if** $\mathbf{e}_I\mathbf{B}^\top = \mathbf{s}_2$ **then**
6: **if** $\mathrm{wt}_L(\mathbf{s}_1 - \mathbf{e}_I\mathbf{A}^\top) = t - v$ **then**
7: Return $\mathbf{e}$, such that $\mathbf{e}_I = \mathbf{e}_I$ and $\mathbf{e}_J = \mathbf{s}_1 - \mathbf{e}_I\mathbf{A}^\top$.
8: Start over with Step 1 and a new selection of I.

We now provide a complexity estimate of Lee-Brickell's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric.

Theorem 5.3.3. *Lee-Brickell's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ requires on average*

$$\begin{aligned}&F(K,v,p^s)^{-1}F(n-K,t-v,p^s)^{-1}F(n,t,p^s)\left((n-k_1)^2(n+1)\right.\\&+F(K,v,p^s)\min\{v,K\}(K-k_1)\\&\left.+F(K,v,p^s)\min\{\mu_{p^s}^{-1}(t-v+1),n-K\}\min\{v,K\}\right)\\&\cdot\left(\lceil\log_2(p^s)\rceil+\lceil\log_2(p^s)\rceil^2\right)\end{aligned}$$

binary operations.

Proof. The first step is to bring the parity-check matrix into systematic form, which is given by

$$(n-k_1)^2(n+1)\left(\lceil\log_2(p^s)\rceil+\lceil\log_2(p^s)\rceil^2\right)$$

binary operations.

In the next step, we go through all $\mathbf{e}_I \in (\mathbb{Z}/p^s\mathbb{Z})^K$ of Lee weight v, and check if $\mathbf{e}_I\mathbf{B}^\top = \mathbf{s}_2$. This requires

$$\min\{v,K\}(K-k_1)\left(\lceil\log_2(p^s)\rceil+\lceil\log_2(p^s)\rceil^2\right)$$

binary operations for each $\mathbf{e}_I$. Hence, in total this step costs

$$F(K,v,p^s)\min\{v,K\}(K-k_1)\left(\lceil\log_2(p^s)\rceil+\lceil\log_2(p^s)\rceil^2\right)$$

binary operations. Note that here we have to use the full cost, as there is no known concept of intermediate sums for the Lee metric over $\mathbb{Z}/p^s\mathbb{Z}$.

Finally, we compute $\mathbf{s}_1-\mathbf{e}_I\mathbf{A}^\top$, for all $\mathbf{e}_I \in (\mathbb{Z}/p^s\mathbb{Z})^K$ of Lee weight v. This usually would require

$$F(K,v,p^s)\min\{v,K\}(n-K)\left(\lceil\log_2(p^s)\rceil+\lceil\log_2(p^s)\rceil^2\right)$$

binary operations, since $\mathbf{e}_I$ has support size of at most $\min\{v,K\}$. However, the algorithm only continues if the Lee weight of $\mathbf{s}_1-\mathbf{e}_I\mathbf{A}^\top$ is $t-v$. Assuming that the resulting vector is uniformly distributed, we have that a random element has average Lee weight μ_{p^s}. Hence we have to compute on average $\mu_{p^s}^{-1}(t-v+1)$ many entries of the resulting vector before we can abort. Since computing one entry of the vector $\mathbf{s}_1-\mathbf{e}_I\mathbf{A}^\top$ costs $\min\{v,K\}\left(\lceil\log_2(p^s)\rceil+\lceil\log_2(p^s)\rceil^2\right)$ binary operations, by applying early abort, we get that this step costs on average

$$F(K,v,p^s)\mu_{p^s}^{-1}(t-v+1)\min\{v,K\}\left(\lceil\log_2(p^s)\rceil+\lceil\log_2(p^s)\rceil^2\right)$$

binary operations.

The success probability is given by having chosen the correct weight distribution of $\mathbf{e}$, which is given by

$$F(K,v,p^s)F(n-K,t-v,p^s)F(n,t,p^s)^{-1}.$$

By multiplying the average number of iterations needed to the cost of one iteration, we get the claimed complexity. □

The case $s = 1$ Such an algorithm was provided in [105] in collaboration with Massimo Battaglioni, Paolo Santini, Anna-Lena Horlemann-Trautmann and Edoardo Persichetti.

We also want to consider the special case where we are over a finite field $\mathbb{F}_p$. In this case the systematic form of the parity-check matrix is easier and the structure of the algorithm is exactly as for Lee-Brickell's algorithm over $\mathbb{F}_p$ in the Hamming metric, hence we refer to Section 4.2.

Algorithm 10 Lee-Brickell's Algorithm over $\mathbb{F}_p$ in the Lee metric

Input: $\mathbf{H} \in \mathbb{F}_p^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_p^{n-k}$, $t \in \mathbb{N}$, $v < \min\{t, \frac{p-1}{2}k\}$.
Output: $\mathbf{e} \in \mathbb{F}_p^n$ with $\mathbf{eH}^\top = \mathbf{s}$ and $\mathrm{wt}_L(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size k and define $J = I^C$.
2: Compute $\mathbf{U} \in \mathbb{F}_p^{(n-k)\times(n-k)}$, such that

$$(\mathbf{UH})_I = \mathbf{A} \quad \text{and } (\mathbf{UH})_J = \mathrm{Id}_{n-k},$$

where $\mathbf{A} \in \mathbb{F}_p^{(n-k)\times k}$.
3: Compute $\mathbf{s}' = \mathbf{sU}^\top$.
4: **for** $\mathbf{e}_I \in \mathbb{F}_p^k$ with $\mathrm{wt}_L(\mathbf{e}_I) = v$ **do**
5: **if** $\mathrm{wt}_L(\mathbf{s}' - \mathbf{e}_I\mathbf{A}^\top) = t - v$ **then**
6: Return $\mathbf{e}$, such that $\mathbf{e}_I = \mathbf{e}_I$ and $\mathbf{e}_J = \mathbf{s}' - \mathbf{e}_I A^\top$.
7: Start over with Step 1 and a new selection of I.

The average complexity of Lee-Brickell's algorithm over $\mathbb{F}_p$ endowed with the Lee metric is given as follows.

Theorem 5.3.4. *Lee-Brickell's algorithm over $\mathbb{F}_p$ equipped with the Lee metric requires on average*

$$F(k,v,p)^{-1}F(n-k,t-v,p)^{-1}F(n,t,p)\left((n-k)^2(n+1)\right.$$
$$\left. + F(k,v,p)\min\{\mu_p^{-1}(t-v+1), n-k\}\min\{v,k\}\right)\left(\lceil\log_2(p)\rceil + \lceil\log_2(p)\rceil^2\right)$$

binary operations.

Proof. Bringing the parity-check matrix into systematic form has a broad cost estimate of

$$(n-k)^2(n+1)\left(\lceil\log_2(p)\rceil + \lceil\log_2(p)\rceil^2\right)$$

binary operations.

In the next step, we go through all $\mathbf{e}_I \in \mathbb{F}_p^k$ of Lee weight v, and compute $\mathbf{s}' - \mathbf{e}_I\mathbf{A}^\top$. This would usually require

$$F(k,v,p)(n-k)\min\{v,k\}\left(\lceil\log_2(p)\rceil + \lceil\log_2(p)\rceil^2\right)$$

binary operations, since $\mathbf{e}_I$ has support of size at most $\min\{v, k\}$. However, using the concept of early abort, this step requires on average

$$F(k, v, p)\mu_p^{-1}(t - v + 1)\min\{v, k\}\left(\lceil \log_2(p)\rceil + \lceil \log_2(p)\rceil^2\right)$$

binary operations.

Since we require that v errors happen in the chosen information set and the remaining $t - v$ outside, the success probability is given by

$$F(k, v, p)F(n - k, t - v, p)F(n, t, p)^{-1}.$$

Then, the estimated overall cost of Lee-Brickell's ISD algorithm over $\mathbb{F}_p$ equipped with the Lee metric is given as in the claim. □

5.3.3 Stern's algorithm

In this section we formulate Stern's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric. Such an algorithm was provided in [104] in collaboration with Massimo Battaglioni, Paolo Santini, Marco Baldi, Franco Chiaraluce and Edoardo Persichetti. For consistency reasons, the cost of the algorithm provided in this section might differ from the one provided in [104].

Recall that in Stern's algorithm we partition the information set into two sets. We assume that v errors happen in both of the sets and that there is a zero-window of size ℓ outside of the information set where no errors happen.

Since the structure of Stern's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$, when only considering $\mathbf{H}$ having the form (5.3.2) is exactly as over $\mathbb{Z}/4\mathbb{Z}$. Hence we refer for the structure to Section 5.2.3.

In the following we provide a complexity estimate of Stern's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric.

Theorem 5.3.5. *Stern's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric requires on average*

$$\begin{aligned}
&F(m_1, v, p^s)^{-1}F(m_2, v, p^s)^{-1}F(n - K - \ell, t - 2v, p^s)^{-1}F(n, t, p^s)\\
&\cdot\left(\lceil \log_2(p^s)\rceil + \lceil \log_2(p^s)\rceil^2\right)\big((n - k_1)^2(n + 1)\\
&+(K - k_1 + \ell)(F(m_1, v, p^s)\min\{v, m_1\} + F(m_2, v, p^s)\min\{v, m_2\})\\
&+\frac{F(m_1, v, p^s)F(m_2, v, p^s)}{(p^s)^{\ell+K-k_1}}\min\{\mu_{p^s}^{-1}(t - 2v + 1), n - K - \ell\}\min\{2v, K\}\Big)
\end{aligned}$$

binary operations.

Proof. We first compute the cost of one iteration. As usual, bringing the parity-check matrix into systematic form has a broad cost estimate of

$$(n - k_1)^2(n + 1)\left(\lceil \log_2(p^s)\rceil + \lceil \log_2(p^s)\rceil^2\right)$$

Algorithm 11 Stern's Algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ in the Lee metric

Input: $\mathbf{H} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k_1)\times n}$, $\mathbf{s} \in (\mathbb{Z}/p^s\mathbb{Z})^{n-k_1}$, $t \in \mathbb{N}$, $K = m_1 + m_2, \ell < n - K$ and $v < \min\{\lfloor \frac{p^s-1}{2} \rfloor m_1, \lfloor \frac{p^s-1}{2} \rfloor m_2, \lfloor \frac{t}{2} \rfloor\}$.
Output: $\mathbf{e} \in (\mathbb{Z}/p^s\mathbb{Z})^n$ with $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$ and $\mathrm{wt}_L(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size K and choose a zero-window $Z \subset I^C$ of size ℓ, and define $J = (I \cup Z)^C$.
2: Partition I into X of size m_1 and Y of size $m_2 = K - m_1$.
3: Compute $\mathbf{U} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k_1)\times(n-k_1)}$, such that

$$(\mathbf{UH})_I = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \\ p\mathbf{C} \end{pmatrix}, \ (\mathbf{UH})_Z = \begin{pmatrix} \mathrm{Id}_\ell \\ \mathbf{0}_{(n-K-\ell)\times\ell} \\ \mathbf{0}_{(K-k_1)\times\ell} \end{pmatrix}, \ (\mathbf{UH})_J = \begin{pmatrix} \mathbf{0}_{\ell\times(n-K-\ell)} \\ \mathrm{Id}_{n-K-\ell} \\ \mathbf{0}_{(K-k_1)\times(n-K-\ell)} \end{pmatrix},$$

where $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell\times K}, \mathbf{B} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K-\ell)\times K}$ and $\mathbf{C} \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{(K-k_1)\times K}$.
4: Compute $\mathbf{s}\mathbf{U}^\top = \begin{pmatrix} \mathbf{s}_1 & \mathbf{s}_2 & p\mathbf{s}_3 \end{pmatrix}$, where $\mathbf{s}_1 \in (\mathbb{Z}/p^s\mathbb{Z})^\ell, \mathbf{s}_2 \in (\mathbb{Z}/p^s\mathbb{Z})^{n-K-\ell}$ and $\mathbf{s}_3 \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{K-k_1}$.
5: Compute the set S

$$S = \{(\mathbf{e}_X\mathbf{A}^\top, \mathbf{e}_X\mathbf{C}^\top, \mathbf{e}_X) \mid \mathbf{e}_X \in (\mathbb{Z}/p^s\mathbb{Z})^K(X), \ \mathrm{wt}_L(\mathbf{e}_X) = v\}.$$

6: Compute the set T

$$T = \{(\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top, \mathbf{s}_3 - \mathbf{e}_Y\mathbf{C}^\top, \mathbf{e}_Y) \mid \mathbf{e}_Y \in (\mathbb{Z}/p^s\mathbb{Z})^K(Y), \ \mathrm{wt}_L(\mathbf{e}_Y) = v\}.$$

7: **for** $(\mathbf{a}, \mathbf{b}, \mathbf{e}_X) \in S$ **do**
8: **for** $(\mathbf{a}, \mathbf{b}, \mathbf{e}_Y) \in T$ **do**
9: **if** $\mathrm{wt}_L(\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top) = t - 2v$ **then**
10: Return $\mathbf{e}$, such that $\mathbf{e}_I = \mathbf{e}_X + \mathbf{e}_Y$, $\mathbf{e}_Z = \mathbf{0}_\ell$ and $\mathbf{e}_J = \mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$.
11: Start over with Step 1 and a new selection of I.

binary operations.

For the set S we want to compute $\mathbf{e}_X\mathbf{A}^\top$ and $\mathbf{e}_X\mathbf{C}^\top$ for all $\mathbf{e}_X$ in $(\mathbb{Z}/p^s\mathbb{Z})^K(X)$ of Lee weight v. This step requires

$$F(m_1, v, p^s)(K - k_1 + \ell)\min\{v, m_1\}\left(\lceil \log_2(p^s) \rceil + \lceil \log_2(p^s) \rceil^2\right)$$

binary operations, where the part

$$\min\{v, m_1\}\ell\left(\lceil \log_2(p^s) \rceil + \lceil \log_2(p^s) \rceil^2\right)$$

comes from computing $\mathbf{e}_X\mathbf{A}^\top$ and the part

$$(K - k_1)\min\{v, m_1\}\left(\lceil \log_2(p^s) \rceil + \lceil \log_2(p^s) \rceil^2\right)$$

comes from computing $\mathbf{e}_X\mathbf{C}^\top$.

Similarly, we can build set T: we want to compute $\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top$ and $\mathbf{s}_3 - \mathbf{e}_Y\mathbf{C}^\top$, for all $\mathbf{e}_Y \in (\mathbb{Z}/p^s\mathbb{Z})^K(Y)$ of Lee weight v. This costs

$$F(m_2, v, p^s)(K - k_1 + \ell)\min\{v, m_2\}\left(\lceil\log_2(p^s)\rceil + \lceil\log_2(p^s)\rceil^2\right)$$

binary operations, where the

$$\ell\min\{v, m_2\}\left(\lceil\log_2(p^s)\rceil + \lceil\log_2(p^s)\rceil^2\right)$$

part comes from computing $\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top$, whereas the

$$(K - k_1)\min\{v, m_2\}\left(\lceil\log_2(p^s)\rceil + \lceil\log_2(p^s)\rceil^2\right)$$

part comes from computing $\mathbf{s}_3 - \mathbf{e}_Y\mathbf{C}^\top$.

In the next step, we want to check for collisions between S and T. Since S consists of all $\mathbf{e}_X \in (\mathbb{Z}/p^s\mathbb{Z})^K(X)$ of Lee weight v, S is of size $F(m_1, v, p^s)$ and similarly T is of size $F(m_2, v, p^s)$. The resulting vectors $\mathbf{e}_X\mathbf{A}^\top$, respectively, $\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top$ live in $(\mathbb{Z}/p^s\mathbb{Z})^\ell$, whereas the second part of the resulting vectors $\mathbf{e}_X\mathbf{C}^\top$ and $\mathbf{s}_3 - \mathbf{e}_Y\mathbf{C}^\top$ live in $(\mathbb{Z}/p^s\mathbb{Z})^{(K-k_1)}$ and we assume that they are uniformly distributed. Hence we have to check on average

$$\frac{F(m_1, v, p^s)F(m_2, v, p^s)}{(p^s)^{\ell+K-k_1}}$$

many collisions. For each collision we have to compute $\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$, which would usually require

$$(n - K - \ell)\min\{2v, K\}\left(\lceil\log_2(p^s)\rceil + \lceil\log_2(p^s)\rceil^2\right)$$

binary operations. However, using the concept of early abort, this step requires on average

$$\mu_{p^s}^{-1}(t - 2v + 1)\min\{2v, K\}\left(\lceil\log_2(p^s)\rceil + \lceil\log_2(p^s)\rceil^2\right)$$

binary operations.

The success probability is given by

$$F(m_1, v, p^s)F(m_2, v, p^s)F(n - K - \ell, t - 2v, p^s)F(n, t, p^s)^{-1}.$$

Multiplying the reciprocal of this, *i.e.*, the number of iterations needed on average, to the cost of one iteration, we get the claim. □

The case $s = 1$ Stern's algorithm over $\mathbb{F}_p$ equipped with the Lee metric was provided in [105] in collaboration with Massimo Battaglioni, Paolo Santini, Anna-Lena Horlemann-Trautmann and Edoardo Persichetti.

As for the previous algorithms, we want to consider also for Stern's algorithm the special case where $s = 1$ and hence we are over a finite field $\mathbb{F}_p$. Since we are over a finite

Algorithm 12 Stern's Algorithm over $\mathbb{F}_p$ in the Lee metric

Input: $\mathbf{H} \in \mathbb{F}_p^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_p^{n-k}$, $t \in \mathbb{N}$, $k = m_1 + m_2, \ell < n-k$ and $v < \min\{\frac{p-1}{2}m_1, \frac{p-1}{2}m_2, \lfloor \frac{t}{2} \rfloor\}$.
Output: $\mathbf{e} \in \mathbb{F}_p^n$ with $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$ and $\mathrm{wt}_L(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size k and choose a zero-winow $Z \subset I^C$ of size ℓ, and define $J = (I \cup Z)^C$.
2: Partition I into X of size m_1 and Y of size $m_2 = k - m_1$.
3: Compute $\mathbf{U} \in \mathbb{F}_p^{(n-k)\times(n-k)}$, such that

$$(\mathbf{UH})_I = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix}, \ (\mathbf{UH})_Z = \begin{pmatrix} \mathrm{Id}_\ell \\ \mathbf{0}_{(n-k-\ell)\times\ell} \end{pmatrix} \text{ and } (\mathbf{UH})_J = \begin{pmatrix} \mathbf{0}_{\ell\times(n-k-\ell)} \\ \mathrm{Id}_{n-k-\ell} \end{pmatrix},$$

where $\mathbf{A} \in \mathbb{F}_p^{\ell\times k}$ and $\mathbf{B} \in \mathbb{F}_p^{(n-k-\ell)\times k}$.
4: Compute $\mathbf{s}\mathbf{U}^\top = \begin{pmatrix} \mathbf{s}_1 & \mathbf{s}_2 \end{pmatrix}$, where $\mathbf{s}_1 \in \mathbb{F}_p^\ell$ and $\mathbf{s}_q \in \mathbb{F}_p^{n-k-\ell}$.
5: Compute the set S

$$S = \{(\mathbf{e}_X\mathbf{A}^\top, \mathbf{e}_X) \mid \mathbf{e}_X \in \mathbb{F}_p^k(X), \ \mathrm{wt}_L(\mathbf{e}_X) = v\}.$$

6: Compute the set T

$$T = \{(\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top, \mathbf{e}_Y) \mid \mathbf{e}_Y \in \mathbb{F}_p^k(Y), \ \mathrm{wt}_L(\mathbf{e}_Y) = v\}.$$

7: **for** $(\mathbf{a}, \mathbf{e}_X) \in S$ **do**
8: **for** $(\mathbf{a}, \mathbf{e}_Y) \in T$ **do**
9: **if** $\mathrm{wt}_L(\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top) = t - 2v$ **then**
10: Return $\mathbf{e}$, such that $\mathbf{e}_I = \mathbf{e}_X + \mathbf{e}_Y$, $\mathbf{e}_Z = \mathbf{0}_\ell$ and $\mathbf{e}_J = \mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$.
11: Start over with Step 1 and a new selection of I.

field, we have the classical systematic form for the parity-check matrix. The structure of the algorithm will hence be exactly as for Stern's algorithm over $\mathbb{F}_p$ equipped with the Hamming metric. For an illustration of the idea of the algorithm we hence refer to Section 4.3.

The average complexity of Stern's algorithm over $\mathbb{F}_p$ equipped with the Lee metric is as follows.

Theorem 5.3.6. *Stern's algorithm over $\mathbb{F}_p$ equipped with the Lee metric requires on average*

$$\begin{aligned} &F(m_1, v, p)^{-1}F(m_2, v, p)^{-1}F(n-k-\ell, t-2v, p)^{-1}F(n, t, p) \\ &\cdot \left(\lceil \log_2(p) \rceil + \lceil \log_2(p) \rceil^2\right)\left((n-k)^2(n+1)\right. \\ &+ \ell(F(m_1, v, p)\min\{m_1, v\} + F(m_2, v, p)\min\{m_2, v\}) \\ &\left. + \frac{F(m_1, v, p)F(m_2, v, p)}{p^\ell}\min\{\mu_p^{-1}(t-2v+1), n-k-\ell\}\min\{k, 2v\}\right) \end{aligned}$$

binary operations.

Proof. As a first step, we bring $\mathbf{H}$ into systematic form and perform the same operations on $\mathbf{s}$. This cost can be estimated by

$$(n-k)^2(n+1)\left(\lceil \log_2(p) \rceil + \lceil \log_2(p) \rceil^2\right)$$

binary operations.

In the next step, we compute the set S, *i.e.*, we want to compute $\mathbf{e}_X\mathbf{A}^\top$ for all $\mathbf{e}_X \in \mathbb{F}_p^k(X)$ of Lee weight v. This costs

$$F(m_1, v, p)\ell \min\{m_1, v\}\left(\lceil \log_2(p) \rceil^2 + \lceil \log_2(p) \rceil\right)$$

binary operations.

Similarly, we can build set T, by computing $\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top$, for all $\mathbf{e}_Y \in \mathbb{F}_p^k(Y)$ of Lee weight v. This costs

$$F(m_2, v, p)\ell \min\{m_2, v\}\left(\lceil \log_2(p) \rceil + \lceil \log_2(p) \rceil^2\right)$$

binary operations.

As a last step, we want to check for collisions between S and T. Note, that S is of size $F(m_1, v, p)$ and similarly T is of size $F(m_2, v, p)$. The resulting vectors $\mathbf{e}_X\mathbf{A}^\top$, respectively, $\mathbf{s}_1 - \mathbf{e}_Y\mathbf{A}^\top$ live in $\mathbb{F}_p^\ell$, and we assume that they are uniformly distributed. Hence, on average we have to check

$$\frac{F(m_1, v, p)F(m_2, v, p)}{p^\ell}$$

many collisions. For each collision we have to compute $\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$, which would usually require

$$(n-k-\ell)\min\{k, 2v\}\left(\lceil \log_2(p) \rceil^2 + \lceil \log_2(p) \rceil\right)$$

binary operations. However, we can use the concept of early abort, as the algorithm only proceeds if the Lee weight of $\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$ is $t - 2v$. Note that computing one entry of the vector $\mathbf{s}_2 - (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$ costs

$$\min\{k, 2v\}\left(\lceil \log_2(p) \rceil^2 + \lceil \log_2(p) \rceil\right)$$

binary operations. Therefore, we get that this step costs on average

$$\mu_p^{-1}(t - 2v + 1)\min\{k, 2v\}\left(\lceil \log_2(p) \rceil^2 + \lceil \log_2(p) \rceil\right)$$

binary operations.

Since we require that v errors happen in the set X and v in the set Y and the remaining $t - 2v$ outside the information set and the zero-window, we get that the success probability is given by

$$F(m_1, v, p)F(m_2, v, p)F(n-k-\ell, t-2v, p)F(n, t, p)^{-1}.$$

□

5.3.4 Generalization

Even though the algorithms from Section 5.3 work over $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric, they are not taking into account the special structure of the parity-check matrix over the finite ring. In this approach we only consider the rows of $\mathbf{H}$ that are not divisible by p and the rows which are, *i.e.*, the code parameter k_1. Considering all the code parameters k_i, for $i \in \{1, \dots, s\}$ would give an overly complex algorithm. However, for generalizing the algorithms, we will see that the case where we only consider the special structure of $\mathbf{H}$ together with the special case over $\mathbb{F}_p$ are enough.

Note that this iterative algorithm was presented in [105], though without an example and a complexity analysis, which we will provide in this section.

For the generalization we need a condition involving p, thus we can only generalize Lee-Brickell's and Stern's algorithm.

In Lee-Brickell's algorithm and in Stern's algorithm we have the condition $p\mathbf{e}_I\mathbf{B}^\top = p\mathbf{s}_2$, respectively $p\mathbf{e}_I\mathbf{C}^\top = p\mathbf{s}_3$. This equation can be reduced to $\mathbf{e}_I\mathbf{B}^\top = \mathbf{s}_2$, respectively $\mathbf{e}_I\mathbf{C}^\top = \mathbf{s}_3$, over $\mathbb{Z}_{p^{s-1}}$ which is again like the initial syndrome decoding problem but over a smaller finite ring, *i.e.*, $\mathbb{Z}_{p^{s-1}}$, and with a smaller matrix size. Thus, an improvement on the cost of the algorithms could be to iteratively reduce the size of the problem to the smallest instance and then update the solution with this partial solution.

Even though the underlying idea is simple, the algorithm is not. In the following, we will only focus on Lee-Brickell's algorithm but it is possible to adapt also Stern's algorithm accordingly.

The iterative algorithm is best seen as a recursive function.

Let $f(\mathbf{M}, \mathbf{s}, s, w)$ be the following function: first bring $\mathbf{M}$ over $\mathbb{Z}_{p^s}$ into the form

$$\mathbf{UM} = \begin{pmatrix} \mathbf{A} & \mathrm{Id} \\ p\mathbf{B} & \mathbf{0} \end{pmatrix}$$

and

$$\mathbf{sU}^\top = \begin{pmatrix} \mathbf{s}_1 & p\mathbf{s}_2 \end{pmatrix}.$$

Then, the function goes one step further and already checks whether for some invertible $\mathbf{V}$, it holds that $\mathbf{VB}$ is of the form $\begin{pmatrix} \mathbf{A}' & \mathrm{Id} \end{pmatrix}$, for some $\mathbf{A}'$. If this is the case, the function solves

$$\mathbf{e}_1\mathbf{B}^\top\mathbf{V}^\top = \begin{pmatrix} \mathbf{e}'_1 & \mathbf{e}'_2 \end{pmatrix} \begin{pmatrix} (\mathbf{A}')^\top \\ \mathrm{Id} \end{pmatrix} = \mathbf{s}' = \mathbf{s}_2\mathbf{V}^\top.$$

Then, it saves $\mathbf{e}_1^{(s-1)} = (\mathbf{e}'_1, \mathbf{e}'_2)$ as a solution.

If $\mathbf{VB}$ is not of this form, then $\mathbf{e}_1^{(s-1)} = f(\mathbf{B}, \mathbf{s}_2, s-1, w)$.

As last step (in both of the cases) it solves

$$\mathbf{eM}^\top\mathbf{U}^\top = \begin{pmatrix} \mathbf{e}_1^{(s-1)} + \tilde{\mathbf{e}}_1 p^{s-1} & \mathbf{e}_2 \end{pmatrix} \begin{pmatrix} \mathbf{A}^\top & p\mathbf{B}^\top \\ \mathrm{Id} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{s}_1 & p\mathbf{s}_2 \end{pmatrix} = \mathbf{sU}^\top,$$

using the previous solution $\mathbf{e}_1^{(s-1)}$.

Note that in this very rudimentary algorithm we partition $\mathbf{e}$ according to the columns of $\mathbf{H}$ being divisible by p^i and assume that each part of the error vector has Lee weight w.

Algorithm 13 Iterative Lee-Brickell over $\mathbb{Z}_{p^s}$ in the Lee metric

Input: The matrix $\mathbf{M} \in \mathbb{Z}_{p^s}^{a \times b}$, the vector $\mathbf{s} \in \mathbb{Z}_{p^s}^{a}$ and the positive integers s, t and w.
Output: $\mathbf{e} \in \mathbb{Z}_{p^s}^{b}$ with $\mathbf{e}\mathbf{M}^\top = \mathbf{s}$ and $\mathrm{wt}_L(\mathbf{e}) = t$.

1: Find $\mathbf{U} \in \mathbb{Z}_{p^s}^{a \times a}$ an invertible matrix, such that J is the largest set (with $\mid J \mid = c$ and $K = J^C$) satisfying $\mathbf{U}\mathbf{M}_K = \begin{pmatrix} \mathbf{A} \\ p\mathbf{B} \end{pmatrix}$ and $\mathbf{U}\mathbf{M}_J = \begin{pmatrix} \mathrm{Id}_c \\ \mathbf{0} \end{pmatrix}$.
2: Compute $\mathbf{s}\mathbf{U}^\top = \begin{pmatrix} \mathbf{s}_1 & p\mathbf{s}_2 \end{pmatrix}$.
3: Find $\mathbf{V} \in \mathbb{Z}_{p^s}^{(a-c)\times(a-c)}$ an invertible matrix, such that J' is the largest set with $\mid J' \mid = c'$ and $K' = (J')^C$ satisfying $\mathbf{V}\mathbf{B}_{K'} = \begin{pmatrix} \mathbf{A}' \\ p\mathbf{B}' \end{pmatrix}$ and $\mathbf{V}\mathbf{B}_{J'} = \begin{pmatrix} \mathrm{Id}_{c'} \\ \mathbf{0} \end{pmatrix}$.
4: **if** $\mathbf{B}' = \mathbf{0}$ **then**
5: Compute $\mathbf{s_2}\mathbf{V}^\top = \mathbf{s}'$.
6: **for** $\mathbf{e}'_1 \in \mathbb{Z}_{p^{s-1}}^{b-c-c'}$ with $\mathrm{wt}_L(\mathbf{e}'_1) = w$ **do**
7: **if** $\mathrm{wt}_L(\mathbf{s}' - \mathbf{e}'_1(\mathbf{A}')^\top) = w$ **then**
8: $(\mathbf{e}_1^{(s-1)})_{K'} = \mathbf{e}'_1$ and $(\mathbf{e}_1^{(s-1)})_{J'} = \mathbf{s}' - \mathbf{e}'_1(\mathbf{A}')^\top$
9: **else**
10: $\mathbf{e}_1^{(s-1)}$ = output of Algorithm 13 on inputs $\mathbf{B}, \mathbf{s}_2, s-1, t, w$
11: **for** $\tilde{\mathbf{e}} \in \mathbb{Z}_p^{b-c}$ **do**
12: **if** $\mathrm{wt}_L(\mathbf{s}_1 - (\mathbf{e}_1^{(s-1)} + \tilde{\mathbf{e}}p^{s-1})\mathbf{A}^\top) = w$ **then**
13: Return $(\mathbf{e}_1)_K = \mathbf{e}_1^{(s-1)} + \tilde{\mathbf{e}}p^{s-1}$ and $(\mathbf{e}_1)_J = \mathbf{s}_1 - (\mathbf{e}_1^{(s-1)} + \tilde{\mathbf{e}}p^{s-1})\mathbf{A}^\top$

This might look confusing, hence we want to give here a toy example, for $s = 3$.

Let $\mathcal{C}$ be a linear code equipped with the Lee metric over $\mathbb{Z}/p^3\mathbb{Z}$, of length n and type

$$\mid \mathcal{C} \mid = \left(p^3\right)^{k_1} \left(p^2\right)^{k_2} p^{k_3}.$$

Let us denote $K = k_1 + k_2 + k_3$.

We have that the systematic form of the parity-check matrix $\mathbf{H} \in (\mathbb{Z}/p^3\mathbb{Z})^{(n-k_1)\times n}$ is given by

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathrm{Id}_{n-K} \\ p\mathbf{D} & p\mathbf{E} & p\,\mathrm{Id}_{k_3} & \mathbf{0}_{k_3\times(n-K)} \\ p^2\mathbf{F} & p^2\,\mathrm{Id}_{k_2} & \mathbf{0}_{k_2\times k_3} & \mathbf{0}_{k_2\times(n-K)} \end{pmatrix},$$

where $\mathbf{A} \in (\mathbb{Z}/p^3\mathbb{Z})^{(n-K)\times k_1}, \mathbf{B} \in (\mathbb{Z}/p^3\mathbb{Z})^{(n-K)\times k_2}, \mathbf{C} \in (\mathbb{Z}/p^3\mathbb{Z})^{(n-K)\times k_3}$, $\mathbf{D} \in (\mathbb{Z}/p^2\mathbb{Z})^{k_3\times k_1}, \mathbf{E} \in (\mathbb{Z}/p^2\mathbb{Z})^{k_3\times k_2}$ and $\mathbf{F} \in (\mathbb{Z}/p\mathbb{Z})^{k_2\times k_1}$.

Thus, splitting also the error vector $\mathbf{e}$ into the corresponding k_1, k_2, k_3 and $n-K$ parts and the syndrome into the corresponding $n-K, k_3$ and k_2 parts, we have the following situation

$$\begin{aligned}\mathbf{e}\mathbf{H}^\top &= \begin{pmatrix}\mathbf{e}_{k_1} & \mathbf{e}_{k_2} & \mathbf{e}_{k_3} & \mathbf{e}_{n-K}\end{pmatrix}\begin{pmatrix}\mathbf{A}^\top & p\mathbf{D}^\top & p^2\mathbf{F}^\top \\ \mathbf{B}^\top & p\mathbf{E}^\top & p^2\,\mathrm{Id}_{k_2} \\ \mathbf{C}^\top & p\,\mathrm{Id}_{k_3} & \mathbf{0}_{k_3\times k_2} \\ \mathrm{Id}_{n-K} & \mathbf{0}_{(n-K)\times k_3} & \mathbf{0}_{(n-K)\times k_2}\end{pmatrix} \\ &= \begin{pmatrix}\mathbf{s}_{n-K} & p\mathbf{s}_{k_3} & p^2\mathbf{s}_{k_2}\end{pmatrix} = \mathbf{s}.\end{aligned}$$

Let us assume that $\mathrm{wt}_L(\mathbf{e}_{k_1}) = v_1, \mathrm{wt}_L(\mathbf{e}_{k_2}) = v_2, \mathrm{wt}_L(\mathbf{e}_{k_3}) = v_3$ and $\mathrm{wt}_L(\mathbf{e}_{n-K}) = t - V$, for $V = v_1 + v_2 + v_3$. We thus get the following three conditions

$$\begin{aligned}\mathbf{e}_{k_1}\mathbf{A}^\top + \mathbf{e}_{k_2}\mathbf{B}^\top + \mathbf{e}_{k_3}\mathbf{C}^\top + \mathbf{e}_{n-K} &= \mathbf{s}_{n-K}, \\ p\mathbf{e}_{k_1}\mathbf{D}^\top + p\mathbf{e}_{k_2}\mathbf{E}^\top + p\mathbf{e}_{k_3} &= p\mathbf{s}_{k_3}, \\ p^2\mathbf{e}_{k_1}\mathbf{F}^\top + p^2\mathbf{e}_{k_2} &= p^2\mathbf{s}_{k_2}.\end{aligned}$$

As a first step, we set $\mathbf{H}_2 \in \mathbb{F}_p^{k_2\times(k_1+k_2)}$ as

$$\mathbf{H}_2 = \begin{pmatrix}\mathbf{F} & \mathrm{Id}_{k_2}\end{pmatrix}.$$

We can apply Lee-Brickell's algorithm over $\mathbb{F}_p$ equipped with the Lee metric to get

$$\mathbf{e}' = \begin{pmatrix}\mathbf{e}_1 & \mathbf{e}_2\end{pmatrix}$$

such that $\mathrm{wt}_L(\mathbf{e}_1) < v_1, \mathrm{wt}_L(\mathbf{e}_2) < v_2$ and

$$\begin{pmatrix}\mathbf{e}_1 & \mathbf{e}_2\end{pmatrix}\begin{pmatrix}\mathbf{F}^\top \\ \mathrm{Id}_{k_2}\end{pmatrix} = \mathbf{s}_{k_2}.$$

As a second step, we then set $\mathbf{H}_3 \in (\mathbb{Z}/p^2\mathbb{Z})^{(k_2+k_3)\times K}$ as

$$\mathbf{H}_3 = \begin{pmatrix}\mathbf{D}' & \mathrm{Id}_{k_3} \\ p\mathbf{H}_2 & \mathbf{0}_{k_2\times k_3}\end{pmatrix},$$

where $\mathbf{D}' = \begin{pmatrix}\mathbf{D} & \mathbf{E}\end{pmatrix}$.

Set $\mathbf{e}'_2 = \mathbf{e}' + p\tilde{\mathbf{e}}$, with the $\mathbf{e}'$ from the previous step and $\tilde{\mathbf{e}} \in (\mathbb{Z}/p\mathbb{Z})^{k_1+k_2}$, which is unknown.

Hence we can use part of Lee-Brickell's algorithm over $\mathbb{Z}/p^2\mathbb{Z}$ to get
$\mathbf{e}'' = \begin{pmatrix}\mathbf{e}'_2 & \mathbf{e}_3\end{pmatrix}$, such that $\mathrm{wt}_L(\mathbf{e}'_2) < v_1 + v_2, \mathrm{wt}_L(\mathbf{e}_3) < v_3$ and

$$\begin{pmatrix}\mathbf{e}'_2 & \mathbf{e}_3\end{pmatrix}\begin{pmatrix}(\mathbf{D}')^\top & p\mathbf{H}_2^\top \\ \mathrm{Id}_{k_3} & \mathbf{0}_{k_3\times k_2}\end{pmatrix} = \begin{pmatrix}\mathbf{s}_{k_3} & p\mathbf{s}_{k_2}\end{pmatrix}.$$

Observe that the condition

$$p\mathbf{e}'_2\mathbf{H}_2^\top = p(\mathbf{e}' + p\tilde{\mathbf{e}})\mathbf{H}_2^\top = p\mathbf{s}_{k_2}$$

is already verified from the previous step. Hence we are only left with the condition

$$\mathbf{e}_2'(\mathbf{D}')^\top + \mathbf{e}_3 = (\mathbf{e}' + p\tilde{\mathbf{e}})(\mathbf{D}')^\top + \mathbf{e}_3 = \mathbf{s}_{k_3}.$$

Thus, the part of Lee-Brickell's algorithm over $\mathbb{Z}/p^2\mathbb{Z}$ that we need is going through all $\tilde{\mathbf{e}} \in \mathbb{F}_p^{k_1+k_2}$ and check if

$$\mathbf{s}_{k_3} - (\mathbf{e}' + p\tilde{\mathbf{e}})(\mathbf{D}')^\top$$

has Lee weight less than or equal to v_3.

We get as output the error vector $\mathbf{e}'' = \begin{pmatrix}\mathbf{e}' + p\tilde{\mathbf{e}} & \mathbf{e}_3\end{pmatrix}$.

As third and last step, we set $\mathbf{H} \in (\mathbb{Z}/p^3\mathbb{Z})^{(n-k_1)\times n}$ as

$$\mathbf{H} = \begin{pmatrix} \mathbf{A}' & \mathrm{Id}_{n-K} \\ p\mathbf{H}_3 & \mathbf{0}_{(k_2+k_3)\times(n-K)} \end{pmatrix},$$

where $\mathbf{A}' = \begin{pmatrix}\mathbf{A} & \mathbf{B} & \mathbf{C}\end{pmatrix}$.

Set $\mathbf{e}_3' = \mathbf{e}'' + p^2\tilde{\mathbf{e}}'$, with the $\mathbf{e}''$ from the previous step and $\tilde{\mathbf{e}}' \in (\mathbb{Z}/p\mathbb{Z})^K$, which is unknown.

Hence we can use part of Lee-Brickell's algorithm over $\mathbb{Z}/p^3\mathbb{Z}$ to get $\mathbf{e} = \begin{pmatrix}\mathbf{e}_3' & \mathbf{e}_{n-K}\end{pmatrix}$, such that $\mathrm{wt}_L(\mathbf{e}_3') = V, \mathrm{wt}_L(\mathbf{e}_{n-K}) = t - V$ and

$$\begin{pmatrix}\mathbf{e}_3' & \mathbf{e}_{n-K}\end{pmatrix} \begin{pmatrix} (\mathbf{A}')^\top & p\mathbf{H}_3^\top \\ \mathrm{Id}_{n-K} & \mathbf{0}_{(n-K)\times(k_2+k_3)} \end{pmatrix} = \begin{pmatrix}\mathbf{s}_{n-K} & p\mathbf{s}'\end{pmatrix},$$

where $\mathbf{s}' = \begin{pmatrix}\mathbf{s}_{k_3} & p\mathbf{s}_{k_2}\end{pmatrix}$.

Observe that again the condition

$$p\mathbf{e}_3'\mathbf{H}_3^\top = p(\mathbf{e}'' + p^2\tilde{\mathbf{e}}')\mathbf{H}_3^\top = p\mathbf{s}'$$

is already verified from the previous step. Hence we are only left with the condition

$$\mathbf{e}_3'(\mathbf{A}')^\top + \mathbf{e}_{n-K} = (\mathbf{e}'' + p^2\tilde{\mathbf{e}}')(\mathbf{A}')^\top + \mathbf{e}_{n-K} = \mathbf{s}_{n-K}.$$

Thus, we go through all $\tilde{\mathbf{e}}' \in \mathbb{F}_p^K$ and check if

$$\mathbf{s}_{n-K} - (\mathbf{e}'' + p^2\tilde{\mathbf{e}}')(\mathbf{A}')^\top$$

has Lee weight $t - V$.

We get as output the error vector

$$\begin{aligned}\mathbf{e} &= \begin{pmatrix}\mathbf{e}'' + p^2\tilde{\mathbf{e}}' & \mathbf{e}_{n-K}\end{pmatrix} \\ &= \begin{pmatrix}\begin{pmatrix}\mathbf{e}' + p\tilde{\mathbf{e}} & \mathbf{e}_3\end{pmatrix} + p^2\tilde{\mathbf{e}}' & \mathbf{e}_{n-K}\end{pmatrix} \\ &= \begin{pmatrix}\begin{pmatrix}\begin{pmatrix}\mathbf{e}_1 & \mathbf{e}_2\end{pmatrix} + p\tilde{\mathbf{e}} & \mathbf{e}_3\end{pmatrix} + p^2\tilde{\mathbf{e}}' & \mathbf{e}_{n-K}\end{pmatrix}.\end{aligned}$$

Observe that this output vector satisfies all the conditions, *i.e.*, $\mathrm{wt}_L(\mathbf{e}) = t$ and $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$.

We can also provide the complexity analysis of the iterative algorithm.

Note that for simplicity we have set in the iterative algorithm $v_i = w$, for all i. Nevertheless, in the example we have set $\mathrm{wt}_L(\mathbf{e}_{k_i}) = v_i$ and $\mathrm{wt}_L(\mathbf{e}_{n-K}) = t - V$, for $V = \sum_{i=1}^{s} v_i$. We will also use the different v_i's in the complexity analysis.

For this define the function

$$c(r,k,v,s) = p^k k \min\{\mu_{p^s}^{-1}(v+1), r\} \left(\lceil \log_2(p^s) \rceil + \lceil \log_2(p^s) \rceil^2 \right).$$

Furthermore, let us denote by $LB(r,k,v_1,v_2)$ the cost of Lee-Brickell's algorithm over $\mathbb{F}_p$ for $\mathbf{H} \in \mathbb{F}_p^{r\times k}$ and $\mathrm{wt}_L(\mathbf{e}) = v_1 + v_2$, where v_1 is the parameter v.

Theorem 5.3.7. *The cost of the iterative Lee-Brickell algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ is given by*

$$LB(k_2, k_1+k_2, v_1, v_2) + \sum_{i=3}^{s} c\left(k_i, \sum_{j=1}^{i-1} k_j, v_i, i-1\right) + c(n-K, K, t-V, s)$$

binary operations.

Proof. Recall, that we want to find $\mathbf{e} \in (\mathbb{Z}/p^s\mathbb{Z})^n$, such that

$$\mathbf{e} = \begin{pmatrix} \mathbf{e}_{k_1} & \mathbf{e}_{k_2} & \cdots & \mathbf{e}_{k_s} & \mathbf{e}_{n-K} \end{pmatrix},$$

with $\mathrm{wt}_L(\mathbf{e}_{k_i}) = v_i$, for all $i \in \{1,\dots,s\}$ and $\mathrm{wt}_L(\mathbf{e}_{n-K}) = t - V$, where $V = \sum_{i=1}^{s} v_i$ and such that $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$, where $\mathbf{s} = \begin{pmatrix} \mathbf{s}_{n-K} & p\mathbf{s}_{k_s} & \cdots & p^{s-1}\mathbf{s}_{k_2} \end{pmatrix}$.

The first step of the iterative algorithm is to perform Lee-Brickell's algorithm over $\mathbb{F}_p$ on the inputs $\mathbf{H} \in \mathbb{F}_p^{k_2\times(k_1+k_2)}$, syndrome $\mathbf{s}_{k_2}$ and $\mathrm{wt}_L(\mathbf{e}) = v_1 + v_2$, where v_1 is the parameter v. Thus, we get a cost of $LB(k_2, k_1+k_2, v_1, v_2)$ binary operations.

The next steps, will all be computing $\mathbf{s} - \mathbf{e}\mathbf{A}^\top$ for $\mathbf{s} \in (\mathbb{Z}/p^i\mathbb{Z})^r, \mathbf{A} \in (\mathbb{Z}/p^i\mathbb{Z})^{r\times k}$, for $i \in \{2,\dots,s\}$. Since in the iterative algorithm we go through all $\mathbf{e} \in \mathbb{Z}/p^s\mathbb{Z}^k$ of Lee weight v, using early abort we get the cost of this step is $c(r,k,v,s)$. On the i-th step of the iterative algorithm, we are on the inputs $r = k_i, k = \sum_{j=1}^{i-1} k_j, v = v_i$ and $s = i-1$. Hence, we get the cost

$$\sum_{i=3}^{s} c\left(k_i, \sum_{j=1}^{i-1} k_j, v_i, i-1\right).$$

In the last step of the algorithm, we are computing again $\mathbf{s} - \mathbf{e}\mathbf{A}^\top$ for $\mathbf{s} \in (\mathbb{Z}/p^s\mathbb{Z})^r$ and $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{r\times k}$, which is given by $c(r,k,v,s)$, with the inputs $r = n-K, k = K, v = t-V, s = s$. Hence we get the claimed cost. □

Chapter 6

Comparison

In this chapter, we want to compare the workfactors of the considered algorithms. For this we provide the SAGE [101] code to compute the complexities of the algorithms in https://www.math.uzh.ch/aa/uploads/media/ISD.sage.

This can be done either by fixing the workfactor of the algorithms and comparing then the size of the input parity-check matrix, or by fixing the size of the considered parity-check matrix and comparing the cost of the algorithms.

Clearly, if applied in a code-based cryptosystem, the workfactor corresponds to the security level and the size of the input matrix corresponds to the public key size.

Note that since the input size and the workfactor are relative to each other, we will focus here only on fixing the workfactor and comparing the input sizes.

We denote by p^s the size of the underlying ring, where p is a prime and s is a positive integer, we denote by n the length of the code and by $k = \log_{p^s}(|\mathcal{C}|)$ and the rate of the code is then given by $R = \frac{\lceil k \rceil}{n}$.

Whenever we are over $\mathbb{Z}/p^s\mathbb{Z}$ with $s > 1$, we assume that the code has type $|\mathcal{C}| = (p^s)^{k_1}\left(p^{s-1}\right)^{k_2}$, since in the algorithms provided in Section 5.3 we only focus on a simplified form of the parity-check matrix, *i.e.*,

$$\mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathrm{Id}_{n-K} \\ p\mathbf{B} & \mathbf{0} \end{pmatrix}.$$

Thus, we also have the parameters k_1 and k_2. The size of the information set is given by $K = k_1 + k_2$, while $k = k_1 + k_2/s$. In addition, we denote by d the minimum distance in the respective metric and by t the error correction capacity, *i.e.*, $t = \left\lfloor \frac{d-1}{2} \right\rfloor$. We denote by v the optimal choice for the internal parameter of Stern's and Lee-Brickells' algorithm and by ℓ the optimal choice for the size of the zero-window in Stern's algorithm.

When fixing the workfactor, we optimize the size of the input matrix (*i.e.*, as small as possible) achieving this workfactor. Thus, the programs have to fix certain relations between the parameters. We will go through all lengths n, after a defined starting

point, we fix the rate of the code to be 1/2 and thus choose $k = \lfloor \frac{n}{2} \rfloor$ and we fix the minimum distance d such that it achieves the Gilbert-Varshamov bound in the considered metric. To assume that the considered codes achieve the Gilbert-Varshamov bound has several reasons; on one hand, if the parameters achieve this bound we have the existence result of codes and on the other hand, a random code achieves the asymptotic Gilbert-Varshamov bound with high probability.

Observe that over $\mathbb{Z}/p^s\mathbb{Z}$ with $s > 1$ there are several choices for k_1. We will thus always provide two choices: one with a small k_1 and one with a large k_1. Usually, it holds that decreasing k_1 decreases the input size for a fixed workfactor, and vice versa. Thus, the input size for any choice of k_1 should be between the two extremal provided cases.

In the following we denote by H.m. the Hamming metric, by L.m. the Lee metric and by L.B. the algorithm of Lee-Brickell. We provide a comparison for a fixed workfactor of 2^{80} binary operations, *i.e.*, corresponding to a security level of 80 bits in Table 6.1 and for a fixed workfactor of 2^{128} binary operations in Table 6.2.

To compare the Lee metric with the Hamming metric, we could compare several different parameter sets. In what follows, we focus on the algorithms over the same ambient spaces or at least spaces with the same amount of elements. In Table 6.3 and 6.4 we compare the provided algorithms over $\mathbb{F}_{101}$ in the Lee metric and the Hamming metric. One can clearly see, that over prime fields the Lee metric provides much lower input sizes than the Hamming metric. More in detail, we expect a reduction of the input size of over 89%.

In Table 6.5 and 6.6 we compare the Hamming metric over $\mathbb{F}_{2^8}$ with the Lee metric over $\mathbb{Z}/2^8\mathbb{Z}$. The Lee metric results again in lower input sizes. Of course, the comparison is not the most fair, as the finite field structure aids the ISD algorithm and thus a larger input matrix has to be used to gain the same workfactor. In fact, the key size reduction of using the Lee metric in this comparison is at least 91%. We can also observe the range of the input size depending on the choice of k_1 over $\mathbb{Z}/2^8\mathbb{Z}$ equipped with the Lee metric. Clearly, with decreasing k_1 we also decrease the input sizes and except for Stern's algorithm, the smallest possible non-trivial choice for k_1 achieves an input size which is half of the input size given by the maximal choice of k_1.

It makes most sense to compare the quaternary algorithms with the algorithms over the binary, due to the Gray isometry that connects them. Thus, in Table 6.7 and 6.8 we compare the algorithms over $\mathbb{Z}/4\mathbb{Z}$ in the Lee metric with the algorithms over the binary field equipped with the Hamming metric. The same behavior as before can also be observed in these tables, although much more extreme. For the minimal non-trivial choice of k_1 the reduction of input size compared to the Hamming metric is 99.9%, which is remarkable. For the maximal choice of k_1 we still have a fair reduction of 46%. And thus in this comparison the range of the input size depending on the choice of k_1 is also very large, *i.e.*, the input size for the minimal k_1 is a reduction of 90% from the input size for the maximal k_1.

The given comparisons lead to several general remarks.

1. First of all, we can see that the algorithms by Lee-Brickell and Stern really are improvements to Prange's algorithm also over the Lee metric. Which implies that we have correctly translated the corresponding speed-up techniques. However, for Stern's algorithm the optimal choice of ℓ, which indicates the size of the zero window is remarkably low. Since in the Lee metric we substantially increase the error correction capacity, the errors are also more likely to spread densely over the whole vector.

2. Secondly, in all cases the Lee metric provided an enormous key size reduction. The reason for this is most likely that for the same parameters the Lee metric can correct many more errors.

3. Lastly, in all comparisons over finite rings endowed with the Lee metric, it appears that a small k_1 is preferable for cryptographic purposes, as there we have the largest workfactor of the algorithms and in turn the smallest input size is needed to achieve a given cost. Having many codewords which live in a non-zero prime ideal seems to be more difficult for the algorithms. Note that this, however, might give an adversary some advantage for a structural attack.

Let us examine one of these examples in the tables more closely. In $\mathbb{Z}/2^8\mathbb{Z}$, *i.e.*, $p = 2$ and $s = 3$, we consider a linear Lee metric code $\mathcal{C}$ of length $n = 29$ and type $| \mathcal{C} | = 4^{12}2^{16}$. Thus $k = k_1 + k_2/s = 14 = \lfloor \frac{n}{2} \rfloor$ and the information set is of size $K = k_1 + k_2 = 28$. If we choose $d_L = 91$, then there must exist a linear Lee metric code with such parameters, since they achieve the Gilbert-Varshamov bound as

$$| B_L(d_L - 1, n, p^s) | > 11495 \cdot 10^{10}$$

and

$$4^{12}2^{16} = | \mathcal{C} | < \frac{\left(2^8\right)^{29}}{(| B_L(90, 29, 2^8) | - 1)\,(2^8 - 1)}.$$

Thus, this code can correct an error vector with Lee weight up to $t_L = 45$. For Prange's algorithm we assume that all these errors are located outside the information set, thus in $n - K = 1$ position. This might seem odd, but in this one position the Lee weight can go up to $\left\lfloor \frac{p^s}{2} \right\rfloor = 128$.

This particular set of parameters appears twice in Table 6.1, once for Prange's algorithm and once for Lee-Brickell's algorithm. However, this does not imply that in the Lee metric Lee-Brickell's algorithm is not an improvement of Prange's algorithm. In fact, the cost of Prange's algorithm for these parameters is given by 104.29 binary operations, while the cost of Lee-Brickell's algorithm for the same parameters is given by 96.49 binary operations.

Table 6.1: Cost of ISD algorithms in the Hamming and Lee metric, for fixed workfactor 80 bits.

ISD	metric	p	s	n	k	k_1	t	v	ℓ	input size (bits)
Prange	H.m.	2	1	895	447		51			200256
L.B.	H.m.	2	1	1077	538		61	2		289982
Stern	H.m.	2	1	1224	612		69	3	26	374544
Prange	H.m.	101	1	251	125		46			104866
Prange	H.m.	2	8	233	116		45			108576
L.B.	H.m.	101	1	279	139		51	1		129568
L.B.	H.m.	2	8	259	129		50	1		134160
Stern	H.m.	101	1	307	153		56	2	5	156880
Stern	H.m.	2	8	276	138		53	2	5	152352
Prange	L.m.	2	2	101	50	5	12			1050
Prange	L.m.	2	2	464	232	231	52			107646
L.B.	L.m.	2	2	120	60	1	14	13		238
L.B.	L.m.	2	2	555	277	276	62	2		154010
Stern	L.m.	2	2	220	110	1	25	12	0	438
Stern	L.m.	2	2	628	314	313	70	3	12	197190
Prange	L.m.	101	1	74	37		70			9115
Prange	L.m.	2	8	29	14	12	45			3360
Prange	L.m.	2	8	51	25	24	77			6656
L.B.	L.m.	101	1	89	44		86	86		13183
L.B.	L.m.	2	8	29	14	12	45	44		3360
L.B.	L.m.	2	8	51	25	24	77	76		6656
Stern	L.m.	101	1	101	50		96	9	5	16978
Stern	L.m.	2	8	43	21	18	66	31	0	7392
Stern	L.m.	2	8	74	37	36	107	20	2	13024

Table 6.2: Cost of ISD algorithms in the Hamming and Lee metric, for fixed workfactor 128 bits.

ISD	metric	p	s	n	k	k_1	t	v	ℓ	input size (bits)
Prange	H.m.	2	1	1675	837		94			701406
L.B.	H.m.	2	1	1893	946		106	2		895862
Stern	H.m.	2	1	2111	1055		118	4	38	1114080
Prange	H.m.	101	1	469	234		85			366135
Prange	H.m.	2	8	437	218		84			381936
L.B.	H.m.	101	1	501	250		91	1		417802
L.B.	H.m.	2	8	464	232		89	1		430592
Stern	H.m.	101	1	542	271		98	2	6	488985
Stern	H.m.	2	8	501	250		96	2	5	502000
Prange	L.m.	2	2	173	86	9	20			3106
Prange	L.m.	2	2	863	431	430	96			372382
L.B.	L.m.	2	2	201	100	1	23	21		598
L.B.	L.m.	2	2	963	481	480	107	2		463682
Stern	L.m.	2	2	381	190	1	43	21	0	1138
Stern	L.m.	2	2	1063	531	530	118	3	13	564982
Prange	L.m.	101	1	135	67		127			30334
Prange	L.m.	2	8	43	21	18	66			7392
Prange	L.m.	2	8	83	41	40	124			16128
L.B.	L.m.	101	1	151	75		142	142		37951
L.B.	L.m.	2	8	43	21	18	66	65		7392
L.B.	L.m.	2	8	93	46	45	139	138		19928
Stern	L.m.	101	1	171	85		159	16	9	48671
Stern	L.m.	2	8	57	28	24	86	21	0	12992
Stern	L.m.	2	8	135	67	66	200	25	1	40256

Table 6.3: Cost of ISD algorithms in the Hamming and Lee metric, for fixed workfactor 80 bits over $\mathbb{F}_{101}$.

ISD	metric	p	n	k	t	v	ℓ	input size (bits)
Prange	H.m.	101	251	125	46			104866
Prange	L.m.	101	74	37	70			9115
L.B.	H.m.	101	279	139	51	1		129568
L.B.	L.m.	101	89	44	86	86		13183
Stern	H.m.	101	307	153	56	2	5	156880
Stern	L.m.	101	101	50	96	9	5	16978

Table 6.4: Cost of ISD algorithms in the Hamming and Lee metric, for fixed workfactor 128 bits over $\mathbb{F}_{101}$.

ISD	metric	p	n	k	t	v	ℓ	input size (bits)
Prange	H.m.	101	469	234	85			366135
Prange	L.m.	101	135	67	127			30334
L.B.	H.m.	101	501	250	91	1		417802
L.B.	L.m.	101	151	75	142	142		37951
Stern	H.m.	101	542	271	98	2	6	488985
Stern	L.m.	101	171	85	159	16	9	48671

Table 6.5: Cost of ISD algorithms in the Hamming over $\mathbb{F}_{2^8}$ and Lee metric over $\mathbb{Z}/2^8\mathbb{Z}$, for fixed workfactor 80 bits.

ISD	metric	p	s	n	k	k_1	t	v	ℓ	input size (bits)
Prange	H.m.	2	8	233	116		45			108576
Prange	L.m.	2	8	29	14	12	45			3360
Prange	L.m.	2	8	51	25	24	77			6656
L.B.	H.m.	2	8	259	129		50	1		134160
L.B.	L.m.	2	8	29	14	12	45	44		3360
L.B.	L.m.	2	8	51	25	24	77	76		6656
Stern	H.m.	2	8	276	138		53	2	5	152352
Stern	L.m.	2	8	43	21	18	66	31	0	7392
Stern	L.m.	2	8	74	37	36	107	20	2	13024

Table 6.6: Cost of ISD algorithms in the Hamming over $\mathbb{F}_{2^8}$ and Lee metric over $\mathbb{Z}/2^8\mathbb{Z}$, for fixed workfactor 128 bits.

ISD	metric	p	s	n	k	k_1	t	v	ℓ	input size (bits)
Prange	H.m.	2	8	437	218		84			381936
Prange	L.m.	2	8	43	21	18	66			7392
Prange	L.m.	2	8	83	41	40	124			16128
L.B.	H.m.	2	8	464	232		89	1		430592
L.B.	L.m.	2	8	43	21	18	66	65		7392
L.B.	L.m.	2	8	93	46	45	139	138		19928
Stern	H.m.	2	8	501	250		96	2	5	502000
Stern	L.m.	2	8	57	28	24	86	21	0	12992
Stern	L.m.	2	8	135	67	66	200	25	1	40256

Table 6.7: Cost of ISD algorithms in the Hamming over the binary and Lee metric over the quaternary, for fixed workfactor 80 bits.

ISD	metric	p	s	n	k	k_1	t	v	ℓ	input size (bits)
Prange	H.m.	2	1	895	447		51			200256
Prange	L.m.	2	2	101	50	5	12			1050
Prange	L.m.	2	2	464	232	231	52			107646
L.B.	H.m.	2	1	1077	538		61	2		289982
L.B.	L.m.	2	2	120	60	1	118	13		238
L.B.	L.m.	2	2	555	277	276	62	2		154010
Stern	H.m.	2	1	1224	612		69	3	26	374544
Stern	L.m.	2	2	220	110	1	25	12	0	438
Stern	L.m.	2	2	628	314	313	70	3	12	197190

Table 6.8: Cost of ISD algorithms in the Hamming over the binary and Lee metric over the quaternary, for fixed workfactor 128 bits.

ISD	metric	p	s	n	k	k_1	t	v	ℓ	input size (bits)
Prange	H.m.	2	1	1675	837		94			701406
Prange	L.m.	2	2	173	86	9	20			3106
Prange	L.m.	2	2	863	431	430	96			372382
L.B.	H.m.	2	1	1893	946		106	2		895862
L.B.	L.m.	2	2	201	100	1	23	21		598
L.B.	L.m.	2	2	963	481	480	107	2		463682
Stern	H.m.	2	1	2111	1055		118	4	38	1114080
Stern	L.m.	2	2	381	190	1	43	21	0	1138
Stern	L.m.	2	2	1063	531	530	118	3	13	564982

Chapter 7

Conclusion and Future Work

In this first part of the thesis we have analyzed three ISD algorithms, namely Prange's algorithm, Lee-Brickell's algorithm and Stern's algorithm. We have provided the algorithms and their complexity analysis in the Hamming metric over the binary and over general finite fields using modern techniques for complexity speed-ups.

The main aim of this part was to provide the three algorithms in the Lee metric. For this we have separated the cases where the ambient space is $\mathbb{Z}/4\mathbb{Z}$, a general Galois ring of the form $\mathbb{Z}/p^s\mathbb{Z}$ and a prime finite field. By providing these algorithms and their costs, we have opened the path for Lee metric code-based cryptosystems. This seems to be an interesting and promising new direction for CBC, as in all considered ambient spaces, the ISD algorithms in the Lee metric have a larger workfactor than their counterparts in the Hamming metric. Thus, possible implications are lower key sizes when using the Lee metric in CBC.

Although the Lee metric is a promising alternative for CBC, it still needs a lot of further research. These are the first ISD algorithms in the Lee metric, thus they are still very premature and naive. Further improvements will clearly decrease the key size reductions we have observed in Chapter 6. In particular, one can investigate if the ISD algorithms which follow the second splitting idea of Dumer are more appropriate for the Lee metric. In addition, an analogue of the intermediate sum technique for the Lee metric should substantially decrease the workfactor of the ISD algorithms. And finally, since the Lee metric has a connection to the L_1-Norm, it can not be excluded that translating algorithms over lattices, which are endowed with the L_2-Norm, to the Lee metric might be faster than our approach, *i.e.*, translating algorithms endowed with the Hamming metric to the Lee metric.

Observe that there do not exist many Lee metric codes, which satisfy all the conditions we want in a public-key cryptosystem, namely: a fast decoding algorithm and a large error correction capacity. Thus, we can provide frameworks of code-based public key cryptosystems using the Lee metric but for a real primitive one needs to determine and propose an appropriate Lee metric code.

Determining a Lee metric code with favorable properties for cryptographic uses is a crucial step changing the interests in the Lee metric from theory to actual applications.

Finally, since the rank metric, as well as the Lee metric, are promising alternatives to the Hamming metric for cryptographic purposes, this work might inspire further research in different metrics. In fact, if the syndrome decoding problem over a certain metric is NP-complete and the best known solvers, *i.e.*, ISD algorithms, have a large cost then such a metric might provide promising applications in code-based cryptography.

Note that the Lee metric has already found some recent applications for example in signature schemes [14] and in single server private information retrieval [6].

Part II

Density

Chapter 8

Introduction

A natural question to ask, is:

How likely is it for a randomly chosen $x \in \mathbb{Z}$, to be in a subset $T \subseteq \mathbb{Z}$?

Observe that there is no uniform probability distribution on $\mathbb{Z}$. In fact, if every element in $\mathbb{Z}$ has the same probability $0 \leq p \leq 1$ and at the same time the probability of the whole space $\mathbb{Z}$ should be 1, we get a contradiction by the countable additivity property for disjoint sets:

$$1 = \mathbb{P}(\mathbb{Z}) = \sum_{x \in \mathbb{Z}} \mathbb{P}(\{x\}) = \sum_{x \in \mathbb{Z}} p.$$

If we choose $p = 0$, then we get $1 = 0$ and if we choose $p > 0$, we would get $1 = \infty$.

Therefore, the notion of natural density was introduced. Originally it was defined over the positive integers $\mathbb{N}$ but it can easily be generalized to $\mathbb{Z}$.

The definition of the natural density is quite canonical: one first counts how many elements of the d-dimensional cube of length H are in the subset of interest T, then one divides this quantity by the size of the d-dimensional cube and finally one lets H go to infinity, *i.e.*, for d a positive integer and $T \subseteq \mathbb{Z}^d$, the natural density of T is given by

$$\rho(T) = \lim_{H \to \infty} \frac{|\, T \cap [-H, H[^d|}{(2H)^d},$$

if this limit exists.

Analogously, one can define the upper density $\overline{\rho}$ and the lower density $\underline{\rho}$ by exchanging the limit with the limit superior, respectively with the limit inferior.

Other densities have been introduced as well, for example by Dirichlet and Schnirel'man (for more examples see [48]). In this part of the thesis, we will focus on the natural density only.

The introduction of the natural density dates back to questions asked by Mertens and Césaro in the 1870's, namely:

What is the probability that two randomly chosen integers are coprime?

Mertens [75] and Césaro [33] solved this question independently with the result being $\frac{6}{\pi^2}$. In fact, the result is $\frac{1}{\zeta(2)}$ and proving that $\zeta(2) = \sum_{n\geq 1} \frac{1}{n^2} = \frac{\pi^2}{6}$ is well known as the Basel problem. We will show an elegant proof of their result in Section 10.1.

To compute the natural density there are various tools. In this thesis we will focus on the technique provided by Poonen and Stoll [91] (also in [92]), namely the local to global principle. This technique allows the computation of the natural density by characterizing the target set locally, *i.e.*, over the p-adic integers.

The local to global principle can also be applied to the algebraic integers of some number field. In fact, its analogous version was provided by Bright *et al.* in [28]. Instead of using their translation to the algebraic integers, in Section 10.4 we will provide the technique introduced in [78], where we use a function F_p, that brings us from the p-adic integers, where we can apply the local to global principle of Poonen and Stoll, to the set of interest in the algebraic integers. Although being less elegant, this method provides a more straightforward application of the local to global principle to number fields.

Furthermore, when seeing the natural density as a $\mathbb{Z}$-analogue of the uniform probability distribution, a natural question is if we can also define the analogue of the mean and the variance. In fact, using tools of analytic number theory, we will compute some of the mean and variances of the sets of interest in Chapter 11. Realizing that these results all follow a certain rule, that becomes visible with the local to global principle, we provide in Chapter 12 an addendum to the principle, that allows to compute the mean and variance directly in a more elegant way.

8.1 Organization of this Part

This second part of the thesis is organized as follows. In Section 8.3 we introduce the notation, that we use throughout this part and in Chapter 9 we recall the basic definitions that will be needed.

More in detail, in Section 9.2 we show some properties of the natural density and see some first examples. In Section 9.3 we state the main tool of this thesis to compute natural densities, namely the local to global principle, and in Section 9.4 we outline the strategy, that we follow to apply this tool.

Then, we use the local to global principle to compute the natural density of coprime pairs, coprime m-tuples, Eisenstein polynomials and rectangular unimodular matrices over the integers in Chapter 10.

In Chapter 11 we introduce the notion of mean and variance with respect to the natural density and compute them for the complement sets of the coprime pairs, coprime m-tuples, Eisenstein polynomials and rectangular unimodular matrices, using tools from analytic number theory.

In Chapter 12 we give an addendum to the local to global principle, which, using the same technique, gives directly the mean and the variance of the target object. We then see again the examples from Chapter 11 as a corollary of the extended local to global principle.

Finally, in Chapter 13 we draw some concluding remarks and state some open problems for future work.

8.2 New Results in this Part

In this part of the thesis we will focus on four sets, whose density can be computed through the local to global principle, namely

1. the set of coprime pairs,
2. the set of coprime m-tuples,
3. the set of Eisenstein polynomials (monic and non-monic),
4. the set of rectangular unimodular $n \times m$ matrices, for $n < m$.

Clearly, the set of rectangular unimodular $n \times m$ matrices contains the set of coprime m-tuples (by setting $n = 1$), which in turn contains the set of coprime pairs (by setting $m = 2$). Nevertheless, we give the full results and proofs or their outlines for all sets. On one hand for the sake of completeness and on the other hand to encourage interested readers to apply the local to global principle to the sets in this order, as this is the best learning practice.

In Chapter 10 we compute the densities of these four sets over $\mathbb{Z}$. The results for the sets of coprime pairs and coprime m-tuples are known (see [75, 33, 85]), but giving the proof via the local to global principle by Poonen and Stoll [91] is, up to the best of our knowledge, only available in this thesis. The result for the Eisenstein polynomials is due to Dubickas [42] in the monic case, and to Heyman and Shparlinski [52] in the non-monic case. The idea of the proof via the local to global principle is already available in the thesis of Micheli [77].

The density of rectangular unimodular matrices was already computed in [73]. Although there was a mistake in the proof, the claimed density is correct and was proven as a corollary in [78]:

On rectangular unimodular matrices over the algebraic integers,
by Giacomo Micheli and Violetta Weger, published in SIAM Journal of discrete mathematics, Volume 33, Number 1, pages 425–437, 2019. DOI.10.1137/18M1177093

In Section 10.4 we provide the idea of applying the local to global principle over the algebraic integers over a number field. This follows the method provided in [78]. We also state some known density results over the algebraic integers, *i.e.*, for coprime pairs, coprime m-tuples [44] and rectangular unimodular matrices [78]. The density results for the Eisenstein polynomials over the algebraic integers is not known yet. In fact, it is a ongoing project of the author, in collaboration with Simran Tinani.

The computation of the mean and the variance with respect to the natural density of Eisenstein polynomials was provided in [70]. Thus, the introduction of such objects is, up to our best knowledge, also due to [70]. Following their proof, we will compute the mean and the variance of coprime pairs, coprime m-tuples and rectangular unimodular matrices in Chapter 11. These results are completely new and only available in this thesis.

Finally, Chapter 12 is based on the following article [79]:

Local to global principle for expected values,
by Giacomo Micheli, Severin Schraven and Violetta Weger, arXiv preprint arXiv:2008.06235, 2020.

For this thesis we have added also the variance and further corollaries, providing the application of the addendum to the four sets of interest, *i.e.*, to the coprime pairs, the coprime m-tuples, the Eisenstein polynomials and the rectangular unimodular matrices. When comparing these results to the computations of Chapter 11, one can see, that the results are consistent and clearly, the application of the addendum provides a more elegant and shorter proof.

8.3 Notation

We denote by $\mathbb{N}$ the set of positive integers, by $\mathbb{N}_0$ the non-negative integers and by $\mathbb{R}_{\geq 0}$ we denote the non-negative reals. For a domain $\mathcal{R}$ and $n < m \in \mathbb{N}$, we denote either by $\mathrm{Mat}_{n\times m}(\mathcal{R})$ or by $\mathcal{R}^{n\times m}$ the set of $n \times m$ matrices over $\mathcal{R}$. For $a \in \mathcal{R}$ we will denote by $I(a), (a)$ or $a\mathcal{R}$ the ideal generated by a. If S is a set, then we denote by 2^S its powerset and by S^C its complement. We denote by $\mathcal{P}$ the set of natural primes. For $p \in \mathcal{P}$, we denote by $\mathbb{Q}_p$ the p-adic numbers and by $\mathbb{Z}_p$ the p-adic integers.

Let $M_{\mathbb{Q}} = \{\infty\} \cup \mathcal{P}$ be the set of all places of $\mathbb{Q}$, where we denote by ∞ the unique archimedean place of $\mathbb{Q}$.

For $d \in \mathbb{N}$, let μ_∞ denote the Lebesgue measure on $\mathbb{R}^d$ and μ_p the normalized Haar measure on $\mathbb{Z}_p^d$.

For T a subset of a metric space, let us denote by $\partial(T)$ its boundary, by T° its interior and by $\overline{T}$ its closure.

For $a, b \in \mathbb{R}$, we denote by $[a, b]$ the set of all real numbers, larger than or equal to a and smaller than or equal to b.

Chapter 9

Preliminaries

9.1 Natural Density

For $d \in \mathbb{N}$, the natural density of a set $T \subseteq \mathbb{Z}^d$ is defined by first restricting to a d-dimensional cube of height H, thus we can count how many elements in the cube are in T, and then dividing by the size of the cube, and finally letting H go to infinity.

Definition 9.1.1 (Density on Integers)**.** Let $d \in \mathbb{N}$. The *natural density* of a set $T \subseteq \mathbb{Z}^d$ is defined to be

$$\rho(T) = \lim_{H \to \infty} \frac{|T \cap [-H, H[^d|}{(2H)^d},$$

if the limit exists. Then, one defines the upper density $\bar{\rho}$ and the lower density $\underline{\rho}$ equivalently with the lim sup and the lim inf respectively.

In more generality, we say that D is a density, if it satisfies the following axioms, slightly changed from the definition by Grekos in [48]. Note, that we cannot see D as a function from 2^X to $[0, 1]$, as it is a priori not defined for every set in 2^X.

Definition 9.1.2 (Density Axioms)**.** A *density* D on an abelian semigroup X has to satisfy the following density axioms:

1. $D(\emptyset) = 0$ and $D(X) = 1$,
2. if $A \subseteq B \subseteq X$, then $D(A) \leq D(B)$,
3. if $T \subseteq X$, then $D(T) \in [0, 1]$,
4. if $A \subseteq X$ and $x \in X$, then $D(A + \{x\}) = D(A)$, where
$$A + \{x\} = \{a + x \mid a \in A\},$$
5. if $F \in X$ is finite, then $D(F) = 0$,
6. if $A, B \subseteq X$ with $A \cap B = \emptyset$, then $D(A \cup B) = D(A) + D(B)$,

assuming that $D(A), D(B)$ and $D(T)$ exist.

Clearly, the natural density is a density. We will include the proof for completeness.

Proposition 9.1.3. *The natural density is a density.*

Proof. Let ρ be as defined in Definition 9.1.1 on $\mathbb{Z}^d$. We will show that all axioms of Definition 9.1.2 are satisfied.

1. Clearly $\rho(\emptyset) = 0$, since

$$\rho(\emptyset) = \lim_{H\to\infty} \frac{|\ \emptyset \cap [-H, H[^d|}{(2H)^d} = \lim_{H\to\infty} \frac{0}{(2H)^d} = 0$$

   and $\rho(\mathbb{Z}^d) = 1$, since

$$\rho(\mathbb{Z}^d) = \lim_{H\to\infty} \frac{|\ \mathbb{Z}^d \cap [-H, H[^d|}{(2H)^d} = \lim_{H\to\infty} \frac{(2H)^d}{(2H)^d} = 1.$$

2. If $A \subseteq B \subseteq \mathbb{Z}^d$, then $\forall H \in \mathbb{N}$ we have that

$$|\ A \cap [-H, H[^d| \leq |\ B \cap [-H, H[^d|$$

   and the claim follows, as

$$\rho(A) = \lim_{H\to\infty} \frac{|\ A \cap [-H, H[^d|}{(2H)^d} \leq \lim_{H\to\infty} \frac{|\ B \cap [-H, H[^d|}{(2H)^d} = \rho(B).$$

3. This follows directly from 1 and 2 by setting on one hand $A = \emptyset$ and $B = T$, and on the other hand setting $A = T$ and $B = \mathbb{Z}^d$.

4. Let $A \subseteq \mathbb{Z}^d$ and $x \in \mathbb{Z}^d$ and let us denote by $|x|$ the height of the d-tuple, *i.e.*, $|x| = \max\{|x_i| \ \mid i \in \{1, \ldots, d\}\}$. Then for $H > |x|$ the claim follows, as

$$\begin{aligned}
\rho(A) &= \lim_{H\to\infty} \frac{|\ A \cap [-H, H[^d|}{(2H)^d} \\
&= \lim_{H\to\infty} \frac{|\ A \cap [-H+|x|, H-|x|[^d|}{(2(H-|x|))^d} \frac{(2(H-|x|))^d}{(2H)^d} \\
&\leq \lim_{H\to\infty} \frac{|\ (A+\{x\}) \cap [-H, H[^d|}{(2H)^d} \\
&= \lim_{H\to\infty} \frac{|\ A \cap [-H+|x|, H+|x|[^d|}{(2H)^d} = \rho(A+\{x\}) \\
&\leq \lim_{H\to\infty} \frac{|\ A \cap [-H-|x|, H+|x|[^d|}{(2(H+|x|))^d} \frac{(2(H+|x|))^d}{(2H)^d} \\
&= \rho(A).
\end{aligned}$$

5. Let $F \subset \mathbb{Z}^d$, with $\mid F \mid < \infty$ and let $f \in F$ be the element in F with the largest absolute value. Since we can assume that $H > f$ the claim follows, as

$$\rho(F) = \lim_{H\to\infty} \frac{\mid F \cap [-H, H[^d \mid}{(2H)^d} = \lim_{H\to\infty} \frac{\mid F \mid}{(2H)^d} = 0.$$

6. If $A, B \subseteq \mathbb{Z}^d$ with $A \cap B = \emptyset$, then

$$\begin{aligned}
\rho(A \cup B) &= \lim_{H\to\infty} \frac{\mid (A \cup B) \cap [-H, H[^d \mid}{(2H)^d} \\
&= \lim_{H\to\infty} \frac{\mid (A \cap [-H, H[^d) \cup (B \cap [-H, H[^d) \mid}{(2H)^d} \\
&= \lim_{H\to\infty} \frac{\mid A \cap [-H, H[^d \mid + \mid B \cap [-H, H[^d \mid}{(2H)^d} \\
&= \rho(A) + \rho(B).
\end{aligned}$$

   In addition, whenever A and B are disjoint and their densities exist, also $\rho(A \cup B)$ exists.

□

An important difference between the natural density and a measure is, that the natural density does not satisfy the countable additivity property. Even stronger, Boole's inequality, also known as union bound, does not hold, *i.e.*,

$$\rho\left(\bigcup_{i\in I} A_i\right) \not\leq \sum_{i\in I} \rho(A_i),$$

for I a countable set and A_i subsets of $\mathbb{Z}$. An easy counterexample is given by taking $I = \mathbb{Z}$ and $A_i = \{i\}$ for all $i \in \mathbb{Z}$, since

$$\rho\left(\bigcup_{i\in\mathbb{Z}} \{i\}\right) = \rho(\mathbb{Z}) = 1 \not\leq \sum_{i\in\mathbb{Z}} \rho(\{i\}) = 0.$$

Nevertheless, this does hold for finite sets I, *i.e.*,

$$\rho\left(\bigcup_{i\in I} A_i\right) \leq \sum_{i\in I} \rho(A_i),$$

for I a finite set and A_i subsets of $\mathbb{Z}$. To see this result we can iteratively use 6 of Definition 9.2.1.

Lastly, we recall here the well-known *Riemann zeta function*:

$$\zeta(s) = \sum_{n\geq 1} \frac{1}{n^s}.$$

An equivalent definition was given by Euler, as

$$\zeta(s) = \prod_{p \in \mathcal{P}} \left(1 - \frac{1}{p^s}\right)^{-1}.$$

9.2 Properties

Apart from the axioms of a density in Definition 9.1.2 the natural density also satisfies further properties.

Proposition 9.2.1 (Properties of Density)**.** *Let $d \in \mathbb{N}$. For $A, B \subset \mathbb{Z}^d$, we have the following properties*

1. $\max\{\rho(A), \rho(B)\} \leq \rho(A \cup B) \leq \min\{\rho(A) + \rho(B), 1\}$,
2. *if* $\mid A \triangle B \mid < \infty$, *then* $\rho(A) = \rho(B)$,
3. *if* $B \subseteq A$, *then* $\rho(A \setminus B) = \rho(A) - \rho(B)$,

assuming that $\rho(A)$ and $\rho(B)$ exist.

Proof. 1. Since

$$\mid (A \cup B) \cap [-H, H[^d \mid \geq \mid A \cap [-H, H[^d \mid$$

and also

$$\mid (A \cup B) \cap [-H, H[^d \mid \geq \mid B \cap [-H, H[^d \mid$$

the first inequality follows. Since, also

$$\mid (A \cup B) \cap [-H, H[^d \mid \leq \mid A \cap [-H, H[^d \mid + \mid B \cap [-H, H[^d \mid$$

the second inequality follows.

2. The symmetric difference $A \triangle B$ is defined as

$$(A \cup B) \setminus (A \cap B) = (A \setminus B) \cup (B \setminus A).$$

Since $A \triangle B$ is finite, $A \setminus B$ and $B \setminus A$ are finite as well. Recall that finite sets have density zero. Since we have that $A = (B \cap A) \cup (A \setminus B)$, and $(B \cap A)$ and $(A \setminus B)$ are disjoint, we get that

$$\begin{aligned}
\rho(A) &= \lim_{H \to \infty} \frac{\mid A \cap [-H, H[^d \mid}{(2H)^d} \\
&= \lim_{H \to \infty} \frac{\mid ((B \cap A) \cup (A \setminus B)) \cap [-H, H[^d \mid}{(2H)^d} \\
&= \lim_{H \to \infty} \frac{\mid (B \cap A) \cap [-H, H[^d \mid + \mid (A \setminus B) \cap [-H, H[^d \mid}{(2H)^d} \\
&= \lim_{H \to \infty} \frac{\mid (B \cap A) \cap [-H, H[^d \mid}{(2H)^d}.
\end{aligned}$$

The same holds for B, *i.e.*,

$$\rho(B) = \lim_{H\to\infty} \frac{|\,(B \cap A) \cap [-H, H[^d|}{(2H)^d} = \rho(A).$$

3. Since $B \subseteq A$, we have that $A = B \cup (A \setminus B)$ and this union is clearly disjoint, *i.e.*, $B \cap (A \cap B^C) = \emptyset$. Hence, we have that

$$\begin{aligned}
\rho(A) &= \lim_{H\to\infty} \frac{|\,A \cap [-H, H[^d|}{(2H)^d} \\
&= \lim_{H\to\infty} \frac{|\,(B \cup (A \cap B^C)) \cap [-H, H[^d|}{(2H)^d} \\
&= \lim_{H\to\infty} \frac{|\,(B \cap [-H, H[^d) \cup ((A \cap B^C) \cap [-H, H[^d)\,|}{(2H)^d} \\
&= \lim_{H\to\infty} \frac{|\,B \cap [-H, H[^d| + |\,(A \cap B^C) \cap [-H, H[^d|}{(2H)^d} \\
&= \lim_{H\to\infty} \frac{|\,B \cap [-H, H[^d|}{(2H)^d} + \lim_{H\to\infty} \frac{|\,(A \cap B^C) \cap [-H, H[^d|}{(2H)^d} \\
&= \rho(B) + \rho(A \setminus B).
\end{aligned}$$

□

From 3 of Proposition 9.2.1 and 1 of Definition 9.1.2 it follows directly, that

$$\rho(A^C) = 1 - \rho(A).$$

We can now give some examples of natural densities.

Example 9.2.2 (Density of $\mathcal{P}$). The density of all primes $\mathcal{P}$ is

$$\rho(\mathcal{P}) = 0.$$

Proof. Let us denote for $x \in \mathbb{N}$ by $\pi(x)$ the amount of prime numbers smaller than x, *i.e.*,

$$\pi(x) = |\,\{p \in \mathcal{P} \mid p \leq x\}\,|\,.$$

Recall, that by the prime number theorem, we have that

$$\pi(x) \sim \frac{x}{\ln(x)}.$$

Hence, we have that

$$\begin{aligned}
\rho(\mathcal{P}) &= \lim_{H\to\infty} \frac{|\,\mathcal{P} \cap [-H, H[|}{2H} \\
&= \lim_{H\to\infty} \frac{\pi(H)}{2H} \\
&= \lim_{H\to\infty} \frac{H}{2H\ln(H)} \\
&= \lim_{H\to\infty} \frac{1}{2\ln(H)} = 0.
\end{aligned}$$

□

Example 9.2.3 (Density of Squares). The density of $S = \{a^2 \mid a \in \mathbb{Z}\}$ is

$$\rho(S) = 0.$$

Proof. For $x \in \mathbb{N}$, let us denote by $s(x)$ the amount of squares smaller than x, *i.e.*,

$$s(x) = \mid \{a^2 \mid a \in \mathbb{Z}, a^2 \leq x\} \mid .$$

Observe, that for all x we have that $s(x) \leq \sqrt{x}$. Hence, we get that

$$\begin{aligned}
\rho(S) &= \lim_{H\to\infty} \frac{\mid S \cap [-H, H[\mid}{2H} \\
&= \lim_{H\to\infty} \frac{s(H)}{2H} \\
&\leq \lim_{H\to\infty} \frac{\sqrt{H}}{2H} \\
&= \lim_{H\to\infty} \frac{1}{2\sqrt{H}} = 0.
\end{aligned}$$

□

Example 9.2.4 (Density of Powers of 2). The density of $\{2^n \mid n \in \mathbb{N}\}$ is

$$\rho(\{2^n \mid n \in \mathbb{N}\}) = 0.$$

Proof. Since $\mid \{2^n \mid n \in \mathbb{N}\} \cap [-H, H[\mid = \lfloor \log_2(H) \rfloor$, we get the claim as

$$\begin{aligned}
\rho(\{2^n \mid n \in \mathbb{N}\}) &= \lim_{H\to\infty} \frac{\mid \{2^n \mid n \in \mathbb{N}\} \cap [-H, H[\mid}{2H} \\
&= \lim_{H\to\infty} \frac{\lfloor \log_2(H) \rfloor}{2H} = 0.
\end{aligned}$$

□

Example 9.2.5 (Density of $p\mathbb{Z}$). For $p \in \mathcal{P}$, the density of $p\mathbb{Z}$ is

$$\rho(p\mathbb{Z}) = \frac{1}{p}.$$

Proof. Since $\mid p\mathbb{Z} \cap [-H, H[\mid = \left\lfloor \frac{2H}{p} \right\rfloor$, we get the claim as

$$\rho(p\mathbb{Z}) = \lim_{H\to\infty} \frac{\mid p\mathbb{Z} \cap [-H, H[\mid}{2H} = \lim_{H\to\infty} \frac{\left\lfloor \frac{2H}{p} \right\rfloor}{2H} = \lim_{H\to\infty} \frac{1}{p} = \frac{1}{p}.$$

□

Example 9.2.6 (Density of Invertible Matrices over $\mathbb{Z}$). For n a positive integer, the density of $\mathrm{GL}_n(\mathbb{Z})$, the invertible matrices in $\mathbb{Z}^{n\times n}$, is

$$\rho(GL_n(\mathbb{Z})) = 0.$$

Proof. Let us consider $A \in \mathbb{Z}^{n\times n}$ with entries in $[-H, H[$. If we fix all entries of the $n \times n$ matrix A but one, *e.g.* $a_{n,n}$, we have $(2H)^{n^2-1}$ choices for A. Let us denote by $A_{i,j}$ the matrix A without the i-th row and j-th column, then the determinant of A is given by the Laplace formula as

$$\det(A) = \sum_{j=1}^{n}(-1)^{n+j}a_{n,j}\det(A_{n,j}) = \sum_{j=1}^{n-1}(-1)^{n+j}a_{n,j}\det(A_{n,j}) + a_{n,n}\det(A_{n,n}).$$

In order for this to be ± 1, we only have at most two choices for $a_{n,n}$. Thus, we get that

$$\rho(\mathrm{GL}_n(\mathbb{Z})) = \lim_{H\to\infty}\frac{|\,\mathrm{GL}_n(\mathbb{Z}) \cap [-H, H[^{n^2}|}{(2H)^{n^2}} = \lim_{H\to\infty}\frac{2(2H)^{n^2-1}}{(2H)^{n^2}} = 0.$$

□

We can also give examples of target sets T for which the natural density does not exist. For this we compute the upper and the lower density of T which do not coincide.

We cover here the extreme case, where the lower density of T is 0 and the upper density of T is 1.

Example 9.2.7. Let $T = A \cup B$, where

$$A = \bigcup_{n\geq 0}\left\{2^{2^{2n}}, 2^{2^{2n}}+1, \ldots, 2^{2^{2n+1}}-1\right\},$$
$$B = \bigcup_{n\geq 0}\left\{-2^{2^{2n}}, -2^{2^{2n}}-1, \ldots, -2^{2^{2n+1}}+1\right\}.$$

Then,

$$\underline{\rho}(T) = 0,$$
$$\overline{\rho}(T) = 1.$$

Proof. First, let us call

$$A_n = \left\{2^{2^{2n}}, 2^{2^{2n}}+1, \ldots, 2^{2^{2n+1}}-1\right\}$$
$$B_n = \left\{-2^{2^{2n}}, -2^{2^{2n}}-1, \ldots, -2^{2^{2n+1}}+1\right\}.$$

We have that $|\,A_n\,| = |\,B_n\,| = 2^{2^{2n+1}} - 2^{2^{2n}}$.

For the limit superior we get a sequence that bounds $|\,T \cap [-H, H[|$ from below by choosing $H = 2^{2^{2n+1}}$, *i.e.*, the minimal H, such that A_n and B_n are still fully inside $[-H, H[$.

$$
\begin{aligned}
1 &\geq \limsup_{H\to\infty} \frac{|\, T \cap [-H, H[\,|}{2H} \\
&\geq \lim_{n\to\infty} \frac{2 \mid A_n \mid}{2 \cdot 2^{2^{2n+1}}} \\
&= \lim_{n\to\infty} \frac{2^{2^{2n+1}} - 2^{2^{2n}}}{2^{2^{2n+1}}} \\
&= 1 - \lim_{n\to\infty} 2^{\left(2^{2n} - 2^{2n+1}\right)} \\
&= 1 - \lim_{n\to\infty} \frac{1}{2^{2^{2n}}} \\
&= 1.
\end{aligned}
$$

Hence,

$$
\overline{\rho}(T) = \limsup_{H\to\infty} \frac{|\, T \cap [-H, H[\,|}{2H} = 1.
$$

For the limit inferior, we get a sequence that bounds $|\, T \cap [-H, H[\,|$ from above by choosing $H = 2^{2^{2n}}$, *i.e.*, the maximal H, such that A_n and B_n are not inside $[-H, H[$.

Observe, that $|\, T \cap [-H, H[\,|$ is clearly smaller than

$$
2 \left| \left\{ 1, 2, \ldots, 2^{2^{2(n-1)+1}} - 1 \right\} \right|,
$$

since $2^{2^{2(n-1)+1}} - 1$ is the last element in A_{n-1}, and we have filled in all gaps between A_i and A_{i+1}, for $i < n - 1$.

$$
\begin{aligned}
0 &\leq \liminf_{H\to\infty} \frac{|\, T \cap [-H, H[\,|}{2H} \\
&\leq \lim_{n\to\infty} \frac{2\left(2^{2^{2(n-1)+1}} - 1\right)}{2 \cdot 2^{2^{2n}}} \\
&= \lim_{n\to\infty} \frac{2^{2^{2n-1}} - 1}{2^{2^{2n}}} \\
&= \lim_{n\to\infty} \frac{1}{2^{2^{2n-1}}} - 0 \\
&= 0.
\end{aligned}
$$

Hence,

$$
\underline{\rho}(T) = \liminf_{H\to\infty} \frac{|\, T \cap [-H, H[\,|}{2H} = 0.
$$

□

9.3 Local to Global Principle

There are various tools to compute the natural density of a target set. In this thesis we will focus on the local to global principle, introduced by Poonen and Stoll in [91]. This will be the main tool of this part of the thesis.

Since the local to global principle computes the natural densities through characterizing the set locally, *i.e.*, in the p-adic integers, we will first recall some basic definitions.

Definition 9.3.1 (p-adic Valuation)**.** Let p be a prime, for $x \in \mathbb{Z} \setminus \{0\}$ we define the *p-adic valuation* of x to be

$$\mathrm{ord}_p(x) = \max\{v \in \mathbb{N} \mid \exists k \in \mathbb{Z},\ p^v \cdot k = x\}$$

and $\mathrm{ord}_p(0) = \infty$. For $y = \frac{a}{b} \in \mathbb{Q}$, we can define the p-adic valuation of y to be

$$\mathrm{ord}_p(y) = \mathrm{ord}_p(a) - \mathrm{ord}_p(b).$$

Definition 9.3.2 (p-adic Norm)**.** Let p be a prime, we define the *p-adic norm*

$$|\cdot|_p \colon \mathbb{Q} \to \mathbb{R}_{\geq 0},$$

as

$$|\ x\ |_p = \begin{cases} \frac{1}{p^{\mathrm{ord}_p(x)}} & \text{if } x \neq 0, \\ 0 & \text{else.} \end{cases}$$

Definition 9.3.3 (p-adic Numbers)**.** Let p be a prime, then the *p-adic numbers*, denoted by $\mathbb{Q}_p$ is the completion of $\mathbb{Q}$ with respect to the p-adic norm.

Definition 9.3.4 (p-adic Integers)**.** Let p be a prime, we denote by $\mathbb{Z}_p$ the *p-adic integers*, defined as

$$\mathbb{Z}_p = \{\alpha \in \mathbb{Q}_p \mid |\ \alpha\ |_p \leq 1\}.$$

Observe that every element $\alpha \in \mathbb{Q}_p$ has a unique p-adic expansion of the form

$$\alpha = \sum_{i=k}^{\infty} a_i p^i,$$

for any $k \in \mathbb{Z}$ and $a_i \in \{0, \ldots, p-1\}$ for all i. Similarly, every element $\beta \in \mathbb{Z}_p$ has a unique p-adic expansion of the form

$$\beta = \sum_{i=0}^{\infty} b_i p^i,$$

with $b_i \in \{0, \ldots, p-1\}$ for all i.

Definition 9.3.5 (Topology on $\mathbb{Z}_p$)**.** We can define the following *topology on the p-adic integers*, by setting all sets of the form

$$\left\{a + \alpha p^k \mid \alpha \in \mathbb{Z}_p,\ k \in \mathbb{N}, a \in \left\{1, \ldots, p^k\right\}\right\}$$

as basis of the open sets.

Proposition 9.3.6. *The set $p\mathbb{Z}_p$ is open and closed.*

Proof. Clearly, $p\mathbb{Z}_p$ is open by definition. We can also write

$$p\mathbb{Z}_p = \bigcap_{i=1}^{p-1} (i + p\mathbb{Z}_p)^C,$$

and since $(i+p\mathbb{Z}_p)$ is open, its complement $(i+p\mathbb{Z}_p)^C$ is closed and a finite intersection of closed sets is again closed. □

Definition 9.3.7 (Haar Measure on $\mathbb{Z}_p$)**.** We can define the *Haar measure on* $\mathbb{Z}_p$ as

$$\mu_p(a + p^k\mathbb{Z}_p) = \frac{1}{p^k},$$

for all $a \in \mathbb{Z}_p$.

By abuse of notation, we will denote the product measure also by μ_p, where we remark that

$$\prod_{j=1}^{n} \mu_p(U_j) = \mu_p(U_1 \times \cdots \times U_n),$$

for $U_j \in \mathbb{Z}_p$ for all $j \in \{1, \ldots, n\}$.

Recall that by μ_∞ we denote the Lebesgue measure on $\mathbb{R}^d$ and by $M_\mathbb{Q}$ we denote the set of all places of $\mathbb{Q}$, *i.e.*, $M_\mathbb{Q} = \{\infty\} \cup \mathcal{P}$. We are now ready to state the main tool of this part of the thesis: the local to global principle.

Theorem 9.3.8 (Local to Global Principle, [91])**.** *Let d be a positive integer. Let $U_\infty \subset \mathbb{R}^d$, such that $\mathbb{R}_{\geq 0} \cdot U_\infty = U_\infty$ and $\mu_\infty(\partial(U_\infty)) = 0$. Let $s_\infty = \frac{1}{2^d}\mu_\infty(U_\infty \cap [-1,1]^d)$. For each prime p, let $U_p \subset \mathbb{Z}_p^d$, such that $\mu_p(\partial(U_p)) = 0$ and define $s_p = \mu_p(U_p)$. Define the following map*

$$\begin{aligned} P : \mathbb{Z}^d &\to 2^{M_\mathbb{Q}}, \\ a &\mapsto \{\nu \in M_\mathbb{Q} \mid a \in U_\nu\}. \end{aligned}$$

If the following is satisfied:

$$\lim_{M\to\infty} \bar{\rho}\left(\left\{a \in \mathbb{Z}^d \mid a \in U_p \text{ for some prime } p > M\right\}\right) = 0, \tag{9.3.1}$$

then:

i) $\sum_{\nu\in M_\mathbb{Q}} s_\nu$ *converges.*

ii) For $\mathcal{S} \subset 2^{M_\mathbb{Q}}$, $\rho(P^{-1}(\mathcal{S}))$ *exists, and defines a measure on* $2^{M_\mathbb{Q}}$.

iii) For each finite set $S \in 2^{M_\mathbb{Q}}$, *we have that*

$$\rho(P^{-1}(\{S\})) = \prod_{\nu\in S} s_\nu \prod_{\nu\notin S} (1 - s_\nu),$$

and if $\mathcal{S}$ *consists of infinite subsets of* $2^{M_\mathbb{Q}}$, *then* $\rho(P^{-1}(\mathcal{S})) = 0$.

To show that Condition (9.3.1) is satisfied, one can often apply the following useful lemma, that can be deduced from the result in [102].

Lemma 9.3.9 ([91], Lemma 2)**.** *Let d and M be positive integers. Let $f, g \in \mathbb{Z}[x_1, \ldots, x_d]$ be relatively prime. Define*

$$S_M(f,g) = \left\{ a \in \mathbb{Z}^d \mid f(a) \equiv g(a) \equiv 0 \mod p \text{ for some prime } p > M \right\},$$

then

$$\lim_{M \to \infty} \bar{\rho}(S_M(f,g)) = 0.$$

Remark 9.3.10. Observe that one can exchange in the statement of [91, Lemma 1] $\mathbb{R}_{\geq 0}$ with $\mathbb{R}_{>0}$. This modification is legitimate, since if the density of a set T was associated to U_∞, which is such that $U_\infty \cdot \mathbb{R}_{>0} = U_\infty$ but $U_\infty \cdot \mathbb{R}_{\geq 0} \neq U_\infty$, then one can define $\widetilde{U}_\infty = U_\infty \cup \{0\}$, which clearly satisfies $\widetilde{U}_\infty \cdot \mathbb{R}_{\geq 0} = \widetilde{U}_\infty$. Let $\widetilde{T}$, be the set, whose density is associated to $\widetilde{U}_\infty$, then $\rho(T) = \rho(\widetilde{T})$, since finite sets have density zero.

Observe that in Theorem 9.3.8 one could always choose the finite set S to be the empty set, which for our purposes will be convenient.

Corollary 9.3.11. *For all $\nu \in M_{\mathbb{Q}}$, let U_ν be chosen as in Theorem 9.3.8, corresponding to a finite set $S \in 2^{M_{\mathbb{Q}}}$. Let us define*

$$U'_\nu = \begin{cases} U_\nu^C & \nu \in S, \\ U_\nu & \nu \notin S, \end{cases}$$

and hence

$$s'_\nu = \begin{cases} 1 - s_\nu & \nu \in S, \\ s_\nu & \nu \notin S, \end{cases}$$

and define

$$\begin{aligned} P' : \mathbb{Z}^d &\to 2^{M_{\mathbb{Q}}}, \\ a &\mapsto \left\{ \nu \in M_{\mathbb{Q}} \mid a \in U'_\nu \right\}. \end{aligned}$$

Then we get

i) $\sum_{\nu \in M_{\mathbb{Q}}} s'_\nu$ *converges.*

ii) For $\mathcal{S} \subset 2^{M_{\mathbb{Q}}}$, $\rho(P'^{-1}(\mathcal{S}))$ exists and defines a measure on $2^{M_{\mathbb{Q}}}$.

iii) $\rho(P'^{-1}(\{\emptyset\})) = \prod_{\nu \in M_{\mathbb{Q}}} (1 - s'_\nu) = \rho(P^{-1}(\{S\}))$*, where P is the map as in Theorem 9.3.8.*

Proof. If $\nu \notin S$ then $U'_\nu = U_\nu$ and all the properties of U_ν in the beginning of Theorem 9.3.8 are trivially satisfied. Hence, let us only consider $\nu \in S$. If $\infty \in S$, then $U'_\infty = U_\infty^C$ and it holds that

a) $\mathbb{R}_{>0} \cdot U'_\infty = U'_\infty$,

b) $\mu_\infty(\delta(U'_\infty)) = 0$.

For a) we relied on Remark 9.3.10. We can assume by contradiction, that there exists a $s \in U'_\infty$, *i.e.*, $s \notin U_\infty$, and there exists a $\eta \in \mathbb{R}_{>0}$, such that $\eta \cdot s \notin U'_\infty$, *i.e.*, $\eta \cdot s \in U_\infty$. We know for every $r \in U_\infty$ that $\lambda \cdot r \in U_\infty \ \forall \lambda \in \mathbb{R}_{>0}$ hence, also for $r = \eta \cdot s$ and $\lambda = \eta^{-1}$. Thus, $\lambda \cdot r = s \in U_\infty$, which is a contradiction. For b) it is enough to observe, that for any subset T of a metric space its boundary $\partial(T)$ can be defined as the intersection of its closure and the closure of the complement, *i.e.*,

$$\partial(T) = \overline{T} \cap \overline{(T^C)} = \partial(T^C).$$

If a prime $p \in S$, then $U'_p = U_p^C$ and it holds that $\mu_p(\partial(U'_p)) = 0$ again by the definition of boundary. Condition (9.3.1) is satisfied, *i.e.*,

$$\lim_{M \to \infty} \bar{\rho}\left(\{a \in \mathbb{Z}^d \mid a \in U_p \text{ for some prime } p > M\}\right) = 0,$$

since if S is a finite set in $2^{M_{\mathbb{Q}}}$, then there exists an $M \in \mathbb{N}$ such that, for all $p > M$: $p \notin S$, thus for all $p > M : U'_p = U_p$ and (9.3.1) follows. The points i) and ii) follow directly from the Theorem 9.3.8 and to check that

$$\rho(P^{-1}(\{S\})) = \rho(P'^{-1}(\{\emptyset\}))$$

is straightforward. □

9.4 Strategy

The local to global principle of Theorem 9.3.8 can be applied on a target set T, if this set can be characterized through primes. Hence in general we have that $t \in T$, if t mod p satisifies some condition C.

Using Corollary 9.3.11, we can always set $S = \emptyset$. Thus, if we want to compute the density of the target set T, we want that $P^{-1}(\{S\}) = P^{-1}(\{\emptyset\}) = T$.

To assure this we will always choose U_p as a complement of the condition in p, that is given by the set T, *i.e.*,

$$U_p = \{x \in \mathbb{Z}_p \mid x \mod p \text{ does not verify condition } C\}.$$

The difficulties now lie in computing $s_p = \mu_p(U_p)$ and proving that the conditions from Theorem 9.3.8 are satisfied. To show that Condition (9.3.1) is verified, we usually use Lemma 9.3.9 and for the condition $\mu_p(\partial(U_p)) = 0$, it is enough to show that U_p is open and closed, and thus $\partial(U_p) = \emptyset$. For this, we will heavily rely on Proposition 9.3.6. Note that we will always choose $U_\infty = \emptyset$, since we can characterize the target sets completely via $\mathbb{Z}_p$. Note, that with this choice all conditions on U_∞ are satisfied. In fact,

$$s_\infty = \frac{1}{2^1}\mu_\infty\left(U_\infty \cap [-1, 1]\right) = 0.$$

In addition, we get that $\mathbb{R}_{\geq 0} \cdot \emptyset = \emptyset$ and $\mu_\infty(\partial(\emptyset)) = \mu_\infty(\emptyset) = 0$. We will hence only focus on the choice of U_p in the subsequent proofs.

Chapter 10

Density Computations over $\mathbb{Z}$

In this chapter we compute the densities of the four target sets over the integers. We start with the coprime pairs, which can be considered a toy example. The set of coprime m-tuples usually follows in the exact same manner as the coprime pairs, thus we will only give the result and point out the differences in the proof. We then compute the density of non-monic Eisenstein polynomials and give the result for the case of monic polynomials and state the differences in the proof. Finally, we compute the density of rectangular unimodular matrices.

10.1 Coprime Pairs

This is a well-known result due to Mertens [75] in the 1870's and has later been independently solved by Césaro [33].

The question they wanted to answer, was of course stated a little differently, *i.e.*:

How likely is it that two randomly chosen integers are coprime?

The question can hence be reformulated to

What is the natural density of coprime pairs over the integers?

Theorem 10.1.1 (Density of Coprime Pairs over $\mathbb{Z}$, [33, 75])**.** *Let the set of coprime pairs over $\mathbb{Z}$ be denoted by C,* i.e.*,*

$$C = \{(a_1, a_2) \in \mathbb{Z}^2 \mid \gcd(a_1, a_2) = 1\}.$$

Then,

$$\rho(C) = \frac{1}{\zeta(2)} = \frac{6}{\pi^2}.$$

Up to our best knowledge, this is the only reference giving the proof of this theorem using the local to global principle of Theorem 9.3.8.

Proof. For $p \in \mathcal{P}$ choose $U_p = (p\mathbb{Z}_p)^2$. By Proposition 9.3.6, we have that U_p is open and closed and thus all conditions on U_p are verified. Hence, we can compute s_p as

$$s_p = \mu_p(U_p) = \frac{1}{p^2}.$$

Now we have to show that Condition (9.3.1) in Theorem 9.3.8 is satisfied, *i.e.*,

$$\lim_{M\to\infty} \overline{\rho}\left(\{(a_1,a_2) \in \mathbb{Z}^2 \mid (a_1,a_2) \in U_p \text{ for some } p > M \text{ prime}\right) = 0.$$

For this we use Lemma 9.3.9. We choose the auxiliary functions $f(x_1,x_2) = x_1$ and $g(x_1,x_2) = x_2$, which are clearly coprime. Now, the set S_M of f and g is given as

$$\begin{aligned} S_M(f,g) &= \{(a_1,a_2) \in \mathbb{Z}^2 \mid p \mid f(a_1,a_2) = a_1 \\ &\qquad\qquad \text{and } p \mid g(a_1,a_2) = a_2 \text{ for some prime } \ p > M\} \\ &= \{(a_1,a_2) \in \mathbb{Z}^2 \mid (a_1,a_2) \in U_p \text{ for some prime } \ p > M\}. \end{aligned}$$

Thus, using Lemma 9.3.9 we have that

$$\lim_{M\to\infty} \overline{\rho}\left(\{(a_1,a_2) \in \mathbb{Z}^2 \mid (a_1,a_2) \in U_p \text{ for some prime } \ p > M\}\right) = 0.$$

As discussed in Corollary 9.3.11, we can choose $S = \emptyset$, since

$$P^{-1}(\{\emptyset\}) = C = \{(a_1,a_2) \in \mathbb{Z}^2 \mid \gcd(a_1,a_2) = 1\}.$$

In fact, if $a \in \mathbb{Z}^2$ is in U_p for any prime p, then p divides a_1 as well as a_2, hence they are not coprime. As a last step, we have to compute $\rho(P^{-1}(\{\emptyset\}))$.

$$\begin{aligned} \rho(P^{-1}(\{\emptyset\})) &= \prod_{\nu\in\emptyset} s_\nu \prod_{\nu\notin\emptyset}(1-s_\nu) \\ &= (1-s_\infty)\prod_{p\in\mathcal{P}}(1-s_p) \\ &= 1 \cdot \prod_{p\in\mathcal{P}}\left(1-\frac{1}{p^2}\right) \\ &= \frac{1}{\zeta(2)} = \frac{6}{\pi^2}. \end{aligned}$$

This concludes the proof. □

Coprime m-tuples

This result is due to Nymann [85] in the 1970's, hence this was answered only 100 years after the result of Mertens and Césaro.

Let $m > 2$ be an integer. Let us denote by C_m the set of coprime m-tuples over $\mathbb{Z}$. Clearly, we can use the following characterization of coprime m-tuples

$$\begin{aligned} C_m &= \{(a_1,\ldots,a_m) \in \mathbb{Z}^m \mid \gcd(a_1,\ldots,a_m) = 1\} \\ &= \{(a_1,\ldots,a_m) \in \mathbb{Z}^m \mid I(a_1,\ldots,a_m) = \mathbb{Z}\}. \end{aligned}$$

Theorem 10.1.2 (Density of Coprime m-tuples over $\mathbb{Z}$, [85])**.** *Let the set of coprime m-tuples over $\mathbb{Z}$ be denoted by C_m,* i.e.*,*

$$C_m = \{(a_1, \ldots, a_m) \in \mathbb{Z}^m \mid I(a_1, \ldots, a_m) = \mathbb{Z}\}.$$

Then,

$$\rho(C_m) = \frac{1}{\zeta(m)}.$$

The proof of this theorem, using the local to global principle, is exactly in the same manner as the proof for the density of coprime pairs. Thus, we will not repeat it here, but rather point out the differences.

The most important difference to the proof of coprime pairs, is that we choose $U_p = (p\mathbb{Z}_p)^m$. Hence, s_p is given by

$$s_p = \frac{1}{p^m}.$$

To verify Condition (9.3.1), we use the same coprime functions, observing that the set of all $(a_1, \ldots, a_m)$ which are in U_p for $p > M$, is contained in S_M and hence also satisfying (9.3.1).

10.2 Eisenstein Polynomials

When considering Eisenstein polynomials, one differs between monic polynomials and non-monic. The density result of the monic case is due to Dubickas [42] in 2003, and the non-monic case is due to Heyman and Shparlinski [52], in 2013.

We cover here the non-monic case in detail, and then point out the differences for the monic case.

Let $f \in \mathbb{Z}[x]$ be a non-monic polynomial of degree d, *i.e.*,

$$f(x) = \sum_{i=0}^{d} a_i x^i = a_0 + a_1 x + \cdots + a_d x^d.$$

Hence we can identify f with the vector $(a_0, \ldots, a_d) \in \mathbb{Z}^{d+1}$, via the standard basis $\{1, x, \ldots, x^d\}$. We call this vector the associated vector or corresponding vector to f.

Definition 10.2.1 (Eisenstein Polynomial)**.** Let $f \in \mathbb{Z}[x]$ of degree d with the associated vector $(a_0, \ldots, a_d) \in \mathbb{Z}^{d+1}$. Then, we call f an *Eisenstein polynomial*, if $p^2 \nmid a_0, p \nmid a_d$ and for all $i < d$, we have that $p \mid a_i$, for some prime p.

Note that the criterion of Eisenstein is a sufficient condition for irreducibility, but not a necessary condition.

For example, the polynomial $5 + 10x + 2x^3$ satisfies the criterion of Eisenstein for the prime $p = 5$, and is thus irreducible. Whereas, the polynomial $1 + x^2$, which is clearly irreducible, does not satisfy the criterion of Eisenstein, as there is no prime dividing the constant term, which is one.

Let us denote the set of Eisenstein polynomials of degree d by E_d. Thus, we can use the following characterization of Eisenstein polynomials:

$$E_d = \{(a_0, \ldots, a_d) \in \mathbb{Z}^{d+1} \mid p^2 \nmid a_0, p \nmid a_d \text{ and } p \mid a_i \ \forall\ i < d \text{ for some prime } p\}.$$

Theorem 10.2.2 (Density of Eisenstein Polynomials over $\mathbb{Z}$, [52])**.** *Let the set of Eisenstein polynomials of degree d over $\mathbb{Z}$ be denoted by E_d. Then,*

$$\rho(E_d) = 1 - \prod_{p\in\mathcal{P}} \left(1 - \frac{(p-1)^2}{p^{d+2}}\right).$$

The idea of the proof using the local to global principle was already given in the thesis of Micheli [77].

Proof. Observe that it is easier to compute the density of the complement, *i.e.*, the polynomials of degree d, which are not Eisenstein for any prime, namely $\mathbb{Z}^{d+1} \setminus E_d = E_d^C$. Note that we can then easily compute the density of E_d, since $\rho(E_d) = 1 - \rho(E_d^C)$.

For all primes p, we choose

$$U_p = (p\mathbb{Z}_p \setminus p^2\mathbb{Z}_p) \times (p\mathbb{Z}_p)^{d-1} \times (\mathbb{Z}_p \setminus p\mathbb{Z}_p).$$

Thus, s_p is given by

$$s_p = \mu_p(U_p) = \left(\frac{1}{p} - \frac{1}{p^2}\right)\frac{1}{p^{d-1}}\left(1 - \frac{1}{p}\right) = \frac{(p-1)^2}{p^{d+2}}.$$

Note that all conditions on U_p are satisfied, since by Proposition 9.3.6 we have that U_p is closed and open, hence $\partial(U_p) = \emptyset$ and $\mu_p(\emptyset) = 0$.

To show that Condition (9.3.1) is verified, we use Lemma 9.3.9. We again choose the auxiliary functions $f(x_1, \ldots, x_{d+1}) = x_1$ and $g(x_1, \ldots, x_{d+1}) = x_2$, which are clearly coprime. Note, that the set A_M, being

$$A_M = \{(a_0, \ldots, a_d) \in \mathbb{Z}^{d+1} \mid (a_0, \ldots, a_d) \in U_p \text{ for some } p > M \text{ prime}\}$$

is a subset of $S_M(f, g)$. Using Lemma 9.3.9, we have that

$$\lim_{M\to\infty} \overline{\rho}(S_M(f, g)) = 0.$$

And since $A_M \subset S_M(f, g)$, it follows that

$$\lim_{M\to\infty} \overline{\rho}(A_M) \leq \lim_{M\to\infty} \overline{\rho}(S_M(f, g)) = 0,$$

and hence $\lim_{M\to\infty} \overline{\rho}(A_M) = 0$. Note, that for $S = \emptyset$, we get

$$\begin{aligned} P^{-1}(\{\emptyset\}) = E_d^C = \{(a_0, \ldots, a_d) \in \mathbb{Z}^{d+1} \mid p^2 \mid a_0 \text{ or } p \mid a_d \text{ or} \\ \exists i < d \text{ with } p \nmid a_i \text{ for any prime } p\}. \end{aligned}$$

In fact, if $a \in \mathbb{Z}^{d+1}$ is in U_p for any prime p, then p divides a_i for all $i < d$, as well as, $p^2 \nmid a_0$ and $p \nmid a_d$, hence the associated polynomial is Eisenstein for this prime p. Finally, we compute $\rho(P^{-1}(\{\emptyset\}))$.

$$\begin{aligned} \rho(P^{-1}(\{\emptyset\})) &= \prod_{\nu\in\emptyset} s_\nu \prod_{\nu\notin\emptyset}(1 - s_\nu) \\ &= (1 - s_\infty)\prod_{p\in\mathcal{P}}(1 - s_p) \\ &= 1 \cdot \prod_{p\in\mathcal{P}} \left(1 - \frac{(p-1)^2}{p^{d+2}}\right). \end{aligned}$$

We can conclude this proof, since

$$\rho(E_d) = 1 - \rho(E_d^C) = 1 - \rho(P^{-1}(\{\emptyset\})) = 1 - \prod_{p \in \mathcal{P}} \left(1 - \frac{(p-1)^2}{p^{d+2}}\right).$$

□

Monic Case

In the monic case, we have that $a_d = 1$. Thus, we can identify f of degree d with an element in $\mathbb{Z}^d$, instead of $\mathbb{Z}^{d+1}$.

Recall, the definition of monic Eisenstein polynomials:

Definition 10.2.3 (Monic Eisenstein Polynomial)**.** Let $f \in \mathbb{Z}[x]$ be monic of degree d, with the associated vector $(a_0, \ldots, a_{d-1}) \in \mathbb{Z}^d$. Then, we call f a *monic Eisenstein polynomial*, if $p^2 \nmid a_0$ and for all $i < d$, we have that $p \mid a_i$, for some prime p.

Hence we can use the following characterization of Eisenstein polynomials:

$$\begin{aligned}&\{f \in \mathbb{Z}[x] \mid f \text{ monic and Eisenstein and } \deg(f) = d\}\\ &= \{(a_0, \ldots, a_{d-1}) \in \mathbb{Z}^d \mid p^2 \nmid a_0 \text{ and } p \mid a_i \ \forall\ i < d \text{ for some prime } p\}.\end{aligned}$$

Theorem 10.2.4 (Density of Monic Eisenstein Polynomials over $\mathbb{Z}$, [42])**.** *Let the set of monic Eisenstein polynomials of degree d over $\mathbb{Z}$ be denoted by M_d,* i.e.*,*

$$M_d = \{(a_0, \ldots, a_{d-1}) \in \mathbb{Z}^d \mid p^2 \nmid a_0 \text{ and } p \mid a_i \ \forall\ i < d \text{ for some prime } p\}.$$

Then,

$$\rho(M_d) = 1 - \prod_{p \in \mathcal{P}} \left(1 - \frac{p-1}{p^{d+1}}\right).$$

The proof has a similar structure as the non-monic case, we hence only indicate their differences.

For all primes p, one chooses

$$U_p = (p\mathbb{Z}_p \setminus p^2\mathbb{Z}_p) \times (p\mathbb{Z}_p)^{d-1},$$

and hence

$$s_p = \left(\frac{1}{p} - \frac{1}{p^2}\right) \frac{1}{p^{d-1}} = \frac{p-1}{p^{d+1}}.$$

To show that all conditions on U_p are verified, in particular Condition (9.3.1) is exactly the same as in the non-monic case.

10.3 Rectangular Unimodular Matrices

Let us define rectangular unimodular matrices.

Definition 10.3.1. Let $\mathcal{R}$ be a domain and $n < m \in \mathbb{N}$. Let $M \in \mathrm{Mat}_{n\times m}(\mathcal{R})$. M is said to be *rectangular unimodular*, if there exist $m-n$ rows in $\mathcal{R}^m$, such that when adjoining these rows to M the resulting $m \times m$ matrix $\widetilde{M}$ is invertible, *i.e.*, $\det(\widetilde{M})$ is a unit in $\mathcal{R}$.

Observe that we exclude the case $n = m$, since then we get the set of invertible matrices over $\mathbb{Z}$, for which we know that it has density zero, *i.e.*,

$$\rho(\mathrm{GL}_n(\mathbb{Z})) = 0.$$

One can imagine rectangular unimodular matrices as the $\mathbb{Z}$-analogue of full rank matrices. Since over $\mathbb{Z}$, we do not have the notion of rank, we can characterize them as the matrices, which have full rank modulo p for any prime p.

Example 10.3.2.

- An example for a rectangular unimodular matrix is

$$A = \begin{pmatrix} 0 & 1 \end{pmatrix},$$

since we can extend this matrix to

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

Observe that for any prime p the matrix $A \mod p$ has full rank 1.

- An example for a rectangular non-unimodular matrix is

$$B = \begin{pmatrix} 2 & 2 \end{pmatrix},$$

since whatever we adjoin as row to B, the determinant of B will be divisible by 2 and hence the extended matrix can not be invertible over $\mathbb{Z}$. Observe that with our characterization through primes, $B \mod 2$ does not have full rank.

Thus, the characterization that we use to compute the density of rectangular unimodular matrices over $\mathbb{Z}$, is as follows: for $n < m$ positive integers the set of all unimodular rectangular $n \times m$ matrices over the integers is given by

$$U_{n\times m} = \{A \in \mathrm{Mat}_{n\times m}(\mathbb{Z}) \mid A \mod p \text{ has full rank } n \text{ for any prime } p\}.$$

In [73] Maze, Rosenthal and Wagner computed the density of rectangular unimodular matrices over the integers. Unfortunately, the proof of [73] Proposition 1 is flawed as the inequality (2) is wrong, *i.e.*, it is not true that

$$\rho\left(\bigcup_{i\in I} A_i\right) \leq \sum_{i\in I} \rho(A_i),$$

for I a countable set. A counterexample is simply obtained by setting $A_i = \{i\}$ for all $i \in \mathbb{Z}$, since then

$$\rho\left(\bigcup_{i\in\mathbb{Z}} \{i\}\right) = \rho(\mathbb{Z}) = 1 \not\leq \sum_{i\in\mathbb{Z}} \rho(\{i\}) = 0.$$

We should note that this is correct, for example, for finite unions (which is not the case in the proof of [73] Proposition 1.)

The density result nevertheless is correct. The proof of this theorem was fixed and can be found in greater generality in [78].

Theorem 10.3.3 (Density of Rectangular Unimodular Matrices over $\mathbb{Z}$, [78])**.** *Let the set of unimodular rectangular $n \times m$ matrices over the integers be denoted by $U_{n\times m}$,* i.e.*,*

$$U_{n\times m} = \{A \in \mathrm{Mat}_{n\times m}(\mathbb{Z}) \mid A \mod p \text{ has full rank } n \text{ for any prime } p\}.$$

Then,

$$\rho(U_{n\times m}) = \prod_{i=0}^{n-1} \frac{1}{\zeta(m-i)}.$$

We give the proof by applying the local to global principle from Theorem 9.3.8. Note that this proof is not exactly as in [78], as we only consider here the rational integers.

Proof. Let us first define the map π_p as

$$\begin{aligned} \pi_p : \mathbb{Z}_p^{n\times m} &\to \mathbb{F}_p^{n\times m}, \\ A &\mapsto A \mod p. \end{aligned}$$

Let $\mathcal{L}_p$ be the set of $n \times m$ matrices over $\mathbb{F}_p$ having full rank n, *i.e.*,

$$\mathcal{L}_p = \{A \in \mathbb{F}_p^{n\times m} \mid \mathrm{rk}(A) = n\}.$$

Hence the size of $\mathcal{L}_p$ is given by

$$\mid \mathcal{L}_p \mid = \prod_{i=0}^{n-1} \left(p^m - p^i\right).$$

Now observe that, if $X \in \mathbb{Z}_p^{n\times m}$ is such that $\pi_p(X) \in \mathcal{L}_p$, then also $X + p\mathbb{Z}_p^{nm}$ is such that $\pi_p(X + p\mathbb{Z}_p^{nm}) \in \mathcal{L}_p$.

For all primes p , let A_p be the set of all matrices over $\mathbb{Z}_p$ such that, modulo p they have full rank, *i.e.*,

$$A_p = \left\{X \in \mathbb{Z}_p^{n\times m} \mid \pi_p(X) \in \mathcal{L}_p\right\} + p\mathbb{Z}_p^{nm},$$

and choose $U_p = A_p^C$.

Note that we can write A_p as

$$A_p = \bigcup_{A\in\mathcal{L}_p} (\pi_p^{-1}(A) + p\mathbb{Z}_p^{nm})$$

since $(\pi_p^{-1}(A) + p\mathbb{Z}_p^{nm})$ is closed and open, and the union is finite, A_p is closed and open as well, and thus so is U_p.

Observe that

$$\begin{aligned}\mu_p(A_p) &= \mu_p(p\mathbb{Z}_p^{nm}) \mid \mathcal{L}_p \mid \\ &= \frac{1}{p^{nm}} \prod_{i=0}^{n-1} \left(p^m - p^i\right) \\ &= \prod_{i=0}^{n-1} \frac{1}{p^m} \left(p^m - p^i\right) \\ &= \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}}\right).\end{aligned}$$

Then, $s_p = \mu_p\left(U_p\right) = 1 - \mu_p(A_p)$ is given by

$$s_p = 1 - \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}}\right).$$

To show that Condition (9.3.1) in Theorem 9.3.8 is satisfied, *i.e.*,

$$\lim_{M\to\infty} \overline{\rho}\left(\{X \in \mathbb{Z}^{n\times m} \mid X \in U_p \text{ for some } p > M \text{ prime}\}\right) = 0,$$

we use Lemma 9.3.9. Let us denote for $A \in \mathbb{Z}^{n\times m}$ its $n \times n$ minors by A_i for $i \in \{1, \ldots, \binom{m}{n}\}$. Then, $A \in U_p$ is equivalent to

$$I\left(A_1, \ldots, A_{\binom{m}{n}}\right) \subseteq (p),$$

and hence,

$$p \mid \gcd\left(A_1, \ldots, A_{\binom{m}{n}}\right).$$

Thus, we can choose the auxiliary functions in $\mathbb{Z}[x_1, \ldots, x_{nm}]$, such that f gives the first minor and g gives the second minor, *i.e.*,

$$\begin{aligned} f : \mathbb{Z}[x_1, \ldots x_{nm}] &\to \mathbb{Z}, \\ A &\mapsto A_1, \text{ and} \\ g : \mathbb{Z}[x_1, \ldots x_{nm}] &\to \mathbb{Z}, \\ A &\mapsto A_2. \end{aligned}$$

Note that f and g are irreducible and since they are determinants and since they involve different variables, they are clearly coprime. Now, the set S_M of f and g is given as

$$\begin{aligned} S_M(f,g) = \{A \in \mathbb{Z}^{n\times m} \mid & p \mid f(A) = A_1 \\ & \text{and } p \mid g(A) = A_2 \text{ for some prime } \ p > M\} \\ \supset \{A \in \mathbb{Z}^{n\times m} \mid & A \in U_p \text{ for some } p > M \text{ prime}\} = B_M. \end{aligned}$$

Thus, by Lemma 9.3.9 we have that

$$\lim_{M\to\infty} \overline{\rho}(S_M(f,g)) = 0.$$

Since $B_M \subset S_M(f,g)$, it follows that

$$\lim_{M\to\infty} \overline{\rho}(B_M) \leq \lim_{M\to\infty} \overline{\rho}(S_M(f,g)) = 0,$$

and hence $\lim_{M\to\infty} \overline{\rho}(B_M) = 0$. We choose $S = \emptyset$, since

$$P^{-1}(\{\emptyset\}) = U_{n\times m} = \left\{ A \in \mathbb{Z}^{n\times m} \mid I\left(A_1, \ldots, A_{\binom{m}{n}}\right) = (p) \right\}.$$

In fact, if $A \in \mathbb{Z}^{n\times m}$ is in U_p for any prime p, then p divides A_i for all $i \in \{1, \ldots, \binom{m}{n}\}$. Thus, A is not rectangular unimodular. As a last step, we have to compute $\rho(P^{-1}(\{\emptyset\}))$, which is the same as $\rho(U_{n\times m})$.

$$\begin{aligned}
\rho(P^{-1}(\{\emptyset\})) &= \prod_{\nu\in\emptyset} s_\nu \prod_{\nu\notin\emptyset} (1 - s_\nu) \\
&= (1 - s_\infty) \prod_{p\in\mathcal{P}} (1 - s_p) \\
&= 1 \cdot \prod_{p\in\mathcal{P}} \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}}\right) \\
&= \prod_{i=0}^{n-1} \frac{1}{\zeta(m-i)}.
\end{aligned}$$

Hence we get the claim. □

10.4 Generalization to Algebraic Integers

A generalization of Poonen and Stoll's local to global principle to algebraic integers over a number field was proposed in [28].

However, a more straightforward technique to compute the densities over algebraic integers, using the original local to global principle, was provided in [78] in collaboration with Giacomo Micheli.

In the mentioned article the density of rectangular unimodular $n \times m$ matrices over the algebraic integers was computed. Thus, this article directly provides the density results for coprime m-tuples over the algberaic integers (by setting $n = 1$) and for coprime pairs (by setting $n = 1$ and $m = 2$).

Although, we will not cover the densities over algebraic integers in full detail in this thesis, we still want to provide here the main idea and the results.

First of all, recall that a number field K is a finite field extension of $\mathbb{Q}$, where we denote by the index $k = [K : \mathbb{Q}]$ the degree of the extension.

Furthermore, the integral closure of $\mathbb{Z}$ in K is called the ring of algebraic integers and denoted by $\mathcal{O}_K$. Observe that $\mathcal{O}_K$ is isomorphic to $\mathbb{Z}^k$ as $\mathbb{Z}$-modules, and thus $\mathcal{O}_K$ has an integral basis $\{e_1, \dots, e_k\}$, which is such that each element $a \in \mathcal{O}_K$ can be written as

$$a = \sum_{i=1}^{k} a_i e_i,$$

for some $(a_1, \dots, a_k) \in \mathbb{Z}^k$. Let us denote this isomorphism by $\mathbb{E}$, *i.e.*,

$$\begin{aligned} \mathbb{E} : \mathbb{Z}^k &\to \mathcal{O}_K \\ (a_1, \dots, a_k) &\mapsto \sum_{i=1}^{k} a_i e_i. \end{aligned}$$

One of the most important properties of algebraic integers, is that they form a Dedekind domain, and thus have a unique factorization of ideals. Note also, that for each prime ideal $\mathfrak{p}$ in $\mathcal{O}_K$ there exists a rational prime p, such that $\mathfrak{p}$ intersects $\mathbb{Z}$ in the ideal $p\mathbb{Z}$. We then say that $\mathfrak{p}$ is lying over p and write $\mathfrak{p} \mid p$. On the other hand, the ideal generated by p has a unique factorization into prime ideals $\mathfrak{p}_i \subset \mathcal{O}_K$, *i.e.*,

$$p\mathcal{O}_K = \mathfrak{p}_1^{\ell_1} \cdots \mathfrak{p}_r^{\ell_r},$$

and for all $i \in \{1, \dots, r\}$ the prime ideals $\mathfrak{p}_i$ lie over p. Furthermore, we have that the residue field $\mathcal{O}_K/\mathfrak{p}$ is isomorphic to $\mathbb{F}_{p^\ell}$ as fields, for some ℓ. In fact, this ℓ is called inertia degree and denoted by $\deg(\mathfrak{p})$.

Let $k = [K : \mathbb{Q}]$ and $E = \{e_1, \dots, e_k\}$ be an integral basis of $\mathcal{O}_K$. Let us denote by $[-H, H[_E$ the $\mathcal{O}_K$-analogue of $[H, H[$, *i.e.*,

$$[-H, H[_E = \left\{ \sum_{i=1}^{k} a_i e_i \mid a_i \in [-H, H[\cap \mathbb{Z} \right\}.$$

The density of a set $T \subseteq \mathcal{O}_K^d$ is then defined as

$$\rho_E(T) = \lim_{H \to \infty} \frac{|\, T \cap [-H, H[_E^d|}{(2H)^{dk}},$$

if the limit exists.

Observe that a priori the density $\rho_E(T)$ depends on the choice of the integral basis E. If $\rho_E(T)$ is independent of the choice of E, we will denote the density again by $\rho(T)$.

Lastly, we want to introduce the Dedekind zeta function over K. Let K be a number field and $\mathcal{O}_K$ be its algebraic integers. Then, the Dedekind zeta function is defined as

$$\zeta_K(s) = \prod_{p \in \mathcal{P}} \prod_{\mathfrak{p} \mid p} \left(1 - \frac{1}{p^{\deg(\mathfrak{p})s}}\right)^{-1}.$$

To use the local to global principle we need functions that bring us from the p-adic integers to the algebraic integers.

Let $k = [K : \mathbb{Q}]$ and $\{e_1, \ldots, e_k\}$ be an integral basis of $\mathcal{O}_K$. Let us first consider the usual reduction modulo a rational prime p

$$\pi_p : \mathbb{Z}_p \to \mathbb{F}_p,$$

which is clearly surjective. Secondly, we consider the map

$$\begin{aligned} \mathbb{E}_p : \mathbb{F}_p^k &\to \mathcal{O}_K/(p), \\ (a_1, \ldots, a_k) &\mapsto \sum_{i=1}^{k} a_i e_i, \end{aligned}$$

which is a bijection. And lastly, the natural surjective map

$$\psi_p : \mathcal{O}_K/(p) \to \prod_{\mathfrak{p}|p} (\mathcal{O}_K/\mathfrak{p}) .$$

Here it might be helpful to observe that

$$\mathcal{O}_K/(p) \cong \mathcal{O}_K/(\mathfrak{p}_1^{\ell_1} \cdots \mathfrak{p}_r^{\ell_r}) \xrightarrow{\psi_p} \mathcal{O}_K/(\mathfrak{p}_1 \cdots \mathfrak{p}_r) \cong \prod_{\mathfrak{p}|p} \mathcal{O}_K/\mathfrak{p}.$$

We get the following composition of maps $F_p = \psi_p \circ \mathbb{E}_p \circ \pi_p$:

$$\mathbb{Z}_p^k \xrightarrow{\pi_p} \mathbb{F}_p^k \xrightarrow{\mathbb{E}_p} \mathcal{O}_K/(p) \xrightarrow{\psi_p} \prod_{\mathfrak{p}|p} \mathcal{O}_K/\mathfrak{p}.$$

Using this map and its componentwise extension to d-tuples, we computed in [78] the density of rectangular unimodular matrices over $\mathcal{O}_K$.

Theorem 10.4.1 (Density of Rectangular Unimodular Matrices, [78])**.** *Let n and m be positive integers such that $n < m$ and K be an algebraic number field. The density of $n \times m$ rectangular unimodular matrices over $\mathcal{O}_K$ is*

$$\prod_{i=0}^{n-1} \frac{1}{\zeta_K(m-i)},$$

where ζ_K denotes the Dedekind zeta function of K.

The density result for coprime m-tuples, and thus also for coprime pairs, over the algebraic integers was already computed in [44], using a different technique. Nevertheless, the proof given in [78] can be adapted to the cases $n = 1$ and $m = 2$, providing hence a proof of the results through the local to global principle and the technique stated here.

Theorem 10.4.2 (Density of Coprime m-tuples over $\mathcal{O}_K$, [44])**.** *Let the set of coprime m-tuples over $\mathcal{O}_K$ be denoted by $C_m(K)$,* i.e.*,*

$$C_m(K) = \{(a_1, \ldots, a_m) \in \mathcal{O}_K^m \mid I(a_1, \ldots, a_m) = \mathcal{O}_K\}.$$

Then,

$$\rho(C_m(K)) = \frac{1}{\zeta_K(m)},$$

where ζ_K denotes the Dedekind zeta function over K.

Corollary 10.4.3 (Density of Coprime Pairs over $\mathcal{O}_K$, [44])**.** *Let the set of coprime pairs over $\mathcal{O}_K$ be denoted by $C(K)$,* i.e.*,*

$$C(K) = \{(a_1, a_2) \in \mathcal{O}_K^2 \mid I(a_1, a_2) = \mathcal{O}_K\}.$$

Then,

$$\rho(C(K)) = \frac{1}{\zeta_K(2)},$$

where ζ_K denotes the Dedekind zeta function over K.

Interestingly, the resulting densities over $\mathcal{O}_K$ only differ from the densities over $\mathbb{Z}$ in the zeta function, *i.e.*, while over $\mathbb{Z}$ we use the Riemann zeta function, over $\mathcal{O}_K$ we use the Dedekind zeta function. More in detail, one expects the following. Let $T \subseteq \mathcal{O}_K^d$ be such that the density of T exists and does not depend on the integral basis. If in addition the density of T can be computed through the described technique and over the integers we have the following density

$$\rho(\mathbb{E}^{-1}(T)) = \prod_{p \in \mathcal{P}} f(p),$$

where $f(p) = (1 - s_p)$ since we use the local to global principle, then we expect that

$$\rho(T) = \prod_{p \in \mathcal{P}} \prod_{\mathfrak{p} \mid p} f(p^{\deg(\mathfrak{p})}).$$

The careful reader might note, that we are missing a set of interest, namely the Eisenstein polynomials. In fact, computing the density of Eisenstein polynomials over the algebraic integers is an ongoing project in collaboration with Simran Tinani.

Chapter 11

Mean and Variance

11.1 Preliminaries

Seeing the natural density as the $\mathbb{Z}$-analogue of a uniform probability distribution, one might also ask for the definitions of the mean and the variance corresponding to the density. Up to our best knowledge, this was first introduced in [70].

For this we have a target set $T \subset \mathbb{Z}^d$, which is such that T can be characterized through a condition C_p, depending on a prime p, *i.e.*,

$$T = \{a \in \mathbb{Z}^d \mid \exists p \ a \text{ satisfies } C_p\}.$$

Let us assume, that the natural density of T exists and is nonzero. Let us introduce the function ψ as

$$\begin{aligned} \psi : \mathbb{Z}^d &\to \mathbb{N}, \\ a &\mapsto \mid \{p \in \mathcal{P} \mid a \text{ satisfies } C_p\} \mid . \end{aligned}$$

For a positive integer H, let us denote by $T(H)$ the elements in T, which are bounded by H, *i.e.*,

$$T(H) = T \cap [-H, H[^d.$$

Then we can define the mean and the variance corresponding to the natural density.

Definition 11.1.1. Let $T \subset \mathbb{Z}^d$, be characterized through the condition C_p, such that $\rho(T) \neq 0$ exists. Then, we define the *mean of* ψ as

$$\mu = \lim_{H \to \infty} \frac{\sum\limits_{a \in T(H)} \psi(a)}{\mid T(H) \mid},$$

if it exists. Further, we define the *variance of* ψ as

$$\sigma^2 = \lim_{H \to \infty} \frac{\sum\limits_{a \in T(H)} (\psi(a) - \mu)^2}{\mid T(H) \mid},$$

if it exists.

One can think of this definition as the expected value and the variance of the "random variable" that counts for how many primes an element is in T, *i.e.*, satisfying the condition C_p.

For example, the analogue question to the natural density, where we asked how likely it is that two randomly chosen integers are coprime, would be

How many primes on average divide two randomly chosen integers?

We will now introduce and recall some important definitions and notation that will be needed for the computations of the mean and the variance.

Recall that a real valued function f is called multiplicative, if for $\gcd(m,n) = 1$, we have $f(mn) = f(m)f(n)$. If this holds for any m and n, then we call the function completely multiplicative.

For a natural number $n > 1$, let us write n in its prime factorization, as

$$n = p_1^{\ell_1} \cdots p_r^{\ell_r}.$$

Then, we can define $\omega(n)$ to be the number of distinct prime factors of n, *i.e.*, $\omega(n) = r$ and we also set $\omega(1) = 0$.

We denote by $\mu(n)$ the Möbius function, *i.e.*,

$$\mu(n) = \begin{cases} (-1)^{\omega(n)} & \text{if } n \text{ is squarefree,} \\ 0 & \text{else,} \end{cases}$$

or equivalently,

$$\mu(n) = \begin{cases} (-1)^r & \text{if } \ell_i = 1 \ \forall i \in \{1, \dots r\}, \\ 0 & \text{else.} \end{cases}$$

Observe that 1 is considered squarefree and thus, $\mu(1) = 1$ and $\mu(n)$ is a multiplicative function. Note that the Möbius function $\mu(n)$ should not be confused with the mean μ, as the mean does not take integers as input.

We denote by $\varphi(n)$ the Euler totient function, *i.e.*,

$$\varphi(n) = \mid (\mathbb{Z}/n\mathbb{Z})^* \mid .$$

Hence $\varphi(n)$, counts the natural numbers, smaller than n and coprime to n. A formula for $\varphi(n)$ is given by

$$\varphi(n) = \prod_{i=1}^{r} p_i^{\ell_i - 1}(p_i - 1).$$

Also, the Euler totient function is a multiplicative function. Further, due to the Euler product, we have the following identity. Let f be a multiplicative function, then

$$\sum_{s \geq 1} \mu(s) f(s) = \prod_{p \in \mathcal{P}} (1 - f(p)).$$

Let us recall here the Möbius inversion formula: let f and g be arithmetic functions, *i.e.*, having as domain $\mathbb{Z}$ and as codomain the complex numbers $\mathbb{C}$. Let g be such that

$$g(n) = \sum_{s|n} f(s),$$

for all $n \geq 1$, then the Möbius inversion formula states that

$$f(n) = \sum_{s|n} \mu(s) g\left(\frac{n}{s}\right),$$

for all $n \geq 1$.

We will also heavily rely on the Landau-symbols in this chapter. For this, let f and g be real valued functions, we say that

$$f(n) = O(g(n)),$$

if $\exists k > 0 \exists n_0 \forall n > n_0$ we have that $\mid f(n) \mid \leq kg(n)$, or equivalently, if g is positive,

$$\limsup_{n\to\infty} \frac{\mid f(n) \mid}{g(n)} < \infty.$$

Intuitively, this means f does not grow faster than g. For example $(n+1)^2 = n^2 + O(n)$.

Similarly, we say that

$$f(n) = o(g(n)),$$

if $\forall k > 0 \exists n_0 \forall n > n_0$ we have that $f(n) \leq kg(n)$, or equivalently, if g is positive,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$

Intuitively, this means g grows much faster than f. For example $\frac{1}{n} = o(1)$.

The difference between the small-o and the big-O, becomes clear in the following example: $2x^2 = O(x^2)$, since $2x^2$ does not grow faster than x^2, but $2x^2 \neq o(x^2)$, since x^2 does not grow much faster than $2x^2$. Hence, the little-o notation implies the big-O notation. In the following we will only use the big-O notation.

The big-O has the following properties:

- multiplying by a constant: if $f = O(g)$ and c is a nonzero constant, then $cf = O(g)$,
- product rule: if $f = O(F)$ and $g = O(G)$, then $fg = O(FG)$, and for any h we have $hO(g) = O(hg)$,
- addition rule: if $f = O(F)$ and $g = O(G)$, then $f + g = O(\max(F, G))$.

Another rule that we use, denoted by the integral rule, states that for f a monotone decreasing, non-negative function,

$$\sum_{p\in\mathcal{P},\ p>H} f(p) = O\left(\int_H^\infty f(x)dx\right).$$

For example, for $s \geq 2$ we have that

$$\sum_{p \in \mathcal{P},\ p > H} \frac{1}{p^s} = O\left(\frac{1}{H^{s-1}}\right).$$

On the other hand, for $s \geq 2$ we have that

$$\sum_{p \in \mathcal{P},\ p \leq H} O\left(\frac{1}{p^s}\right) = O\left(\int_1^H \frac{1}{x^s} dx\right) = O\left(\frac{1}{1-s}\left(H^{1-s} - 1\right)\right) = O\left(1\right).$$

And for the special case $s = 1$ we get that

$$\sum_{p \in \mathcal{P},\ p \leq H} \frac{1}{p} = O\left(\ln(H)\right)$$

since,

$$\sum_{p \in \mathcal{P},\ p \leq H} \frac{1}{p} \leq \sum_{i=1}^{H} \frac{1}{n} \leq 1 + \int_1^H \frac{1}{x} dx \leq 2\ln(H).$$

The strategy, following [70], of computing the mean and the variance can be summarized as follows. For $s \geq 2$ a positive integer, let

$$\mathcal{H}(s, H) = \{a \in \mathbb{Z}^d \cap [-H, H[^d | \ a \text{ satisfies } C_s\}.$$

1. Compute the size of $\mathcal{H}(s, H)$.
2. Compute $\sum_{a \in T(H)} \psi(a) = \sum_{p \in \mathcal{P},\ p < H} \mid \mathcal{H}(p, H) \mid$.
3. Compute $\sum_{a \in T(H)} \psi(a)^2$ using $\mid \mathcal{H}(p, H) \mid$ and $\mid \mathcal{H}(pq, H) \mid$.
4. Compute $\mid T(H) \mid$ using the inclusion-exclusion principle, *i.e.*,

$$\mid T(H) \mid = -\sum_{s=2}^{H} \mu(s) \mid \mathcal{H}(s, H) \mid .$$

5. Lastly, compute μ and σ^2, by putting the previous results together.

11.2 Coprime Pairs

In this section we want to compute the mean and the variance of the number of primes that divide a non-coprime pair.

Let $H \in \mathbb{N}$ and $C(H)$ be the set of coprime pairs over the integers of height at most H, *i.e.*,

$$C(H) = \{a = (a_1, a_2) \in \mathbb{Z}^2 \mid \max\{\mid a_1 \mid, \mid a_2 \mid\} \leq H, \gcd(a_1, a_2) = 1\}.$$

Let us then denote by $C^C(H)$ the set of pairs, which are not coprime, but bounded by H.

For $s \geq 2$ a positive integer, let us define $\mathcal{H}(s,H)$ to be the set of pairs over the integers, which are of height at most H and divisible by s, *i.e.*,

$$\mathcal{H}(s,H) = \{a = (a_1,a_2) \in \mathbb{Z}^2 \mid \max\{\mid a_1 \mid, \mid a_2 \mid\} \leq H, s \mid a_1, s \mid a_2\}.$$

Let us define

$$\alpha = \sum_{p \in \mathcal{P}} \frac{1}{p^2},$$
$$\beta = \sum_{p \in \mathcal{P}} \frac{1}{p^4}.$$

We first compute the size of $\mathcal{H}(s,H)$.

Lemma 11.2.1. *For $s \leq H$, we have that*

$$\mid \mathcal{H}(s,H) \mid = \frac{(2H)^2}{s^2} + O\left(\frac{H}{s}\right).$$

Proof. The number of admissible values of a_i is

$$2\left\lfloor \frac{H}{s} \right\rfloor + 1 = \frac{2H}{s} + O(1).$$

Hence in total we have

$$\left(\frac{2H}{s} + O(1)\right)^2 = \frac{(2H)^2}{s^2} + O\left(\frac{H}{s}\right)$$

choices for (a_1,a_2). □

Let us denote by $\psi(a)$ the number of primes that divide $a = (a_1,a_2)$, *i.e.*,

$$\psi(a) = \mid \{p \in \mathcal{P} \mid p \mid a_1, p \mid a_2\} \mid .$$

Lemma 11.2.2. *Let $H \in \mathbb{N}$, then*

$$\sum_{a \in C^C(H)} \psi(a) = (2H)^2\alpha + O(H\ln(H)).$$

Proof. Observe that

$$\sum_{a \in C^C(H)} \psi(a) = \sum_{p \in \mathcal{P},\ p \leq H} \mid \mathcal{H}(p,H) \mid .$$

Now, we can apply Lemma 11.2.1 and get

$$\begin{aligned}\sum_{a\in C^C(H)}\psi(a) &= \sum_{p\in\mathcal{P},\ p\le H}\left(\frac{(2H)^2}{p^2}+O\left(\frac{H}{p}\right)\right)\\ &= (2H)^2\sum_{p\in\mathcal{P},\ p\le H}\frac{1}{p^2}+\sum_{p\in\mathcal{P},\ p\le H}O\left(\frac{H}{p}\right)\\ &= (2H)^2\left(\sum_{p\in\mathcal{P}}\frac{1}{p^2}-\sum_{p\in\mathcal{P},\ p>H}\frac{1}{p^2}\right)+O(H\ln(H))\\ &= (2H)^2\sum_{p\in\mathcal{P}}\frac{1}{p^2}+(2H)^2O\left(\frac{1}{H}\right)+O(H\ln(H))\\ &= (2H)^2\alpha+O(H\ln(H)).\end{aligned}$$

□

For $a=(a_1,a_2)\in\mathbb{Z}^2$, let us define

$$\tau(a,p)=\begin{cases}1 & \text{if } p\mid a_1, p\mid a_2,\\ 0 & \text{else.}\end{cases}$$

Lemma 11.2.3. *For $H\in\mathbb{N}$, we have that*

$$\sum_{a\in C^C(H)}\psi(a)^2=(2H)^2(\alpha+\alpha^2-\beta)+o(H\ln(H)).$$

Proof. First, we observe that $\psi(a)=\sum_{p\in\mathcal{P}}\tau(a,p)$, then

$$\begin{aligned}\sum_{a\in C^C(H)}\psi(a)^2 &= \sum_{a\in C^C(H)}\left(\sum_{p\in\mathcal{P}}\tau(a,p)\right)^2\\ &= \sum_{a\in C^C(H)}\left(\sum_{p,q\in\mathcal{P}}\tau(a,p)\tau(a,q)\right)\\ &= \sum_{p,q\in\mathcal{P}}\sum_{a\in C^C(H)}\tau(a,p)\tau(a,q).\end{aligned}$$

Now, we can split this sum into $p=q$ and $p\neq q$, where in the first case we observe, that $\tau(a,p)^2=\tau(a,p)$ and in the latter case, we have that

$$\tau(a,p)\tau(a,q)=\begin{cases}1 & \text{if } \tau(a,p)=\tau(a,q)=1,\\ 0 & \text{else.}\end{cases}$$

Hence we have that $\tau(a,p)\tau(a,q)=\tau(a,pq)$ and we can write

$$\sum_{a\in C^C(H)} \psi(a)^2 = \sum_{p\in\mathcal{P}} \sum_{a\in C^C(H)} \tau(a,p) + \sum_{p\neq q\in\mathcal{P}} \sum_{a\in C^C(H)} \tau(a,pq).$$

Further, we observe that

$$\sum_{a\in C^C(H)} \tau(a,s) = \mid \mathcal{H}(s,H) \mid .$$

Thus, we get

$$\sum_{a\in C^C(H)} \psi(a)^2 = \sum_{p\in\mathcal{P},\ p\leq H} \mid \mathcal{H}(p,H) \mid + \sum_{p\neq q\in\mathcal{P},\ pq\leq H} \mid \mathcal{H}(pq,H) \mid .$$

Using Lemma 11.2.2, we know that

$$\sum_{p\in\mathcal{P},p\leq H} \mid \mathcal{H}(p,H) \mid = (2H)^2\alpha + O(H\ln(H)),$$

and using Lemma 11.2.1, we know that

$$\begin{aligned}
&\sum_{p\neq q\in\mathcal{P},\ pq\leq H} \mid \mathcal{H}(pq,H) \mid \\
=&\sum_{p\neq q\in\mathcal{P},pq\leq H} \left(\frac{(2H)^2}{p^2q^2} + O\left(\frac{H}{pq}\right)\right) \\
=&(2H)^2 \sum_{p\neq q\in\mathcal{P},\ pq\leq H} \frac{1}{p^2q^2} + \sum_{p\neq q\in\mathcal{P},\ pq\leq H} O\left(\frac{H}{pq}\right) \\
=&(2H)^2 \sum_{p\neq q\in\mathcal{P},\ pq\leq H} \frac{1}{p^2q^2} + O(H\ln(H)) \\
=&(2H)^2 \sum_{p,q\in\mathcal{P},\ pq\leq H} \frac{1}{p^2q^2} - (2H)^2 \sum_{p\in\mathcal{P},\ p^2\leq H} \frac{1}{p^4} + O(H\ln(H)).
\end{aligned}$$

Let us first consider the first summand, which we can rewrite as

$$\begin{aligned}
(2H)^2 \sum_{p,q\in\mathcal{P},\ pq\leq H} \frac{1}{p^2q^2} &= (2H)^2 \left(\sum_{p,q\in\mathcal{P}} \frac{1}{p^2q^2} - \sum_{p,q\in\mathcal{P},\ pq>H} \frac{1}{p^2q^2}\right) \\
&= (2H)^2 \sum_{p,q\in\mathcal{P}} \frac{1}{p^2q^2} + (2H)^2 O\left(\frac{1}{H}\right) \\
&= (2H)^2 \sum_{p,q\in\mathcal{P}} \frac{1}{p^2q^2} + O(H) \\
&= (2H)^2 \left(\sum_{p\in\mathcal{P}} \frac{1}{p^2}\right)^2 + O(H) \\
&= (2H)^2\alpha^2 + O(H).
\end{aligned}$$

Whereas, the second summand can be written as

$$\begin{aligned}(2H)^2 \sum_{p\in\mathcal{P},\ p^2\leq H} \frac{1}{p^4} &= (2H)^2 \sum_{p\in\mathcal{P}} \frac{1}{p^4} - (2H)^2 \sum_{p\in\mathcal{P},\ p>\sqrt{H}} \frac{1}{p^4} \\ &= (2H)^2 \sum_{p\in\mathcal{P}} \frac{1}{p^4} + (2H)^2 O\left(\frac{1}{H^{3/2}}\right) \\ &= (2H)^2\beta + O\left(H^{1/2}\right).\end{aligned}$$

Hence we get the claim

$$\begin{aligned}\sum_{a\in C^C(H)} \psi(a)^2 &= (2H)^2\alpha + (2H)^2\alpha^2 - (2H)^2\beta + O(H\ln(H)) \\ &= (2H)^2(\alpha+\alpha^2-\beta) + O(H\ln(H)).\end{aligned}$$

□

Theorem 11.2.4. *For $H \in \mathbb{N}$, we have that*

$$\mid C^C(H) \mid = (2H)^2\rho(C^C) + O(H\ln(H)).$$

Proof. Using the inclusion-exclusion principle, we get that

$$\mid C^C(H) \mid = -\sum_{s=2}^{H} \mu(s) \mid \mathcal{H}(s,H) \mid .$$

Now, we can apply Lemma 11.2.1, to get that

$$\begin{aligned}\mid C^C(H) \mid &= -\sum_{s=2}^{H} \left(\mu(s)\frac{(2H)^2}{s^2} + O\left(\frac{H}{s}\right)\right) \\ &= -\sum_{s=2}^{H} \mu(s)\frac{(2H)^2}{s^2} + \sum_{s=2}^{H} O\left(\frac{H}{s}\right) \\ &= -(2H)^2 \sum_{s\geq 2} \mu(s)\frac{1}{s^2} + (2H)^2 \sum_{s\geq H+1} \mu(s)\frac{1}{s^2} + O\left(H\ln(H)\right) \\ &= -(2H)^2 \sum_{s\geq 2} \mu(s)\frac{1}{s^2} + (2H)^2 O\left(\frac{1}{H}\right) + O\left(H\ln(H)\right) \\ &= -(2H)^2 \sum_{s\geq 2} \mu(s)\frac{1}{s^2} + O\left(H\ln(H)\right).\end{aligned}$$

With the help of the Euler product, we get

$$
\begin{aligned}
-\sum_{s\geq 2}\mu(s)\frac{1}{s^2} &= 1-\sum_{s\geq 1}\mu(s)\frac{1}{s^2}\\
&= 1-\prod_{p\in\mathcal{P}}\left(1-\frac{1}{p^2}\right)\\
&= 1-\rho(C) = \rho(C^C).
\end{aligned}
$$

Therefore, we can conclude, as

$$
\begin{aligned}
\mid C^C(H)\mid &= (2H)^2\left(-\sum_{s\geq 2}\mu(s)\frac{1}{s^2}\right)+O(H\ln(H))\\
&= (2H)^2\rho(C^C)+O(H\ln(H)).
\end{aligned}
$$

□

Note that this can be seen as an alternative proof for the density of coprime pairs. We are ready to state the main theorem of this section.

Theorem 11.2.5. *The mean and the variance of the function $\psi(a)$, where a ranges over all non-coprime pairs over the integers are*

$$
\begin{aligned}
\mu &= \lim_{H\to\infty}\frac{\sum\limits_{a\in C^C(H)}\psi(a)}{\mid C^C(H)\mid} = \frac{\alpha}{\rho(C^C)},\\
\sigma^2 &= \lim_{H\to\infty}\frac{\sum\limits_{a\in C^C(H)}(\psi(a)-\mu)^2}{\mid C^C(H)\mid} = \frac{\alpha+\alpha^2-\beta-\mu\alpha}{\rho(C^C)}.
\end{aligned}
$$

Proof. From Lemma 11.2.2 and Theorem 11.2.4, it follows directly that

$$
\begin{aligned}
\mu &= \lim_{H\to\infty}\frac{\sum\limits_{a\in C^C(H)}\psi(a)}{\mid C^C(H)\mid}\\
&= \lim_{H\to\infty}\frac{(2H)^2\alpha+O(H\ln(H))}{(2H)^2\rho(C^C)+O(H\ln(H))}\\
&= \frac{\alpha}{\rho(C^C)}.
\end{aligned}
$$

For the variance we use Lemma 11.2.2, Lemma 11.2.3 and Theorem 11.2.4, to get

$$\begin{aligned}
\sigma^2 &= \lim_{H\to\infty} \frac{\sum\limits_{a\in C^C(H)} (\psi(a)-\mu)^2}{\mid C^C(H) \mid} \\
&= \lim_{H\to\infty} \frac{\sum\limits_{a\in C^C(H)} (\psi(a)^2 - 2\mu\psi(a) + \mu^2)}{\mid C^C(H) \mid} \\
&= \lim_{H\to\infty} \frac{\sum\limits_{a\in C^C(H)} \psi(a)^2 - 2\mu \sum\limits_{a\in C^C(H)} \psi(a) + \mu^2 \mid C^C(H) \mid}{\mid C^C(H) \mid} \\
&= \lim_{H\to\infty} \frac{(2H)^2(\alpha+\alpha^2-\beta-2\mu\alpha+\mu^2\rho(C^C)) + O(H\ln(H))}{(2H)^2\rho(C^C) + O(H\ln(H))} \\
&= \frac{\alpha+\alpha^2-\beta-2\mu\alpha+\mu^2\rho(C^C)}{\rho(C^C)} \\
&= \frac{\alpha+\alpha^2-\beta-2\mu\alpha+\mu\frac{\alpha}{\rho(C^C)}\rho(C^C)}{\rho(C^C)} \\
&= \frac{\alpha+\alpha^2-\beta-\mu\alpha}{\rho(C^C)}.
\end{aligned}$$

□

Coprime m-tuples

Since the result for coprime m-tuples is similar, even easier, than the case of coprime pairs, we will just give the results and point out the differences to the coprime pairs.

Let $m > 2$ and H be positive integers and $C_m(H)$ be the set of coprime m-tuples over the integers of height at most H, *i.e.*,

$$C_m(H) = \{(a_1,\ldots,a_m) \in \mathbb{Z}^m \mid \max_{i\in\{1,\ldots,m\}} \{\mid a_i \mid\} \leq H, I(a_1,\ldots,a_m) = \mathbb{Z}\}.$$

For $s \geq 2$ a positive integer, let us define $\mathcal{H}_m(s,H)$ to be the set of m-tuples over the integers, which are of height at most H and divisible by s, *i.e.*,

$$\begin{aligned}
\mathcal{H}_m(s,H) = \{a = (a_1,\ldots,a_m) \in \mathbb{Z}^m \mid &\max_{i\in\{1,\ldots,m\}} \{\mid a_i \mid\} \leq H, \\
&s \mid a_i \ \forall\ i \in \{1,\ldots,m\}\}.
\end{aligned}$$

An important step is to compute the size of $\mathcal{H}_m(s,H)$.

Lemma 11.2.6. *For $s \leq H \in \mathbb{N}$, we have that*

$$\mid \mathcal{H}_m(s,H) \mid = \frac{(2H)^m}{s^m} + O\left(\frac{H^{m-1}}{s^{m-1}}\right).$$

The subresults are similar to the case $m = 2$, with the exception that we always have $O(1)$, instead of $O(H\ln(H))$.

Let us define

$$\alpha = \sum_{p \in \mathcal{P}} \frac{1}{p^m},$$
$$\beta = \sum_{p \in \mathcal{P}} \frac{1}{p^{2m}}.$$

Then, the mean and the variance of primes diving non-coprime m-tuples is given by the following theorem.

Theorem 11.2.7. *The mean and the variance of the function $\psi(a)$, where a ranges over all non-coprime m-tuples over the integers are*

$$\mu = \lim_{H \to \infty} \frac{\sum\limits_{a \in C_m^C(H)} \psi(a)}{\mid C_m^C(H) \mid} = \frac{\alpha}{\rho(C_m^C)},$$
$$\sigma^2 = \lim_{H \to \infty} \frac{\sum\limits_{a \in C_m^C(H)} (\psi(a) - \mu)^2}{\mid C_m^C(H) \mid} = \frac{\alpha + \alpha^2 - \beta - \mu\alpha}{\rho(C_m^C)}.$$

11.3 Eisenstein Polynomials

In this section we compute the mean and the variance of the number of primes for which an Eisenstein polynomial satisfies the criterion of Eisenstein.

For the non-monic Eisenstein polynomials this result is due to [70], where they used the work of Heyman and Shparlinski [52]. We follow their proofs.

Since the computations for the monic case are similar, we will just give the results and indicate the differences.

First, recall that

$$\varphi(s) = s \sum_{m \mid s} \frac{\mu(m)}{m}$$

and for $s \leq H$, it holds that $2^{\omega(s)} = O(H)$, see [52].

Further, from Equation 11 in [52] we get that

$$\sum_{s=2}^{H} \frac{2^{\omega(s)}}{s^{d-1}} = O(1). \tag{11.3.1}$$

Let us define

$$\chi(s, H) = \mid \{a \mid \mid a \mid \leq H, \gcd(a, s) = 1\} \mid .$$

We need the following lemma.

Lemma 11.3.1 (Lemma 4, [52])**.** *For $2 \leq s \leq H$, we have that*

$$\chi(s, H) = \frac{2H\varphi(s)}{s} + O\left(2^{\omega(s)}\right).$$

Proof. We can write $\chi(s,H)$ as

$$\chi(s,H) = \sum_{\substack{|a|\leq H \\ \gcd(a,s)=1}} 1.$$

With the inclusion-exclusion principle, we have that

$$\begin{aligned}
\chi(s,H) &= \sum_{m|s} \mu(m) \sum_{\substack{|a|\leq H \\ m|a}} 1 \\
&= \sum_{m|s} \mu(m) \left(2 \left\lfloor \frac{H}{m} \right\rfloor + 1 \right) \\
&= 2H \sum_{m|s} \frac{\mu(m)}{m} + O\left(\sum_{m|s} \mid \mu(m) \mid \right) \\
&= \frac{2H\varphi(s)}{s} + O\left(2^{\omega(s)}\right)
\end{aligned}$$

□

Let $H \in \mathbb{N}, d > 2$ and $E_d(H)$ be the set of Eisenstein polynomials of degree d and of height at most H, *i.e.*,

$$\begin{aligned}
E_d(H) = \{a = (a_0, \ldots, a_d) \in \mathbb{Z}^{d+1} \mid &\max_{i\in\{0,\ldots,d\}} \{\mid a_i \mid\} \leq H, \\
&\exists p \; p^2 \nmid a_0, p \nmid a_d \text{ and } \forall i < d, p \mid a_i\}.
\end{aligned}$$

For $s \geq 2$ a positive integer, we define $\mathcal{H}_d(s,H)$ to be the set of polynomials of degree d, which are of height at most H and such that the polynomial is s-Eisenstein, *i.e.*,

$$\begin{aligned}
\mathcal{H}_d(s,H) = \{a = (a_0, \ldots, a_d) \in \mathbb{Z}^{d+1} \mid &\max_{i\in\{0,\ldots,d\}} \{\mid a_i \mid\} \leq H, \\
&\forall \; i < d, s \mid a_i, \; \gcd(\frac{a_0}{s}, s) = 1, \; \gcd(a_d, s) = 1\}.
\end{aligned}$$

Let us define

$$\begin{aligned}
\alpha &= \sum_{p\in\mathcal{P}} \frac{(p-1)^2}{p^{d+2}}, \\
\beta &= \sum_{p\in\mathcal{P}} \frac{(p-1)^4}{p^{2(d+2)}}.
\end{aligned}$$

We first compute the size of $\mathcal{H}_d(s,H)$.

Lemma 11.3.2 (Lemma 5, [52])**.** *For $s \leq H \in \mathbb{N}$, we have that*

$$\mid \mathcal{H}_d(s,H) \mid = \frac{(2H)^{d+1}\varphi(s)^2}{s^{d+2}} + O\left(\frac{H^d}{s^{d-1}} 2^{\omega(s)} \right).$$

Proof. The number of admissible values of a_i for $i \in \{1, \ldots, d-1\}$ is

$$2\left\lfloor \frac{H}{s} \right\rfloor + 1 = \frac{2H}{s} + O(1).$$

Whereas, the number of admissible values for a_d, which are such that $\gcd(a_d, s) = 1$, is given by Lemma 11.3.1 as

$$\chi(s, H) = \frac{2H\varphi(s)}{s} + O\left(2^{\omega(s)}\right).$$

Lastly, the number of admissible values for a_0, which are such that $s \mid a_0$ and $\gcd(\frac{a_0}{s}, s) = 1$ is given by Lemma 11.3.1 as

$$\chi\left(s, \left\lfloor \frac{H}{s} \right\rfloor\right) = \frac{2H\varphi(s)}{s^2} + O\left(2^{\omega(s)}\right).$$

Hence in total we have

$$\begin{aligned}
&\left(\frac{2H}{s} + O(1)\right)^{d-1} \left(\frac{2H\varphi(s)}{s^2} + O\left(2^{\omega(s)}\right)\right)\left(\frac{2H\varphi(s)}{s} + O\left(2^{\omega(s)}\right)\right) \\
=&\left(\frac{(2H)^{d-1}}{s^{d-1}} + O\left(\frac{H^{d-2}}{s^{d-2}}\right)\right)\left(\frac{2H\varphi(s)}{s^2} + O\left(2^{\omega(s)}\right)\right) \\
&\cdot\left(\frac{2H\varphi(s)}{s} + O\left(2^{\omega(s)}\right)\right) \\
=&\left(\frac{(2H)^{d-1}}{s^{d-1}} + O\left(\frac{H^{d-2}}{s^{d-2}}\right)\right)\left(\frac{(2H)^2\varphi(s)^2}{s^3} + O\left(H2^{\omega(s)}\right)\right) \\
=&\frac{(2H)^{d+1}\varphi(s)^2}{s^{d+2}} + O\left(\frac{H^d}{s^{d-1}}2^{\omega(s)}\right)
\end{aligned}$$

choices for $(a_0, \ldots, a_d)$. □

Let us denote by $\psi(a)$ the number of primes, such that the polynomial associated to $a = (a_0, \ldots, a_d)$ is Eisenstein, *i.e.*,

$$\psi(a) = \mid \{p \in \mathcal{P} \mid \ \forall\ i < d\ p \mid a_i,\ p^2 \nmid a_0,\ p \nmid a_d\} \mid .$$

Lemma 11.3.3 (Lemma 4, [70])**.** *Let $H \in \mathbb{N}$, then*

$$\sum_{a \in E_d(H)} \psi(a) = (2H)^{d+1}\alpha + O(H^d).$$

Proof. We note that

$$\sum_{a \in E_d(H)} \psi(a) = \sum_{p \in \mathcal{P}, p \leq H} \mid \mathcal{H}_d(p, H) \mid .$$

Hence we can apply Lemma 11.3.2 and get

$$\begin{aligned}\sum_{a\in E_d(H)}\psi(a) &= \sum_{p\in\mathcal{P},p\le H}\left(\frac{(2H)^{d+1}(p-1)^2}{p^{d+2}}+O\left(\frac{H^d}{p^{d-1}}2^{\omega(p)}\right)\right)\\ &= (2H)^{d+1}\sum_{p\in\mathcal{P},p\le H}\frac{(p-1)^2}{p^{d+2}}+\sum_{p\in\mathcal{P},p\le H}O\left(\frac{H^d}{p^{d-1}}\right)\\ &= (2H)^{d+1}\left(\sum_{p\in\mathcal{P}}\frac{(p-1)^2}{p^{d+2}}-\sum_{p\in\mathcal{P},p>H}\frac{(p-1)^2}{p^{d+2}}\right)+O(H^d)\\ &= (2H)^{d+1}\sum_{p\in\mathcal{P}}\frac{(p-1)^2}{p^{d+2}}-(2H)^{d+1}O\left(\frac{1}{H^{d-1}}\right)+O(H^d)\\ &= (2H)^{d+1}\alpha+O(H^d).\end{aligned}$$

□

For $a=(a_0,\dots,a_d)\in\mathbb{Z}^{d+1}$, we define $\tau(a,p)$ to be the following map

$$\tau(a,p)=\begin{cases}1 & \text{if } \forall i<d\ p\mid a_i,\ p^2\nmid a_0,\ p\nmid a_d\\ 0 & \text{else.}\end{cases}$$

Lemma 11.3.4 (Lemma 5, [70]). *For $H\in\mathbb{N}$, we have that*

$$\sum_{a\in E_d(H)}\psi(a)^2=(2H)^{d+1}(\alpha+\alpha^2-\beta)+O(H^d).$$

Proof. By the definition of τ, we have that $\psi(a)=\sum_{p\in\mathcal{P}}\tau(a,p)$. Therefore, we get

$$\begin{aligned}\sum_{a\in E_d(H)}\psi(a)^2 &= \sum_{a\in E_d(H)}\left(\sum_{p\in\mathcal{P}}\tau(a,p)\right)^2\\ &= \sum_{a\in E_d(H)}\left(\sum_{p,q\in\mathcal{P}}\tau(a,p)\tau(a,q)\right)\\ &= \sum_{p,q\in\mathcal{P}}\sum_{a\in E_d(H)}\tau(a,p)\tau(a,q).\end{aligned}$$

Again, we can split this sum into $p=q$ and $p\neq q$. In the first case, observe that $\tau(a,p)^2=\tau(a,p)$ and in the second case, we have that $\tau(a,p)\tau(a,q)=\tau(a,pq)$, since

$$\tau(a,p)\tau(a,q)=\begin{cases}1 & \text{if } \tau(a,p)=\tau(a,q)=1,\\ 0 & \text{else.}\end{cases}$$

Hence we can write

$$\sum_{a\in E_d(H)}\psi(a)^2=\sum_{p\in\mathcal{P}}\sum_{a\in E_d(H)}\tau(a,p)+\sum_{p\neq q\in\mathcal{P}}\sum_{a\in E_d(H)}\tau(a,pq).$$

Since

$$\sum_{a\in E_d(H)} \tau(a,s) =\mid \mathcal{H}_d(s,H) \mid,$$

we get

$$\sum_{a\in E_d(H)} \psi(a)^2 = \sum_{p\in\mathcal{P},p\leq H} \mid \mathcal{H}_d(p,H) \mid + \sum_{p\neq q\in\mathcal{P},pq\leq H} \mid \mathcal{H}_d(pq,H) \mid .$$

Using Lemma 11.3.3, we know that

$$\sum_{p\in\mathcal{P},p\leq H} \mid \mathcal{H}_d(p,H) \mid= (2H)^{d+1}\alpha + O(H^d),$$

and by Lemma 11.3.2 we have that

$$\begin{aligned}
&\sum_{p\neq q\in\mathcal{P},\ pq\leq H} \mid \mathcal{H}_d(pq,H) \mid \\
=&\sum_{p\neq q\in\mathcal{P},\ pq\leq H} \left(\frac{(2H)^{d+1}(p-1)^2(q-1)^2}{p^{d+2}q^{d+2}} + O\left(\frac{H^d}{p^{d-1}q^{d-1}}2^{\omega(pq)}\right)\right) \\
=&(2H)^{d+1}\sum_{p\neq q\in\mathcal{P},\ pq\leq H} \frac{(p-1)^2(q-1)^2}{p^{d+2}q^{d+2}} \\
&+\sum_{p\neq q\in\mathcal{P},\ pq\leq H} O\left(\frac{H^d}{p^{d-1}q^{d-1}}\right) \\
=&(2H)^{d+1}\sum_{p\neq q\in\mathcal{P},\ pq\leq H} \frac{(p-1)^2(q-1)^2}{p^{d+2}q^{d+2}} + O(H^d) \\
=&(2H)^{d+1}\sum_{p,q\in\mathcal{P},\ pq\leq H} \frac{(p-1)^2(q-1)^2}{p^{d+2}q^{d+2}} \\
&-(2H)^{d+1}\sum_{p\in\mathcal{P},\ p^2\leq H} \frac{(p-1)^4}{p^{2(d+2)}} + O(H^d).
\end{aligned}$$

The first summand gives us $(2H)^{d+1}\alpha^2 + O(H^2)$, since

$$\begin{aligned}
&(2H)^{d+1} \sum_{p,q\in\mathcal{P},\ pq\leq H} \frac{(p-1)^2(q-1)^2}{p^{d+2}q^{d+2}} \\
&= (2H)^{d+1} \left(\sum_{p,q\in\mathcal{P}} \frac{(p-1)^2(q-1)^2}{p^{d+2}q^{d+2}} - \sum_{p,q\in\mathcal{P},\ pq>H} \frac{(p-1)^2(q-1)^2}{p^{d+2}q^{d+2}} \right) \\
&= (2H)^{d+1} \sum_{p,q\in\mathcal{P}} \frac{(p-1)^2(q-1)^2}{p^{d+2}q^{d+2}} + (2H)^{d+1} O\left(\frac{1}{H^{d-1}}\right) \\
&= (2H)^{d+1} \left(\sum_{p\in\mathcal{P}} \frac{(p-1)^2}{p^{d+1}} \right)^2 + O(H^2) \\
&= (2H)^{d+1}\alpha^2 + O(H^2).
\end{aligned}$$

Whereas, the second summand can be written as

$$\begin{aligned}
&(2H)^{d+1} \sum_{p\in\mathcal{P},\ p^2\leq H} \frac{(p-1)^4}{p^{2(d+2)}} \\
=&(2H)^{d+1} \sum_{p\in\mathcal{P}} \frac{(p-1)^4}{p^{2(d+2)}} - (2H)^{d+1} \sum_{p\in\mathcal{P},\ p>\sqrt{H}} \frac{(p-1)^4}{p^{2(d+2)}} \\
=&(2H)^{d+1} \sum_{p\in\mathcal{P}} \frac{(p-1)^4}{p^{2(d+2)}} + (2H)^{d+1} O\left(H^{-d+\frac{1}{2}}\right) \\
=&(2H)^{d+1}\beta + O\left(H^{3/2}\right).
\end{aligned}$$

We can conclude, as

$$\begin{aligned}
\sum_{a\in E_d(H)} \psi(a)^2 =&(2H)^{d+1}\alpha + (2H)^{d+1}\alpha^2 - (2H)^{d+1}\beta + O\left(H^d\right) \\
=&(2H)^{d+1}(\alpha + \alpha^2 - \beta) + O(H^d).
\end{aligned}$$

□

As a next step, we compute the size of $E_d(H)$.

Theorem 11.3.5 (Theorem 2, [52]). *For $H \in \mathbb{N}$ we have that*

$$| E_d(H) | = (2H)^{d+1}\rho(E_d) + O(H^d).$$

Proof. By the inclusion-exclusion principle, we get that

$$| E_d(H) | = -\sum_{s=2}^{H} \mu(s) \mid \mathcal{H}_d(s, H) \mid .$$

Using (11.3.1) and Lemma 11.3.2, we get that

$$\begin{aligned}
\mid E_d(H) \mid &= -\sum_{s=2}^{H}\left(\mu(s)\frac{(2H)^{d+1}\varphi(s)^2}{s^{d+2}} + O\left(\frac{H^d}{s^{d-1}}2^{\omega(s)}\right)\right) \\
&= -\sum_{s=2}^{H}\mu(s)\frac{(2H)^{d+1}\varphi(s)^2}{s^{d+2}} + \sum_{s=2}^{H}O\left(\frac{H^d}{s^{d-1}}2^{\omega(s)}\right) \\
&= -(2H)^{d+1}\sum_{s=2}^{H}\mu(s)\frac{\varphi(s)^2}{s^{d+2}} + O(H^d) \\
&= -(2H)^{d+1}\sum_{s\geq 2}\mu(s)\frac{\varphi(s)^2}{s^{d+2}} \\
&\quad + (2H)^{d+1}\sum_{s\geq H+1}\mu(s)\frac{\varphi(s)^2}{s^{d+2}} + O(H^d) \\
&= -(2H)^{d+1}\sum_{s\geq 2}\mu(s)\frac{\varphi(s)^2}{s^{d+2}} + (2H)^{d+1}O\left(\frac{1}{H^{d-1}}\right) + O(H^d) \\
&= -(2H)^{d+1}\sum_{s\geq 2}\mu(s)\frac{\varphi(s)^2}{s^{d+2}} + O(H^d).
\end{aligned}$$

Since $\frac{\varphi(s)^2}{s^{d+2}}$ is a multiplicative function, we have that

$$\begin{aligned}
-\sum_{s\geq 2}\mu(s)\frac{\varphi(s)^2}{s^{d+2}} &= 1 - \sum_{s\geq 1}\mu(s)\frac{\varphi(s)^2}{s^{d+2}} \\
&= 1 - \prod_{p\in\mathcal{P}}\left(1 - \frac{(p-1)^2}{p^{d+2}}\right) \\
&= \rho(E_d).
\end{aligned}$$

We can conclude, as

$$\begin{aligned}
\mid E_d(H) \mid &= (2H)^{d+1}\left(-\sum_{s\geq 2}\mu(s)\frac{\varphi(s)^2}{s^{d+2}}\right) + O(H^d) \\
&= (2H)^{d+1}\rho(E_d) + O(H^d).
\end{aligned}$$

□

Note that this is an alternative proof for the density of Eisenstein polynomials. We are ready to state the main theorem of this section.

Theorem 11.3.6 (Theorem 2, [70]). *The mean and the variance of the function $\psi(a)$, where a ranges over all Eisenstein polynomials are*

$$\mu = \lim_{H\to\infty} \frac{\sum\limits_{a\in E_d(H)} \psi(a)}{\mid E_d(H) \mid} = \frac{\alpha}{\rho(E_d)},$$
$$\sigma^2 = \lim_{H\to\infty} \frac{\sum\limits_{a\in E_d(H)} (\psi(a)-\mu)^2}{\mid E_d(H) \mid} = \frac{\alpha+\alpha^2-\beta-\mu\alpha}{\rho(E_d)}.$$

Proof. To compute the mean we can use Lemma 11.3.3 and Theorem 11.3.5, which give

$$\begin{aligned}\mu &= \lim_{H\to\infty} \frac{\sum\limits_{a\in E_d(H)} \psi(a)}{\mid E_d(H) \mid} \\ &= \lim_{H\to\infty} \frac{(2H)^{d+1}\alpha + O(H^d)}{(2H)^{d+1}\rho(E_d) + O(H^d)} \\ &= \frac{\alpha}{\rho(E_d)}.\end{aligned}$$

For the variance, we use Lemma 11.3.3, Lemma 11.3.4 and Theorem 11.3.5, to get

$$\begin{aligned}\sigma^2 &= \lim_{H\to\infty} \frac{\sum\limits_{a\in E_d(H)} (\psi(a)-\mu)^2}{\mid E_d(H) \mid} \\ &= \lim_{H\to\infty} \frac{\sum\limits_{a\in E_d(H)} (\psi(a)^2 - 2\mu\psi(a) + \mu^2)}{\mid E_d(H) \mid} \\ &= \lim_{H\to\infty} \frac{\sum\limits_{a\in E_d(H)} \psi(a)^2 - 2\mu \sum\limits_{a\in E_d(H)} \psi(a) + \mu^2 \mid E_d(H) \mid}{\mid E_d(H) \mid} \\ &= \lim_{H\to\infty} \frac{(2H)^{d+1}(\alpha+\alpha^2-\beta-2\mu\alpha+\mu^2\rho(E_d)) + O(H^d)}{(2H)^{d+1}\rho(E_d) + O(H^d)} \\ &= \frac{\alpha+\alpha^2-\beta-2\mu\alpha+\mu^2\rho(E_d)}{\rho(E_d)} \\ &= \frac{\alpha+\alpha^2-\beta-2\mu\alpha+\mu\frac{\alpha}{\rho(E_d)}\rho(E_d)}{\rho(E_d)} \\ &= \frac{\alpha+\alpha^2-\beta-\mu\alpha}{\rho(E_d)}.\end{aligned}$$

□

The Monic Case

Note that the monic case is in fact easier than the non-monic case, thus we just state the result, and indicate differences in the proof.

Let $H \in \mathbb{N}, d > 2$ and $M_d(H)$ be the set of monic Eisenstein polynomials of degree d and of height at most H, *i.e.*,

$$M_d(H) = \{a = (a_0, \ldots, a_{d-1}) \in \mathbb{Z}^d \mid \max_{i \in \{0,\ldots,d-1\}} \{| a_i |\} \leq H, \\ \exists p \; \forall i < d \; p^2 \nmid a_0, \; p \mid a_i\}.$$

For $s \geq 2$ a positive integer, let us define $\mathcal{H}_d(s, H)$ to be the set of monic polynomials of degree d, which are of height at most H and such that the polynomial is s-Eisenstein, *i.e.*,

$$\mathcal{H}_d(s, H) = \left\{a = (a_0, \ldots, a_{d-1}) \in \mathbb{Z}^d \mid \max_{i \in \{0,\ldots,d-1\}} \{| a_i |\} \leq H, \right. \\ \left. \forall \; i < d \; s \mid a_i, \; \gcd\left(\frac{a_0}{s}, s\right) = 1\right\}.$$

The main difference between the two cases, is already in the first step, when computing the size of $\mathcal{H}_d(s, H)$.

Lemma 11.3.7. *For $s \leq H \in \mathbb{N}$, we have that*

$$| \mathcal{H}_d(s, H) | = \frac{(2H)^d \varphi(s)}{s^{d+1}} + O\left(\frac{H^{d-1}}{s^{d-1}} 2^{\omega(s)}\right).$$

The remaining sub-results follow in the exact same manner as for the non-monic case. We define

$$\alpha = \sum_{p \in \mathcal{P}} \frac{p-1}{p^{d+1}}, \\ \beta = \sum_{p \in \mathcal{P}} \frac{(p-1)^2}{p^{2(d+1)}}.$$

Theorem 11.3.8. *The mean and the variance of the function $\psi(a)$, where a ranges over all monic Eisenstein polynomials are*

$$\mu = \lim_{H \to \infty} \frac{\sum\limits_{a \in M_d(H)} \psi(a)}{| M_d(H) |} = \frac{\alpha}{\rho(M_d)}, \\ \sigma^2 = \lim_{H \to \infty} \frac{\sum\limits_{a \in M_d(H)} (\psi(a) - \mu)^2}{| M_d(H) |} = \frac{\alpha + \alpha^2 - \beta - \mu\alpha}{\rho(M_d)}.$$

11.4 Rectangular Unimodular Matrices

In this section we want to compute the mean and the variance of the number of primes for which a rectangular non-unimodular matrix has not full rank modulo this prime.

For simplicity we will restrict ourselves in the following to the case $n < m - 1$. However, in the case $n = m - 1$ the results remain true with the slight difference of having an error term of the form $O(H \ln(H))$.

Let $H \in \mathbb{N}, n < m-1 \in \mathbb{N}$ and $R_{n,m}(H)$ be the set of rectangular unimodular matrices in $\mathbb{Z}^{n\times m}$ of height at most H, *i.e.*,

$$R_{n,m}(H) = \left\{ A \in \mathbb{Z}^{n\times m} \mid \max_{\substack{i\in\{1,\ldots,n\}\\ j\in\{1,\ldots,m\}}} \{| a_{i,j} |\} \leq H,\ \forall p \in \mathcal{P}\, \mathrm{rk}(A \mod p) = n \right\}.$$

For p a prime, let us define $\mathcal{H}_{n,m}(p,H)$ to be the set of matrices in $\mathbb{Z}^{n\times m}$, which are of height at most H and such that the $\mathrm{rk}(A \mod p) \neq n$, *i.e.*,

$$\mathcal{H}_{n,m}(p,H) = \{A \in \mathbb{Z}^{n\times m} \mid \max_{\substack{i\in\{1,\ldots,n\}\\ j\in\{1,\ldots,m\}}} \{| a_{i,j} |\} \leq H, \mathrm{rk}(A \mod p) \neq n\}.$$

This means, that all n-minors of A are divisible by p.

For p,q two distinct primes, let us define $\mathcal{H}_{n,m}(p,q,H)$ to be the set of matrices in $\mathbb{Z}^{n\times m}$, which are of height at most H and such that the $\mathrm{rk}(A \mod p) \neq n$ and also $\mathrm{rk}(A \mod q) \neq n$, *i.e.*,

$$\mathcal{H}_{n,m}(p,q,H) = \{A \in \mathbb{Z}^{n\times m} \mid \max_{\substack{i\in\{1,\ldots,n\}\\ j\in\{1,\ldots,m\}}} \{| a_{i,j} |\} \leq H, \mathrm{rk}(A \mod p) \neq n, \mathrm{rk}(A \mod q) \neq n\}.$$

In this case all n-minors of A are divisible by p and q, and thus by pq.

Thus, we can define for an integer $s \geq 2$, $\mathcal{H}_{n,m}(s,H)$ to be the set of matrices in $\mathbb{Z}^{n\times m}$, which are of height at most H and such that all maximal sized minors m_i are zero modulo s, *i.e.*,

$$\mathcal{H}_{n,m}(s,H) = \{A \in \mathbb{Z}^{n\times m} \mid \max_{\substack{i\in\{1,\ldots,n\}\\ j\in\{1,\ldots,m\}}} \{| a_{i,j} |\} \leq H,\ \forall n\text{- minors } m_i,\ m_i \equiv 0 \mod s\}.$$

Clearly, $\mathcal{H}_{n,m}(p,q,H) = \mathcal{H}_{n,m}(pq,H)$.

Let us define

$$\alpha = \sum_{p\in\mathcal{P}} \left(1 - \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}}\right)\right),$$

$$\beta = \sum_{p\in\mathcal{P}} \left(1 - \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}}\right)\right)^2.$$

We usually would compute the size of $\mathcal{H}_{n,m}(s,H)$ for any integer s, but since we only apply this on $s = p$ prime and $s = pq$ for $p < q$ primes and in addition on s a square-free positive integer, since only then we have that $\mu(s) \neq 0$, we can restrict to these cases.

For p a prime, and $n < m$ we have that the number of full rank matrices in $\mathbb{F}_p^{n\times m}$ is

$$\prod_{i=0}^{n-1} (p^m - p^i).$$

Hence we can now compute the size of $\mathcal{H}_{n,m}(p,H)$.

Lemma 11.4.1. *For $p \leq H \in \mathbb{N}$, with p prime, we have that*

$$\mid \mathcal{H}_{n,m}(p,H) \mid = (2H)^{nm} \left(1 - \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}}\right)\right) + O\left(\frac{(2H)^{nm-1}}{p^{m-n}}\right).$$

Proof. We start by counting the number of non-full rank $n \times m$ matrices modulo p. This given by

$$p^{nm} - \prod_{i=0}^{n-1}(p^m - p^i) = p^{nm}\left(1 - \prod_{i=0}^{n-1}\left(1 - \frac{1}{p^{m-i}}\right)\right).$$

As a next step, we fix a choice of non-full rank matrix modulo p and lift it to $[-H, H[^{n\times m}$, for each fixed matrix we have (at most)

$$\left(\left\lceil \frac{2H}{p} \right\rceil\right)^{nm}$$

choices for a lift. Hence in total we get

$$\begin{aligned}
\mid \mathcal{H}_{n,m}(p,H) \mid &= \left(p^{nm}\left(1 - \prod_{i=0}^{n-1}\left(1 - \frac{1}{p^{m-i}}\right)\right)\right)\left(\left\lceil \frac{2H}{p} \right\rceil\right)^{nm} \\
&= \left(p^{nm}\left(1 - \prod_{i=0}^{n-1}\left(1 - \frac{1}{p^{m-i}}\right)\right)\right)\left(\frac{2H}{p} + O(1)\right)^{nm} \\
&= \left(p^{nm}\left(1 - \prod_{i=0}^{n-1}\left(1 - \frac{1}{p^{m-i}}\right)\right)\right) \\
&\quad \cdot \left(\left(\frac{2H}{p}\right)^{nm} + O\left(\left(\frac{2H}{p}\right)^{nm-1}\right)\right) \\
&= (2H)^{nm}\left(1 - \prod_{i=0}^{n-1}\left(1 - \frac{1}{p^{m-i}}\right)\right) + O\left(\frac{(2H)^{nm-1}}{p^{m-n}}\right).
\end{aligned}$$

□

With this we can now compute the size of $\mathcal{H}_{n,m}(pq,H)$.

Lemma 11.4.2. *For $p < q$ primes and $pq \leq H \in \mathbb{N}$, we have that*

$$\begin{aligned}
&\mid \mathcal{H}_{n,m}(pq,H) \mid \\
=&(2H)^{nm}\left(1 - \prod_{i=0}^{n-1}\left(1 - \frac{1}{p^{m-i}}\right)\right)\left(1 - \prod_{i=0}^{n-1}\left(1 - \frac{1}{q^{m-i}}\right)\right) + O\left(\frac{(2H)^{nm-1}}{(pq)^{m-n}}\right).
\end{aligned}$$

Proof. We start by counting the number of $n \times m$ matrices, which have not full rank

modulo p and modulo q. With the Chinese Remainder Theorem this is given by

$$\begin{aligned}&\left(p^{nm}-\prod_{i=0}^{n-1}\left(p^m-p^i\right)\right)\left(q^{nm}-\prod_{i=0}^{n-1}\left(q^m-q^i\right)\right)\\ =&(pq)^{nm}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{q^{m-i}}\right)\right).\end{aligned}$$

As a next step, we fix a choice of such a matrix modulo pq and lift it to $[-H,H[^{n\times m}$, for each fixed matrix we have

$$\left(\left\lceil\frac{2H}{pq}\right\rceil\right)^{nm}$$

choices for a lift. Hence in total we get

$$\begin{aligned}&\mid\mathcal{H}_{n,m}(pq,H)\mid\\ =&(pq)^{nm}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\cdot\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{q^{m-i}}\right)\right)\left(\left\lceil\frac{2H}{pq}\right\rceil\right)^{nm}\\ =&(2H)^{nm}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\cdot\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{q^{m-i}}\right)\right)+O\left(\frac{(2H)^{nm-1}}{(pq)^{m-n}}\right).\end{aligned}$$

□

Finally, we can extend Lemma 11.4.2 directly to square-free positive integers s.

Lemma 11.4.3. *For s a square-free positive integer and $s\leq H\in\mathbb{N}$, we have that*

$$\mid\mathcal{H}_{n,m}(s,H)\mid=(2H)^{nm}\prod_{p\mid s}\left(1-\prod_{i=o}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)+O\left(\frac{(2H)^{nm-1}}{s^{m-n}}\right).$$

To simplify this we can write

$$g(s)=\prod_{p\mid s}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right).$$

Note that since $\sum\limits_{p>H}g(p)=O\left(\frac{1}{H^{m-n}}\right)$, we also have that

$$\sum_{s>H}g(s)=O\left(\frac{2^{\omega(s)}}{H^{m-n}}\right).$$

Let us denote by $\psi(A)$ the number of primes, such that the matrix A has not full rank modulo this prime, *i.e.*,

$$\psi(A)=\mid\{p\in\mathcal{P}\mid\ \mathrm{rk}(A\mod p)\neq n\}\mid.$$

Lemma 11.4.4. *Let $H \in \mathbb{N}$, then*

$$\sum_{A \in R^C_{n,m}(H)} \psi(A) = (2H)^{nm}\alpha + O\left(H^{nm-1}\right).$$

Proof. Observe that

$$\sum_{A \in R^C_{n,m}(H)} \psi(A) = \sum_{p \in \mathcal{P},\ p \leq H} |\ \mathcal{H}_{n,m}(p,H)\ |\ .$$

Now, we can apply Lemma 11.4.1 and get

$$\begin{aligned}
&\sum_{A \in R^C_{n,m}(H)} \psi(A) \\
=&\sum_{p \in \mathcal{P},\ p \leq H} \left((2H)^{nm} \left(1 - \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}} \right) \right) + O\left(\frac{(2H)^{nm-1}}{p^{m-n}} \right) \right) \\
=&(2H)^{nm} \sum_{p \in \mathcal{P},\ p \leq H} \left(1 - \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}} \right) \right) + \sum_{p \in \mathcal{P},\ p \leq H} O\left(\frac{(2H)^{nm-1}}{p^{m-n}} \right) \\
=&(2H)^{nm} \left(\sum_{p \in \mathcal{P}} \left(1 - \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}} \right) \right) \right. \\
&\left. - \sum_{p \in \mathcal{P},\ p > H} \left(1 - \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}} \right) \right) \right) + O\left(H^{nm-1}\right) \\
=&(2H)^{nm} \sum_{p \in \mathcal{P}} \left(1 - \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}} \right) \right) \\
&- (2H)^{nm} \sum_{p \in \mathcal{P},\ p > H} \frac{1}{p^{nm}} \left(p^{nm} - \prod_{i=0}^{n-1} \left(p^m - p^i \right) \right) + O\left(H^{nm-1}\right) \\
=&(2H)^{nm} \sum_{p \in \mathcal{P}} \left(1 - \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}} \right) \right) + (2H)^{nm} O\left(\frac{1}{H^{m-n}} \right) + O\left(H^{nm-1}\right) \\
=&(2H)^{nm}\alpha + O\left(H^{nm-1}\right).
\end{aligned}$$

□

For $A \in \mathbb{Z}^{n \times m}$, let us define

$$\tau(A,p) = \begin{cases} 1 & \text{if } \operatorname{rk}(A \mod p) \neq n \\ 0 & \text{else.} \end{cases}$$

Lemma 11.4.5. *For $H \in \mathbb{N}$, we have that*

$$\sum_{A \in R^C_{n,m}(H)} \psi(A)^2 = (2H)^{nm}(\alpha + \alpha^2 - \beta) + O\left(H^{nm-1}\right).$$

Proof. Since again we have that $\psi(A) = \sum_{p\in\mathcal{P}} \tau(A,p)$, we can write

$$\begin{aligned}\sum_{A\in R^C_{n,m}(H)} \psi(A)^2 &= \sum_{A\in R^C_{n,m}(H)} \left(\sum_{p\in\mathcal{P}} \tau(A,p)\right)^2 \\ &= \sum_{A\in R^C_{n,m}(H)} \left(\sum_{p,q\in\mathcal{P}} \tau(A,p)\tau(A,q)\right) \\ &= \sum_{p,q\in\mathcal{P}} \sum_{A\in R^C_{n,m}(H)} \tau(A,p)\tau(A,q).\end{aligned}$$

We can again split this sum into $p = q$ and $p \neq q$, where in the first case observe that $\tau(A,p)^2 = \tau(A,p)$.

$$\sum_{A\in R^C_{n,m}(H)} \psi(A)^2 = \sum_{p\in\mathcal{P}} \sum_{A\in R^C_{n,m}(H)} \tau(A,p) + \sum_{p\neq q\in\mathcal{P}} \sum_{A\in R^C_{n,m}(H)} \tau(A,p)\tau(A,q).$$

Further, we observe that

$$\sum_{A\in R^C_{n,m}(H)} \tau(A,p) = \mid \mathcal{H}_{n,m}(p,H) \mid$$

and

$$\sum_{A\in R^C_{n,m}(H)} \tau(A,p)\tau(A,q) = \mid \mathcal{H}_{n,m}(pq,H) \mid .$$

Therefore, we have that

$$\sum_{A\in R^C_{n,m}(H)} \psi(A)^2 = \sum_{p\in\mathcal{P}, p\leq H} \mid \mathcal{H}_{n,m}(p,H) \mid + \sum_{p\neq q\in\mathcal{P}, pq\leq H} \mid \mathcal{H}_{n,m}(pq,H) \mid .$$

Using Lemma 11.4.4, we know that

$$\sum_{p\in\mathcal{P}, p\leq H} \mid \mathcal{H}_{nm,}(p,H) \mid = (2H)^{nm}\alpha + O\left(H^{nm-1}\right).$$

By Lemma 11.4.2, we know that

$$\begin{aligned}
&\sum_{p\neq q\in\mathcal{P},\ pq\leq H} |\ \mathcal{H}_{n,m}(pq,H)\ | \\
=&\sum_{p\neq q\in\mathcal{P},\ pq\leq H}\left((2H)^{nm}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{q^{m-i}}\right)\right)\right. \\
&\left.+O\left(\frac{(2H)^{nm-1}}{(pq)^{m-n}}\right)\right) \\
=&(2H)^{nm}\sum_{p\neq q\in\mathcal{P},\ pq\leq H}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{q^{m-i}}\right)\right) \\
&+\sum_{p\neq q\in\mathcal{P},\ pq\leq H}O\left(\frac{(2H)^{nm-1}}{(pq)^{m-n}}\right) \\
=&(2H)^{nm}\sum_{p\neq q\in\mathcal{P},\ pq\leq H}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{q^{m-i}}\right)\right) \\
&+O\left(H^{nm-1}\right) \\
=&(2H)^{nm}\sum_{p,q\in\mathcal{P},\ pq\leq H}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{q^{m-i}}\right)\right) \\
&-(2H)^{nm}\sum_{p\in\mathcal{P},\ p^2\leq H}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)^2+O\left(H^{nm-1}\right).
\end{aligned}$$

Note that the first summand can be written as

$$\begin{aligned}
&(2H)^{nm}\sum_{p,q\in\mathcal{P},\ pq\leq H}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{q^{m-i}}\right)\right) \\
=&(2H)^{nm}\sum_{p,q\in\mathcal{P}}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{q^{m-i}}\right)\right) \\
&-(2H)^{nm}\sum_{p,q\in\mathcal{P},\ pq>H}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{q^{m-i}}\right)\right) \\
=&(2H)^{nm}\sum_{p,q\in\mathcal{P}}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{q^{m-i}}\right)\right) \\
&+(2H)^{nm}O\left(\frac{1}{H^{m-n}}\right) \\
=&(2H)^{nm}\left(\sum_{p\in\mathcal{P}}\left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)\right)^2+O\left(H^{nm-m+n}\right) \\
=&(2H)^{nm}\alpha^2+O\left(H^{nm-m+n}\right).
\end{aligned}$$

Further, the second summand can be written as

$$\begin{aligned}
&(2H)^{nm} \sum_{p\in\mathcal{P},\ p^2\leq H} \left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)^2 \\
=&(2H)^{nm} \sum_{p\in\mathcal{P}} \left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)^2 \\
&-(2H)^{nm} \sum_{p\in\mathcal{P},\ p>\sqrt{H}} \left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)^2 \\
=&(2H)^{nm} \sum_{p\in\mathcal{P}} \left(1-\prod_{i=0}^{n-1}\left(1-\frac{1}{p^{m-i}}\right)\right)^2 - (2H)^{nm} O\left(\frac{1}{H^{m-n+1/2}}\right) \\
=&(2H)^{nm}\beta + O\left(H^{nm-m+n-1/2}\right).
\end{aligned}$$

Thus, we can conclude, as

$$\begin{aligned}
\sum_{A\in R_{n,m}^C(H)} \psi(A)^2 =&(2H)^{nm}\alpha + (2H)^{nm}\alpha^2 - (2H)^{nm}\beta + O\left(H^{nm-1}\right) \\
=&(2H)^{nm}(\alpha+\alpha^2-\beta) + O\left(H^{nm-1}\right).
\end{aligned}$$

□

Theorem 11.4.6. *For $H \in \mathbb{N}$, we have that*

$$\mid R_{n,m}^C(H) \mid = (2H)^{nm}\rho(R_{n,m}^C) + O\left(H^{nm-1}\right).$$

Proof. Using the inclusion-exclusion principle, we get that

$$\mid R_{n,m}^C(H) \mid = -\sum_{s=2}^{H} \mu(s) \mid \mathcal{H}_{n,m}(s,H) \mid .$$

With the considerations before and Lemma 11.4.3, we can write this as

$$\begin{aligned}
\mid R_{n,m}^{C}(H) \mid &= -\sum_{s=2}^{H} \mu(s) \left((2H)^{nm} g(s) + O\left(\frac{(2H)^{nm-1}}{s^{m-n}} \right) \right) \\
&= -\sum_{s=2}^{H} \mu(s)(2H)^{nm} g(s) + \sum_{s=2}^{H} O\left(\frac{(2H)^{nm-1}}{s^{m-n}} \right) \\
&= -(2H)^{nm} \sum_{s=2}^{H} \mu(s) g(s) + O\left(H^{nm-1} \right) \\
&= -(2H)^{nm} \sum_{s\geq 2} \mu(s) g(s) + (2H)^{nm} \sum_{s\geq H+1} \mu(s) g(s) + O\left(H^{nm-1} \right) \\
&= -(2H)^{nm} \sum_{s\geq 2} \mu(s) g(s) + (2H)^{nm} O\left(\frac{2^{\omega(s)}}{H^{m-n}} \right) + O\left(H^{nm-1} \right) \\
&= -(2H)^{mn} \sum_{s\geq 2} \mu(s) g(s) + O\left(H^{nm-1} \right).
\end{aligned}$$

Since $g(s)$ is a multiplicative function, we have that

$$\begin{aligned}
-\sum_{s\geq 2} \mu(s) \frac{g(s)}{s^{mn}} &= 1 - \sum_{s\geq 1} \mu(s) g(s) \\
&= 1 - \prod_{p\in\mathcal{P}} (1 - g(p)) \\
&= 1 - \prod_{p\in\mathcal{P}} \prod_{i=0}^{n-1} \left(1 - \frac{1}{p^{m-i}} \right) \\
&= 1 - \prod_{i=0}^{n-1} \frac{1}{\zeta(m-i)} \\
&= \rho(R_{n,m}^{C}).
\end{aligned}$$

Hence, we get the claim

$$\begin{aligned}
\mid R_{n,m}^{C}(H) \mid &= (2H)^{nm} \left(-\sum_{s\geq 2} \mu(s) g(s) \right) + O\left(H^{nm-1} \right) \\
&= (2H)^{nm} \rho(R_{n,m}^{C}) + O\left(H^{nm-1} \right).
\end{aligned}$$

□

Note, that this can be considered as an alternative proof of the density of rectangular unimodular matrices.

We are ready to state the main theorem of this section.

Theorem 11.4.7. *The mean and the variance of the function $\psi(A)$, where A ranges over all rectangular non-unimodular matrices are*

$$\mu = \lim_{H\to\infty} \frac{\sum\limits_{A\in R^C_{n,m}(H)} \psi(A)}{\mid R^C_{n,m}(H) \mid} = \frac{\alpha}{\rho(R^C_{n,m})},$$
$$\sigma^2 = \lim_{H\to\infty} \frac{\sum\limits_{A\in R^C_{n,m}(H)} (\psi(A)-\mu)^2}{\mid R^C_{n,m}(H) \mid} = \frac{\alpha+\alpha^2-\beta-\mu\alpha}{\rho(R^C_{n,m})}.$$

Proof. We can use Lemma 11.4.4 and Theorem 11.4.6 to compute the mean:

$$\begin{aligned}\mu &= \lim_{H\to\infty} \frac{\sum\limits_{A\in R^C_{n,m}(H)} \psi(A)}{\mid R^C_{n,m}(H) \mid} \\ &= \lim_{H\to\infty} \frac{(2H)^{nm}\alpha + O\left(H^{nm-1}\right)}{(2H)^{nm}\rho(R^C_{n,m}) + O\left(H^{nm-1}\right)} \\ &= \frac{\alpha}{\rho(R^C_{n,m})}.\end{aligned}$$

Further, from Lemma 11.4.4, Lemma 11.4.5 and Theorem 11.4.6, we can compute the variance, as

$$\begin{aligned}\sigma^2 &= \lim_{H\to\infty} \frac{\sum\limits_{A\in R^C_{n,m}(H)} (\psi(A)-\mu)^2}{\mid R^C_{n,m}(H) \mid} \\ &= \lim_{H\to\infty} \frac{\sum\limits_{A\in R^C_{n,m}(H)} (\psi(A)^2 - 2\mu\psi(A) + \mu^2)}{\mid R^C_{n,m}(H) \mid} \\ &= \lim_{H\to\infty} \frac{\sum\limits_{A\in R^C_{n,m}(H)} \psi(A)^2 - 2\mu \sum\limits_{A\in R^C_{n,m}(H)} \psi(A) + \mu^2 \mid R^C_{n,m}(H) \mid}{\mid R^C_{n,m}(H) \mid} \\ &= \lim_{H\to\infty} \frac{(2H)^{nm}\left(\alpha+\alpha^2-\beta-2\mu\alpha+\mu^2\rho(R^C_{n,m})\right) + O\left(H^{nm-1}\right)}{(2H)^{nm}\rho(R^C_{n,m}) + O\left(H^{nm-1}\right)} \\ &= \frac{\alpha+\alpha^2-\beta-2\mu\alpha+\mu^2\rho(R^C_{n,m})}{\rho(R^C_{n,m})} \\ &= \frac{\alpha+\alpha^2-\beta-2\mu\alpha+\mu\frac{\alpha}{\rho(R^C_{n,m})}\rho(R^C_{n,m})}{\rho(R^C_{n,m})} \\ &= \frac{\alpha+\alpha^2-\beta-\mu\alpha}{\rho(R^C_{n,m})}.\end{aligned}$$

□

Chapter 12

Addendum to the Local to Global Principle

We can observe, that the mean and the variance we have computed in Sections 11.2, 11.3, and 11.4 always follow a certain rule, *i.e.*,

$$\mu = \lim_{H\to\infty} \frac{\sum\limits_{a\in T(H)} \psi(a)}{\mid T(H) \mid} = \frac{\alpha}{\rho(T)},$$
$$\sigma^2 = \lim_{H\to\infty} \frac{\sum\limits_{a\in T(H)} (\psi(a)-\mu)^2}{\mid T(H) \mid} = \frac{\alpha+\alpha^2-\beta-\mu\alpha}{\rho(T)},$$

for some α and β. Now, if we compare the considered α and β with the Sections 10.1, 10.2, and 10.3, we can observe that

$$\alpha = \sum_{\nu\in M_{\mathbb{Q}}} s_\nu$$

and

$$\beta = \sum_{\nu\in M_{\mathbb{Q}}} s_\nu^2,$$

where we usually have $s_\infty = 0$ and can thus only consider $s_p = \mu_p(U_p)$. Thus, there is a direct connection of the mean and the variance with the local to global principle and the $U_p \subset \mathbb{Z}_p$ we have chosen there.

In what follows, we will present an addendum to the local to global principle by adding a few conditions on U_p to make sure that the mean and the variance exist and then show that the mean and the variance can be computed through s_p.

We adapt the definition of mean and variance: in the section before we have considered

$$\mu = \lim_{H\to\infty} \frac{\sum\limits_{a\in T(H)} \psi(a)}{\mid T(H) \mid},$$

and

$$\sigma^2 = \lim_{H\to\infty} \frac{\sum\limits_{a\in T(H)} (\psi(a)-\mu)^2}{\mid T(H) \mid},$$

where $\psi(a)$ was defined to be the number of primes for which a satisfies the condition C_p. In order to define the mean and the variance of the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$, we hence use

$$\mid \{\nu \in M_\mathbb{Q} \mid A \in U_\nu\} \mid$$

instead of $\psi(a)$.

More in detail, we will define the expected value of the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ to be

$$\mu = \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[^d} \mid \{\nu \in M_\mathbb{Q} \mid A \in U_\nu\} \mid}{(2H)^d},$$

if it exists and the variance of the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ to be

$$\sigma^2 = \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[^d} (\mid \{\nu \in M_\mathbb{Q} \mid A \in U_\nu\} \mid -\mu)^2}{(2H)^d},$$

if it exists.

One can think of these definitions as the expected value and the variance of the "random variable" that counts how many times an element is expected to be in one of the U_ν's.

Further, we will exclude all A's which lie in infinitely many U_ν's, as they have density zero, and should thus not contribute to the sum.

We then introduce the mean and the variance of the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ restricted to some target set T and we show that the results we get from the addendum to the local to global principle coincide with the computations of the mean and variance we have done before.

In fact, we get that the mean of the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ is given by

$$\sum_{\nu\in M_\mathbb{Q}} s_\nu = \alpha,$$

the variance of the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ is given by

$$\sum_{\nu\in M_\mathbb{Q}} s_\nu - \sum_{\nu\in M_\mathbb{Q}} s_\nu^2 = \alpha - \beta,$$

and when we restrict to the target set, we get that the restricted mean is given by

$$\mu_T = \frac{\mu}{\rho(T)} = \frac{\alpha}{\rho(T)},$$

and the restricted variance is

$$\sigma_T^2 = \frac{\mu + \mu^2 - \sum_{\nu\in M_\mathbb{Q}} s_\nu^2 - \mu_T\mu}{\rho(T)} = \frac{\alpha + \alpha^2 - \beta - \mu_T\alpha}{\rho(T)},$$

which are exactly as in the computations before.

We have thus found a more elegant and faster way to compute the mean and the variance corresponding to the density.

12.1 Main Theorem

For a fixed $\nu \in M_{\mathbb{Q}}$ and U_ν as in Theorem 9.3.8, the density of $U_\nu \cap \mathbb{Z}^d$ can be computed as follows.

Corollary 12.1.1. *Let $\nu \in M_{\mathbb{Q}}$ and U_ν be chosen as in Theorem 9.3.8, then*

$$\rho(U_\nu \cap \mathbb{Z}^d) = \mu_\nu(U_\nu) = s_\nu.$$

Proof. Simply set in Theorem 9.3.8

$$U'_{\nu'} = \begin{cases} U_\nu & \nu' = \nu, \\ \emptyset & \nu' \neq \nu, \end{cases}$$

and let

$$\begin{aligned} P' : \mathbb{Z}^d &\to 2^{M_{\mathbb{Q}}}, \\ a &\mapsto \{\nu \in M_{\mathbb{Q}} \mid a \in U'_\nu\}. \end{aligned}$$

Then, compute $\rho(P'^{-1}(\{\nu\}))$. □

Similarly, for $\nu, \eta \in M_{\mathbb{Q}}$ with $\nu \neq \eta$, U_ν and U_η chosen as in Theorem 9.3.8, the density of $(U_\nu \cap \mathbb{Z}^d) \cap (U_\eta \cap \mathbb{Z}^d)$ can be computed.

Corollary 12.1.2. *Let $\nu, \eta \in M_{\mathbb{Q}}$ with $\nu \neq \eta$ and U_ν, U_η be chosen as in Theorem 9.3.8, then*

$$\rho((U_\nu \cap \mathbb{Z}^d) \cap (U_\eta \cap \mathbb{Z}^d)) = \mu_\nu(U_\nu)\mu_\eta(U_\eta) = s_\nu s_\eta.$$

Proof. Simply set in Theorem 9.3.8

$$U'_{\nu'} = \begin{cases} U_\nu & \nu' = \nu, \\ U_\eta & \nu' = \eta, \\ \emptyset & \text{else}, \end{cases}$$

and let

$$\begin{aligned} P' : \mathbb{Z}^d &\to 2^{M_{\mathbb{Q}}}, \\ a &\mapsto \{\nu \in M_{\mathbb{Q}} \mid a \in U'_\nu\}. \end{aligned}$$

Then, compute $\rho(P'^{-1}(\{\nu, \eta\}))$. □

We observe that the elements $A \in \mathbb{Z}^d$, which are in U_ν for infinitely many $\nu \in M_{\mathbb{Q}}$ have density zero.

Lemma 12.1.3. *Let U_ν be chosen as in Theorem 9.3.8 for all $\nu \in M_{\mathbb{Q}}$. Then, we have that*

$$\rho(\{A \in \mathbb{Z}^d \mid A \in U_\nu \text{ for infinitely many } \nu \in M_{\mathbb{Q}}\}) = 0.$$

Proof. Since U_ν were chosen as in Theorem 9.3.8, Condition (9.3.1) holds, *i.e.*,

$$\lim_{M\to\infty} \bar{\rho}\left(\left\{A \in \mathbb{Z}^d \mid A \in U_p \text{ for some prime } p > M\right\}\right) = 0.$$

Let us call

$$C_M = \left\{A \in \mathbb{Z}^d \mid A \in U_p \text{ for some prime } p > M\right\},$$
$$I = \left\{A \in \mathbb{Z}^d \mid A \in U_\nu \text{ for infinitely many } \nu \in M_{\mathbb{Q}}\right\}.$$

Clearly, $I \subset C_M$ for all $M \in \mathbb{N}$, hence

$$\rho(I) \leq \lim_{M\to\infty} \overline{\rho}(C_M) = 0.$$

□

The usual definition of expected value does not take into account the events of probability zero. Over $\mathbb{Z}^d$, we will use the density to give an analogue definition. Observe that an event with density zero will not have any influence on the expected value, this legitimates that over $\mathbb{Z}^d$ we will exclude the elements $A \in \mathbb{Z}^d$, which are in infinitely many U_ν, namely $A \in I$. Let us define

$$[-H, H[_I^d = ([-H, H[^d \cap \mathbb{Z}^d) \setminus I.$$

Definition 12.1.4. Let H and d be positive integers and assume that $(U_\nu)_{\nu\in M_{\mathbb{Q}}}$ satisfy the assumptions of Theorem 9.3.8, then we define *the expected value of the system* $(U_\nu)_{\nu\in M_{\mathbb{Q}}}$ to be

$$\mu = \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[_I^d} |\{\nu \in M_{\mathbb{Q}} \mid A \in U_\nu\}|}{(2H)^d},$$

if it exists.

This limit essentially gives the expected value of the number of places ν, such that a random element in $\mathbb{Z}^d$ is in U_ν. We will sometimes refer to the expected value of the system $(U_\nu)_{\nu\in M_{\mathbb{Q}}}$ as the mean of the system $(U_\nu)_{\nu\in M_{\mathbb{Q}}}$.

Definition 12.1.5. For a set T, for which we can compute its density via the local to global principle as in Theorem 9.3.8, we say that a *system* $(U_\nu)_{\nu\in M_{\mathbb{Q}}}$ *corresponds to* T, if $T^C = P^{-1}(\{\emptyset\})$.

Observe, that we can restrict Definition 12.1.4 to subsets, *i.e.*, we define the expected value of the system $(U_\nu)_{\nu\in M_{\mathbb{Q}}}$ restricted to T to be

$$\mu_T = \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[_I^d\cap T} |\{\nu \in M_{\mathbb{Q}} \mid A \in U_\nu\}|}{|[-H, H[_I^d \cap T|},$$

if it exists. Note that this is similar to the conditional expected value.

One can easily pass from μ to μ_T and viceversa:

Lemma 12.1.6. *If the density of T exists and is nonzero and T is such that $T^C \subseteq P^{-1}(\{\emptyset\})$, then μ_T exists and is given by $\mu = \mu_T \rho(T)$.*

Proof. We observe that

$$\begin{aligned}\mu_T &= \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[_I^d\cap T} |\,\{\nu\in M_{\mathbb{Q}} \mid A\in U_\nu\}\,|}{|\,[-H,H[_I^d\cap T\,|} \\ &= \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[_I^d\cap T} |\,\{\nu\in M_{\mathbb{Q}} \mid A\in U_\nu\}\,|}{(2H)^d} \frac{(2H)^d}{|\,[-H,H[_I^d\cap T\,|} \\ &= \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[_I^d\cap T} |\,\{\nu\in M_{\mathbb{Q}} \mid A\in U_\nu\}\,|}{(2H)^d} \frac{1}{\rho(T)}.\end{aligned}$$

Let us define

$$\tau(A,\nu) = \begin{cases} 1 & A\in U_\nu, \\ 0 & \text{else.}\end{cases}$$

Note that one can write $[-H,H[_I^d\cap T$ as $[-H,H[_I^d\backslash([-H,H[_I^d\cap T^C)$, doing so we observe that we can ignore the T:

$$\begin{aligned}\mu_T\rho(T) &= \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[_I^d\cap T}\sum\limits_{\nu\in M_{\mathbb{Q}}} \tau(A,\nu)}{(2H)^d} \\ &= \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[_I^d}\sum\limits_{\nu\in M_{\mathbb{Q}}} \tau(A,\nu) - \sum\limits_{A\in[-H,H[_I^d\cap T^C}\sum\limits_{\nu\in M_{\mathbb{Q}}} \tau(A,\nu)}{(2H)^d}.\end{aligned}$$

If $T^C \subseteq P^{-1}(\{\emptyset\})$, then it holds that $\tau(A,\nu) = 0$ for all $A\in T^C$, hence we are left with μ. □

In the applications one usually chooses $T^C = P^{-1}(\{\emptyset\})$ and computes the expected value restricted to T. This is a natural choice, since by the definition of T^C none of its elements lie in any of the U_ν, thus we are only considering the subset T, where nonzero values are added to the expected value.

Similarly, we define the variance over $\mathbb{Z}^d$.

Definition 12.1.7. Let H and d be a positive integers and for all $\nu\in M_{\mathbb{Q}}$ we define U_ν as in Theorem 9.3.8, then we define *the variance of the sytem* $\{U_\nu\}_{\nu\in M_{\mathbb{Q}}}$ to be

$$\sigma^2 = \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[_I^d} (|\,\{\nu\in M_{\mathbb{Q}} \mid A\in U_\nu\}\,| - \mu)^2}{|\,[-H,H[_I^d|},$$

if it exists.

We can restrict this definition to subsets, *i.e.*, we define the variance of the system $(U_\nu)_{\nu\in M_{\mathbb{Q}}}$ restricted to T to be

$$\sigma_T^2 = \lim_{H\to\infty} \frac{\sum_{A\in[-H,H[_I^d\cap T} (\mid \{\nu\in M_{\mathbb{Q}} \mid A\in U_\nu\}\mid -\mu_T)^2}{\mid [-H,H[_I^d\cap T\mid},$$

if it exists.

Note that Condition (9.3.1) of Theorem 9.3.8 is not enough to ensure the existence of the mean, for this see Example 10 in [79].

Now we are ready to state the main theorem, which is an addendum to the local to global principle by Poonen and Stoll (9.3.8) in [91]. Note that the part on the mean can be found in [79].

Theorem 12.1.8. *Let H and d be positive integers. Let $U_\infty\subset\mathbb{R}^d$, such that $\mathbb{R}_{\geq 0}\cdot U_\infty = U_\infty$ and $\mu_\infty(\partial(U_\infty)) = 0$. Let $s_\infty = \frac{1}{2^d}\mu_\infty(U_\infty\cap[-1,1]^d)$. For each prime p, let $U_p\subset\mathbb{Z}_p^d$, such that $\mu_p(\partial(U_p)) = 0$ and define $s_p = \mu_p(U_p)$. Define the following map*

$$\begin{aligned} P:\mathbb{Z}^d &\to 2^{M_{\mathbb{Q}}},\\ a &\mapsto \{\nu\in M_{\mathbb{Q}} \mid a\in U_\nu\}. \end{aligned}$$

We assume that Condition (9.3.1) is satisfied and that for some $\alpha\in[0,\infty)$, there exists an absolute constant $c\in\mathbb{Z}$, and some $m\in\mathbb{N}$ such that for all $H\geq 1$ and for all $A\in\mathbb{Z}^d$

$$\ell_{A,H} = \mid \{p\in\mathcal{P} \mid p > H^\alpha, A\in U_p\cap[-H,H[_I^d\} \mid < c \tag{12.1.1}$$

and there exists a sequence $(v_p)_{p\in\mathcal{P}}$, such that for all $p < H^\alpha$ holds

$$\mid U_p\cap[-H,H[_I^d\mid \leq v_p(2H)^d, \tag{12.1.2}$$

$$\sum_{p\in\mathcal{P}} v_p \quad \text{converges.} \tag{12.1.3}$$

Then the following hold:

1. *The mean of the system $(U_\nu)_{\nu\in M_{\mathbb{Q}}}$ exists and is given by*

$$\mu = \sum_{\nu\in M_{\mathbb{Q}}} s_\nu. \tag{12.1.4}$$

2. *If in addition, there exists some $\beta\in[0,\infty)$, we have that there exists an absolute constant $c'\in\mathbb{Z}$, and some $m'\in\mathbb{N}$ such that for all $H\geq 1$ and for all $A\in\mathbb{Z}^d$*

$$r_{A,H} = \mid \{p\in\mathcal{P} \mid p > H^\beta, A\in U_p\cap[-H,H[_I^d\} \mid < c' \tag{12.1.5}$$

and there exists a sequence $(\tilde{v}_p)_{p\in\mathcal{P}}$, such that for all $p,q < H^\beta$ holds

$$\mid (U_p\cap U_q)\cap[-H,H[_I^d\mid \leq \tilde{v}_p\tilde{v}_q(2H)^d, \tag{12.1.6}$$

$$\sum_{p\in\mathcal{P}} \tilde{v}_p \quad \text{converges,} \tag{12.1.7}$$

then the variance of the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$*, exists and is given by:*

$$\sigma^2 = \sum_{\nu\in M_\mathbb{Q}} s_\nu - \sum_{\nu\in M_\mathbb{Q}} s_\nu^2. \tag{12.1.8}$$

3. *If in addition, we have some* $T \subseteq \mathbb{Z}^d$ *such that* $T^C \subseteq P^{-1}(\{\emptyset\})$ *and* $\rho(T)$ *exists and is not zero, then* σ_T^2 *exists and is given by:*

$$\sigma_T^2 = \frac{\mu + \mu^2 - \sum_{\nu\in M_\mathbb{Q}} s_\nu^2 - \mu_T\mu}{\rho(T)}. \tag{12.1.9}$$

Proof. Let us first consider the mean μ.

1. For $H, M > 0$ we split

$$\sum_{A\in[-H,H[_I^d} \frac{|\{\nu\in M_\mathbb{Q} \mid A\in U_\nu\}|}{(2H)^d} = s_1(H,M) + s_2(H,M) + s_3(H,M),$$

where

$$s_1(H,M) = \sum_{A\in[-H,H[_I^d} \frac{|\{p\in\mathcal{P} \mid M\le H^\alpha < p, A\in U_p\}|}{(2H)^d},$$
$$s_2(H,M) = \sum_{A\in[-H,H[_I^d} \frac{|\{p\in\mathcal{P} \mid M< p < H^\alpha, A\in U_p\}|}{(2H)^d},$$
$$s_3(H,M) = \sum_{A\in[-H,H[_I^d} \frac{|\{\nu\in\mathcal{P} \mid \nu=\infty \text{ or } \nu\le M, A\in U_\nu\}|}{(2H)^d}.$$

We are going to show that for $j\in\{1,2\}$ we have

$$\limsup_{M\to\infty}\limsup_{H\to\infty}|s_j(H,M)| = 0$$

and

$$\lim_{M\to\infty}\lim_{H\to\infty} s_3(H,M) = \sum_{\nu\in M_\mathbb{Q}} s_\nu,$$

which readily implies that

$$\mu = \lim_{H\to\infty} \frac{\sum_{A\in[-H,H[_I^d} |\{\nu\in M_\mathbb{Q} \mid A\in U_\nu\}|}{(2H)^d}$$

exists and that

$$\mu = \sum_{\nu\in M_\mathbb{Q}} s_\nu.$$

First we consider the case $\alpha \neq 0$. From Condition (12.1.1) we have that for $H > 0$ and $A \in \mathbb{Z}^d$

$$\ell_{A,H} =\mid \left\{p \in \mathcal{P} \mid p > H^\alpha, A \in U_p \cap [-H, H[_I^d\right\} \mid < c. \tag{12.1.10}$$

Therefore, we get

$$\begin{aligned}
0 &\leq \limsup_{M\to\infty} \limsup_{H\to\infty} |s_1(H,M)| \\
&\leq \limsup_{M\to\infty} \limsup_{H\to\infty} \sum_{A\in[-H,H[_I^d \cap \bigcup_{M<p\in\mathcal{P}} U_p} \frac{\ell_{A,H}}{(2H)^d} \\
&\leq \limsup_{M\to\infty} \limsup_{H\to\infty} \sum_{A\in[-H,H[_I^d \cap \bigcup_{M<p\in\mathcal{P}} U_p} \frac{c}{(2H)^d} \\
&= c \limsup_{M\to\infty} \limsup_{H\to\infty} \frac{|[-H,H[_I^d \cap \bigcup_{M<p\in\mathcal{P}} U_p|}{(2H)^d} \\
&= c \limsup_{M\to\infty} \overline{\rho}\left(\bigcup_{M<p\in\mathcal{P}} U_p\right) = 0,
\end{aligned}$$

where the last equality follows from Condition (9.3.1). Using the Conditions (12.1.2) and (12.1.3) we get

$$\begin{aligned}
0 &\leq \limsup_{M\to\infty} \limsup_{H\to\infty} |s_2(H,M)| \\
&= \limsup_{M\to\infty} \limsup_{H\to\infty} \sum_{p\in\mathcal{P}: M<p<H^\alpha} \frac{|U_p \cap [-H,H[_I^d|}{(2H)^d} \\
&\leq \limsup_{M\to\infty} \limsup_{H\to\infty} \sum_{p\in\mathcal{P}: M<p<H^\alpha} v_p = 0.
\end{aligned}$$

For $\alpha = 0$ on the other hand, we have for $M > 1$ that

$$s_1(H,M) = 0 = s_2(H,M).$$

Using Corollary 12.1.1, we get

$$\begin{aligned}
&\lim_{M\to\infty} \lim_{H\to\infty} s_3(H,M) \\
&= \lim_{M\to\infty} \lim_{H\to\infty} \sum_{\nu\in M_\mathbb{Q}: \nu\leq M \text{ or } \nu=\infty} \frac{|\,[-H,H[_I^d \cap U_\nu\,|}{(2H)^d} \\
&= \lim_{M\to\infty} \sum_{\nu\in M_\mathbb{Q}: \nu\leq M \text{ or } \nu=\infty} \rho(U_\nu \cap \mathbb{Z}^d) \\
&= \lim_{M\to\infty} \sum_{\nu\in M_\mathbb{Q}: \nu\leq M \text{ or } \nu=\infty} s_\nu \\
&= \sum_{\nu\in M_\mathbb{Q}} s_\nu.
\end{aligned}$$

2. For the variance σ^2 we are going to use a similar argument. We split for $H > 1$

$$\frac{\sum\limits_{A\in[-H,H[_I^d} (\mid \{\nu \in M_{\mathbb{Q}} \mid A \in U_\nu\} \mid -\mu)^2}{(2H)^d} = T_1(H) + T_2(H), \tag{12.1.11}$$

where

$$T_1(H) = \frac{\sum\limits_{A\in[-H,H[_I^d} \mid \{\nu \in M_{\mathbb{Q}} \mid A \in U_\nu\} \mid^2}{(2H)^d},$$
$$T_2(H) = -2\mu\frac{\sum\limits_{A\in[-H,H[_I^d} \mid \{\nu \in M_{\mathbb{Q}} \mid A \in U_\nu\} \mid}{(2H)^d} + \mu^2\frac{\sum\limits_{A\in[-H,H[_I^d} 1}{(2H)^d}.$$

We can use the same argument as for the mean μ and Lemma 12.1.3 to obtain

$$\lim_{H\to\infty} T_2(H) = -2\mu^2 + \mu^2 = -\mu^2. \tag{12.1.12}$$

We split $T_1(H)$ for $M > 1$ into

$$T_1(H) = V_1(H, M) + V_2(H, M) + V_3(H, M), \tag{12.1.13}$$

where

$$V_1(H, M) = \frac{\sum\limits_{A\in[-H,H[_I^d} (\sum\limits_{\substack{\nu>M,\\ \nu\neq\infty}} \tau(A, \nu))^2}{(2H)^d}$$
$$V_2(H, M) = \frac{\sum\limits_{A\in[-H,H[_I^d} (\sum\limits_{\substack{\nu\leq M,\\ \nu=\infty}} \tau(A, \nu))^2}{(2H)^d},$$
$$V_3(H, M) = \frac{\sum\limits_{A\in[-H,H[_I^d} (\sum\limits_{\substack{\nu>M,\\ \nu\neq\infty}} \tau(A, \nu))(\sum\limits_{\substack{\nu\leq M,\\ \nu=\infty}} \tau(A, \nu))}{(2H)^d},$$

and where $\tau(A, \nu) = 1$ if $A \in U_\nu$ and zero otherwise. We will show that

$$\limsup_{M\to\infty} \limsup_{H\to\infty} V_1(H, M) = 0.$$

For this we split

$$V_1(H, M) = R_1(H, M) + R_2(H, M) + 2R_3(H, M), \tag{12.1.14}$$

where

$$R_1(H,M) = \sum_{\substack{A\in[-H,H[_I^d \\ A\in\cup_{p,q>M}(U_p\cap U_q)}} \frac{|\left\{p,q>H^\beta \mid A\in U_p\cap U_q\right\}|}{(2H)^d},$$

$$R_2(H,M) = \sum_{\substack{A\in[-H,H[_I^d \\ A\in\cup_{p,q>M}(U_p\cap U_q)}} \frac{|\left\{M<p,q<H^\beta \mid A\in U_p\cap U_q\right\}|}{(2H)^d},$$

$$R_3(H,M) = \sum_{\substack{A\in[-H,H[_I^d \\ A\in\cup_{p,q>M}(U_p\cap U_q)}} \frac{|\left\{M<p<H^\beta<q \mid A\in U_p\cap U_q\right\}|}{(2H)^d}.$$

Note that we have omitted to write $(p,q)\in\mathcal{P}^2$ to save space and by abuse of notation we write $A\in U_p\cap U_q$, instead of $A\in(U_p\cap U_q)\cap\mathbb{Z}^d$.

From Condition (12.1.5) we have that

$$r_{A,H} = \left|\left\{p\in\mathcal{P}\mid p>H^\beta, A\in U_p\cap[-H,H[_I^d\right\}\right| < \tilde{c},$$

for some absolute constant $\tilde{c}$. Hence

$$\begin{aligned}
&\limsup_{M\to\infty}\limsup_{H\to\infty} R_1(H,M)\\
=&\limsup_{M\to\infty}\limsup_{H\to\infty} \sum_{\substack{A\in[-H,H[_I^d \\ A\in\cup_{p,q>M}(U_p\cap U_q)}} \frac{|\left\{p,q>H^\beta \mid A\in(U_p\cap U_q)\cap\mathbb{Z}^d\right\}|}{(2H)^d}\\
\le&\limsup_{M\to\infty}\limsup_{H\to\infty} \sum_{\substack{A\in[-H,H[_I^d \\ A\in\cup_{p,q>M}(U_p\cap U_q)}} \frac{r_{A,H}^2}{(2H)^d}\\
\le&\limsup_{M\to\infty}\limsup_{H\to\infty} \sum_{\substack{A\in[-H,H[_I^d \\ A\in\cup_{p,q>M}(U_p\cap U_q)}} \frac{\tilde{c}^2}{(2H)^d}\\
=&\tilde{c}^2\limsup_{M\to\infty}\limsup_{H\to\infty}\frac{|\,[-H,H[_I^d\cap\cup_{p,q>M}(U_p\cap U_q)\,|}{(2H)^d}\\
\le&\tilde{c}^2\limsup_{M\to\infty}\overline{\rho}\left(\bigcup_{p,q>M}(U_p\cap U_q)\right)\\
\le&\tilde{c}^2\limsup_{M\to\infty}\overline{\rho}\left(\bigcup_{p>M}U_p\right)=0.
\end{aligned}$$

As $R_1(H,M)\ge 0$, we get

$$\limsup_{M\to\infty}\limsup_{H\to\infty} R_1(H,M) = 0. \tag{12.1.15}$$

Using (12.1.6) and (12.1.7), we get that

$$\begin{aligned}
&\limsup_{M\to\infty}\limsup_{H\to\infty} R_2(H,M)\\
&=\limsup_{M\to\infty}\limsup_{H\to\infty}\sum_{\substack{A\in[-H,H[_I^d\\ A\in\cup_{p,q>M}(U_p\cap U_q)}}\frac{|\left\{M<p,q<H^\beta \mid A\in U_p\cap U_q\right\}|}{(2H)^d}\\
&=\limsup_{M\to\infty}\limsup_{H\to\infty}\sum_{\substack{(p,q)\in\mathcal{P}^2\\ M<p,q<H^\beta}}\frac{|\,(U_p\cap U_q)\cap[-H,H[_I^d|}{(2H)^d}\\
&\le\limsup_{M\to\infty}\limsup_{H\to\infty}\sum_{\substack{(p,q)\in\mathcal{P}^2\\ M<p,q<H^\beta}}\tilde{v}_p\tilde{v}_q\\
&\le\limsup_{M\to\infty}\left(\sum_{p\in\mathcal{P}:p>M}\tilde{v}_p\right)^2=0.
\end{aligned}$$

Again, as $0\le R_2(H,M)$ we get

$$\limsup_{M\to\infty}\limsup_{H\to\infty} R_2(H,M)=0. \tag{12.1.16}$$

By (12.1.6) we have $|U_p\cap[-H,H[_I^d|=|U_p\cap U_p\cap[-H,H[_I^d|\le\tilde{v}_p^2(2H)^d$. Combining this with $r_{A,H}\le\tilde{c}$, we obtain

$$\begin{aligned}
&\limsup_{M\to\infty}\limsup_{H\to\infty} R_3(H,M)\\
&=\limsup_{M\to\infty}\limsup_{H\to\infty}\sum_{\substack{A\in[-H,H[_I^d\\ A\in\cup_{p,q>M}(U_p\cap U_q)}}\frac{|\left\{M<p<H^\beta<q \mid A\in U_p\cap U_q\right\}|}{(2H)^d}\\
&\le\limsup_{M\to\infty}\limsup_{H\to\infty}\sum_{\substack{A\in[-H,H[_I^d\\ A\in\cup_{p>M}U_p}}\frac{|\left\{M<p<H^\beta \mid A\in U_p\right\}|\cdot|\left\{H^\beta<q \mid A\in U_q\right\}|}{(2H)^d}\\
&\le\tilde{c}\limsup_{M\to\infty}\limsup_{H\to\infty}\sum_{A\in[-H,H[_I^d\cap\cup_{p>M}U_p}\frac{|\left\{p\in\mathcal{P} \mid M<p<H^\beta, A\in U_p\right\}|}{(2H)^d}\\
&\le\tilde{c}\limsup_{M\to\infty}\limsup_{H\to\infty}\sum_{p\in\mathcal{P}:M<p<H^\beta}\frac{|\,U_p\cap[-H,H[_I^d|}{(2H)^d}\\
&\le\tilde{c}\limsup_{M\to\infty}\sum_{p\in\mathcal{P}:M<p}\tilde{v_p}^2=0.
\end{aligned}$$

Where we used (12.1.7) to get that $\sum_{p\in\mathcal{P}}\tilde{v}_p^2$ converges. As $R_3(H,M)\ge 0$ we get

$$\limsup_{M\to\infty}\limsup_{H\to\infty} R_3(H,M)=0. \tag{12.1.17}$$

Combining (12.1.14), (12.1.15), (12.1.16) and (12.1.17) we obtain

$$\limsup_{M\to\infty}\limsup_{H\to\infty} V_1(H,M) = 0. \tag{12.1.18}$$

Now, we deal with V_2. We split

$$V_2(H,M) = W_1(H,M) + W_2(H,M), \tag{12.1.19}$$

where

$$W_1(H,M) = \frac{\sum\limits_{A\in[-H,H[_I^d}\ \sum\limits_{\substack{\nu\le M,\eta\le M,\\ \nu\neq\eta,\\ \nu=\infty\vee\eta=\infty}} \tau(A,\nu)\tau(A,\eta)}{(2H)^d},$$

$$W_2(H,M) = \frac{\sum\limits_{A\in[-H,H[_I^d}\ \sum\limits_{\substack{\nu\le M,\\ \nu=\infty}} \tau(A,\nu)}{(2H)^d}.$$

By the same computation as for the mean, we get

$$\lim_{M\to\infty}\lim_{H\to\infty} W_2(H,M) = \mu. \tag{12.1.20}$$

For W_1 we compute

$$\begin{aligned}
&\lim_{M\to\infty}\lim_{H\to\infty} W_1(H,M)\\
&= \lim_{M\to\infty}\lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[_I^d}\ \sum\limits_{\substack{\nu\le M,\eta\le M,\\ \nu\neq\eta,\\ \nu=\infty\vee\eta=\infty}} \tau(A,\nu)\tau(A,\eta)}{(2H)^d}\\
&= \lim_{M\to\infty} \sum_{\substack{\nu\le M,\eta\le M,\\ \nu\neq\eta,\\ \nu=\infty\vee\eta=\infty}} \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[_I^d} \tau(A,\nu)\tau(A,\eta)}{(2H)^d}\\
&= \lim_{M\to\infty} \sum_{\substack{\nu\le M,\eta\le M,\\ \nu\neq\eta,\\ \nu=\infty\vee\eta=\infty}} \lim_{H\to\infty} \frac{|\,[-H,H[_I^d\cap((U_\nu\cap\mathbb{Z}^d)\cap(U_\eta\cap\mathbb{Z}^d))\,|}{(2H)^d}\\
&= \lim_{M\to\infty} \sum_{\substack{\nu\le M,\eta\le M,\\ \nu\neq\eta,\\ \nu=\infty\vee\eta=\infty}} \rho((U_\nu\cap\mathbb{Z}^d)\cap(U_\eta\cap\mathbb{Z}^d))\\
&= \lim_{M\to\infty} \sum_{\substack{\nu\le M,\eta\le M,\\ \nu\neq\eta,\\ \nu=\infty\vee\eta=\infty}} s_\nu s_\eta = \sum_{\substack{\nu,\eta\in M_{\mathbb{Q}}\\ \nu\neq\eta}} s_\nu s_\eta,
\end{aligned}$$

where we used Corollary 12.1.2 for the second last equality. Hence

$$\lim_{M\to\infty}\lim_{H\to\infty} W_1(H,M) = \mu^2 - \sum_{\nu\in M_{\mathbb{Q}}} s_\nu^2. \tag{12.1.21}$$

Thus, we have by (12.1.19), (12.1.20) and (12.1.21) that

$$\lim_{M\to\infty}\lim_{H\to\infty} V_2(H,M) = \mu^2 - \sum_{\nu\in M_{\mathbb{Q}}} s_\nu^2 + \mu. \tag{12.1.22}$$

Using the Cauchy-Schwarz inequality, we get

$$V_3(H,M) \leq \sqrt{V_1(H,M)}\cdot\sqrt{V_2(H,M)}$$

and thus, we get by (12.1.18) and (12.1.22) that

$$\lim_{M\to\infty}\limsup_{H\to\infty} V_3(H,M) = 0. \tag{12.1.23}$$

By (12.1.11), (12.1.13) we get

$$\begin{aligned}&\frac{\sum\limits_{A\in[-H,H[_I^d} (\mid \{\nu\in M_{\mathbb{Q}} \mid A\in U_\nu\} \mid -\mu)^2}{(2H)^d}\\ =&V_1(H,M)+V_2(H,M)+V_3(H,M)+T_2(H).\end{aligned}$$

By (12.1.18), (12.1.22), (12.1.23) and the fact that $V_1(H,M)\geq 0$ we get that $\lim_{H\to\infty} T_1(H)$ exists and is given by

$$\lim_{H\to\infty} T_1(H) = \mu^2 - \sum_{\nu\in M_{\mathbb{Q}}} s_\nu^2 + \mu. \tag{12.1.24}$$

Hence by (12.1.24) and (12.1.12) we get that σ^2 exists and is given by

$$\begin{aligned}\sigma^2 =& \lim_{H\to\infty} \frac{\sum\limits_{A\in[-H,H[_I^d} (\mid \{\nu\in M_{\mathbb{Q}} \mid A\in U_\nu\} \mid -\mu)^2}{(2H)^d}\\ =& \mu^2 - \sum_{\nu\in M_{\mathbb{Q}}} s_\nu^2 + \mu - \mu^2 = \mu - \sum_{\nu\in M_{\mathbb{Q}}} s_\nu^2.\end{aligned}$$

3. By Lemma 12.1.6, we know that μ_T exists and that we have $\mu = \mu_T\rho(T)$. We split

$$\frac{\sum\limits_{A\in[-H,H[_I^d\cap T} (\mid \{\nu\in M_{\mathbb{Q}} \mid A\in U_\nu\} \mid -\mu_T)^2}{\mid [-H,H[_I^d\cap T \mid} = \Lambda_1(H)+\Lambda_2(H),$$

where

$$\begin{aligned}\Lambda_1(H) =& \sum_{A\in[-H,H[_I^d\cap T} \frac{\mid \{\nu\in M_{\mathbb{Q}} \mid A\in U_\nu\} \mid^2}{|[-H,H]_I^d\cap T|},\\ \Lambda_2(H) =& -2\mu_T \sum_{A\in[-H,H[_I^d\cap T} \frac{\mid \{\nu\in M_{\mathbb{Q}} \mid A\in U_\nu\} \mid}{|[-H,H]_I^d\cap T|}\\ &+\mu_T^2 \sum_{A\in[-H,H[_I^d\cap T} \frac{1}{|[-H,H]_I^d\cap T|}.\end{aligned}$$

Using the condition $T \subseteq P^{-1}(\{\emptyset\})$ and the proof of Lemma 12.1.6 we get

$$\begin{aligned}
\Lambda_1(H) &= \frac{(2H)^d}{|[-H,H]_I^d \cap T|} \sum_{A \in [-H,H[_I^d} \frac{|\{\nu \in M_{\mathbb{Q}} \mid A \in U_\nu\}|^2}{(2H)^d} \\
&= \frac{(2H)^d}{|[-H,H]_I^d \cap T} T_1(H), \\
\Lambda_2(H) &= -2\mu_T \frac{(2H)^d}{|[-H,H]_I^d \cap T|} \sum_{A \in [-H,H[_I^d} \frac{|\{\nu \in M_{\mathbb{Q}} \mid A \in U_\nu\}|}{(2H)^d} \\
&\quad + \mu_T^2 \sum_{A \in [-H,H[_I^d \cap T} \frac{1}{|[-H,H]_I^d \cap T|}.
\end{aligned}$$

Using the computations in (2), yields that σ_T^2 exists and is given by

$$\begin{aligned}
\sigma_T^2 &= \lim_{H \to \infty} \frac{\sum_{A \in [-H,H[_I^d \cap T} (|\{\nu \in M_{\mathbb{Q}} \mid A \in U_\nu\}| - \mu_T)^2}{|[-H,H[_I^d \cap T|} \\
&= \frac{1}{\rho(T)} \left(\mu^2 - \sum_{\nu \in M_{\mathbb{Q}}} s_\nu^2 + \mu \right) - 2\frac{\mu_T}{\rho(T)}\mu + \mu_T^2.
\end{aligned}$$

Which implies (12.1.9).

□

12.2 Examples

We can apply Theorem 12.1.8 to sets, whose densities were computed via the local to global principle of Theorem 9.3.8 and fulfill Conditions (12.1.1), (12.1.2), (12.1.3), (12.1.5), (12.1.6) and (12.1.7).

Coprime Pairs

For example we can compute the expected number of common prime divisors of a non-coprime pair. The rigorous statement reads as follows:

Corollary 12.2.1. *Let us denote by C the set of coprime pairs in $\mathbb{Z}^2$. Then, the corresponding system $(U_\nu)_{\nu \in M_{\mathbb{Q}}}$ is given as in Section 10.1,* i.e., *$U_\infty = \emptyset$ and for $p \in \mathcal{P}$ denote by $U_p = (p\mathbb{Z}_p)^2$.*

Then, the expected value and the variance of the system $(U_\nu)_{\nu \in M_{\mathbb{Q}}}$ exist and are given by

$$\mu = \sum_{\nu \in M_{\mathbb{Q}}} s_\nu = \sum_{p \in \mathcal{P}} \frac{1}{p^2}, \tag{12.2.1}$$

$$\sigma^2 = \mu - \sum_{\nu \in M_{\mathbb{Q}}} s_\nu^2 = \sum_{p \in \mathcal{P}} \frac{1}{p^2} - \sum_{p \in \mathcal{P}} \frac{1}{p^4}. \tag{12.2.2}$$

And the average number and the variance of primes that divide a non-coprime pair are given by

$$\mu_{C^C} = \frac{\sum_{p\in\mathcal{P}} \frac{1}{p^2}}{1-\frac{1}{\zeta(2)}}, \tag{12.2.3}$$

$$\sigma^2_{C^C} = \frac{\sum_{p\in\mathcal{P}} \frac{1}{p^2} + \left(\sum_{p\in\mathcal{P}} \frac{1}{p^2}\right)^2 - \sum_{p\in\mathcal{P}} \frac{1}{p^4} - \left(\sum_{p\in\mathcal{P}} \frac{1}{p^2}\right)^2 \frac{1}{1-\frac{1}{\zeta(2)}}}{1-\frac{1}{\zeta(2)}}, \tag{12.2.4}$$

where ζ denotes the Riemann zeta function.

Proof. Recall, from Section 10.1, that all conditions of Theorem 9.3.8 are satisfied for a corresponding system $(U_\nu)_{\nu\in M_\mathbb{Q}}$, that for $p \in \mathcal{P}$ we have that $s_p = \frac{1}{p^2}$ and thus,

$$\rho(C) = \rho(P^{-1}(\{\emptyset\})) = \frac{1}{\zeta(2)}.$$

Thanks to Lemma 12.1.6, we are left with proving that the additional assumptions on the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ of Theorem 12.1.8 are satisfied. For this example we choose $\alpha = 1$. Observe that $I = \{0\} \subset \mathbb{Z}^2$, since only the zero is in all U_p. Thus, we only exclude that a is zero. Recall, that

$$\ell_{a,H} =\mid \{p \in \mathcal{P} \mid p > H, a \in [-H, H[_I^2 \cap U_p\} \mid .$$

Now, since for $x < H$ nonzero, there are no primes $p > H$ dividing x. Hence, $\ell_{a,H} = 0$ and Condition (12.1.1) is satisfied.

For Condition (12.1.2), we want to show that there exists a sequence $(v_p)_{p\in\mathcal{P}}$, such that for all $p < H$ we have that $\mid U_p \cap [-H, H[_I^2 \mid \leq v_p(2H)^2$. In $\mathbb{F}_p^2$ there is only one element (a_1, a_2) that is divisible by p, *i.e.*, the zero element. If we lift this element to $[-H, H[_I^2$ we get $\left(\left\lceil \frac{2H}{p} \right\rceil\right)^2$ choices, hence in total we have at most

$$\left(\frac{2H}{p} + 1\right)^2 \leq \frac{9H^2}{p^2}$$

elements in $U_p \cap [-H, H[_I^2$. Thus $(v_p)_{p\in\mathcal{P}}$ can be chosen to be $(\frac{3}{p^2})_{p\in\mathcal{P}}$, which also satisfies Condition (12.1.3). Hence, (12.2.1) follows and Lemma 12.1.6 implies (12.2.3).

For this example, we choose $\beta = \frac{1}{2}$. Recall, that

$$r_{a,H} =\mid \{p \in \mathcal{P} \mid p > \sqrt{H}, a \in U_p \cap [-H, H[_I^2\} \mid .$$

For $a \in [-H, H[_I^2$, observe that $a_1, a_2 \leq \sqrt{H}^2$ and

$$\sqrt{H}^{r_{a,H}} \leq \prod_{\substack{p\in\mathcal{P} \\ p>\sqrt{H}, p\mid a}} p,$$

since all $p \mid a$, are such that $p > \sqrt{H}$ and we have $r_{a,H}$ many such prime divisors. Further, observe that

$$\prod_{\substack{p\in\mathcal{P}\\ p>\sqrt{H},\ p|a}} p \leq \gcd(a_1, a_2) \leq \sqrt{H}^2.$$

Thus, we get that $\sqrt{H}^{r_{a,H}} \leq \sqrt{H}^2$, which implies that, for H large enough, $r_{a,H} \leq 2$. Hence, Condition (12.1.5) is satisfied. Next, we prove that Conditions (12.1.6) and (12.1.7) are verified for all $pq \leq H$. Note, that $|U_p \cap U_q \cap [-H, H[_I^2|$ is counting the number of elements (a_1, a_2) which are zero modulo p and modulo q, which in $\mathbb{Z}_{pq}^2$ is just the zero element. Thus, if we lift this element to $[-H, H[_I^2$ we get $\left(\left\lceil \frac{2H}{pq} \right\rceil\right)^2$ choices, hence in total we have at most

$$\left(\frac{2H}{pq} + 1\right)^2 \leq \frac{9H^2}{p^2q^2}$$

elements in $U_p \cap U_q \cap [-H, H[_I^2$. Thus, $(\tilde{v}_p)_{p\in\mathcal{P}}$ can be chosen to be $(\frac{2}{p^2})_{p\in\mathcal{P}}$, which also satisfies Condition (12.1.7). □

Coprime m-Tuples

Another example is the number of common prime divisors of a non-coprime m-tuple.

Corollary 12.2.2. *Let us denote by C_m the set of coprime m-tuples in $\mathbb{Z}^m$. Then, a corresponding system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ is given as in Section 10.1,* i.e., *$U_\infty = \emptyset$ and for $p \in \mathcal{P}$ denote by $U_p = (p\mathbb{Z}_p)^m$.*

Then, the expected value and the variance of the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ exist and are given by

$$\mu = \sum_{\nu\in M_\mathbb{Q}} s_\nu = \sum_{p\in\mathcal{P}} \frac{1}{p^m}, \tag{12.2.5}$$

$$\sigma^2 = \mu - \sum_{\nu\in M_\mathbb{Q}} s_\nu^2 = \sum_{p\in\mathcal{P}} \frac{1}{p^m} - \sum_{p\in\mathcal{P}} \frac{1}{p^{2m}}. \tag{12.2.6}$$

And the average number and the variance of primes that divide a non-coprime m-tuple are given by

$$\mu_{C_m^C} = \frac{\sum_{p\in\mathcal{P}} \frac{1}{p^m}}{1 - \frac{1}{\zeta(m)}}, \tag{12.2.7}$$

$$\sigma^2_{C^C} = \frac{\sum_{p\in\mathcal{P}} \frac{1}{p^m} + \left(\sum_{p\in\mathcal{P}} \frac{1}{p^m}\right)^2 - \sum_{p\in\mathcal{P}} \frac{1}{p^{2m}} - \left(\sum_{p\in\mathcal{P}} \frac{1}{p^m}\right)^2 \frac{1}{1-\frac{1}{\zeta(m)}}}{1 - \frac{1}{\zeta(m)}}, \tag{12.2.8}$$

where ζ denotes the Riemann zeta function.

The proof works exactly in the same manner as for coprime pairs. A small difference is the sequence $(\nu_p)_{p\in\mathcal{P}}$, which in the case of coprime m-tuples can be chosen as $(\frac{4}{p^2})_{p\in\mathcal{P}}$. In fact, the only element in $\mathbb{F}_p^m$, that is divisible by p is the zero element, and lifting the zero element to $[-H,H[_I^m$, we get at most

$$\left(\frac{2H}{p}+1\right)^m \leq \frac{4}{p^2}(2H)^m$$

elements in $U_p \cap [-H,H[_I^m$. Whereas the sequence $(\tilde{v}_p)_{p\in\mathcal{P}}$ can be chosen $(\frac{2}{p^2})_{p\in\mathcal{P}}$. Since we have at most

$$\left(\frac{2H}{pq}+1\right)^m \leq \frac{2}{p^2}\frac{2}{q^2}(2H)^m$$

choices for lifting the zero element in $(\mathbb{Z}/pq\mathbb{Z})^m$ to $[-H,H[_I^m$.

Eisenstein Polynomials

Also the result of [70] follows directly, in a shorter and more elegant way.

Corollary 12.2.3. *Let us denote by E_d the set of Eisenstein polynomials of degree $d>2$ over $\mathbb{Z}$. Then, the corresponding system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ is given as in Section 10.2,* i.e., *$U_\infty=\emptyset$ and for $p\in\mathcal{P}$ denote by*

$$U_p = (p\mathbb{Z}_p \setminus p^2\mathbb{Z}_p)\times(p\mathbb{Z}_p)^{d-1}\times(\mathbb{Z}_p\setminus p\mathbb{Z}_p).$$

Then, the expected value and the variance of the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ exist and are given by

$$\mu = \sum_{\nu\in M_\mathbb{Q}} s_\nu = \sum_{p\in\mathcal{P}} \frac{(p-1)^2}{p^{d+2}}, \tag{12.2.9}$$

$$\sigma^2 = \mu - \sum_{\nu\in M_\mathbb{Q}} s_\nu^2 = \sum_{p\in\mathcal{P}} \frac{(p-1)^2}{p^{d+2}} - \sum_{p\in\mathcal{P}} \frac{(p-1)^4}{p^{2(d+2)}}. \tag{12.2.10}$$

And the average number and the variance of primes for which an Eisenstein polynomial is Eisenstein are given by

$$\mu_{E_d} = \frac{\mu}{1-\prod_{p\in\mathcal{P}}\left(1-\frac{(p-1)^2}{p^{d+2}}\right)}, \tag{12.2.11}$$

$$\sigma^2_{E_d} = \frac{\mu+\mu^2-\sum_{p\in\mathcal{P}}\frac{(p-1)^4}{p^{2(d+2)}} - \mu^2\frac{1}{1-\prod_{p\in\mathcal{P}}\left(1-\frac{(p-1)^2}{p^{d+2}}\right)}}{1-\prod_{p\in\mathcal{P}}\left(1-\frac{(p-1)^2}{p^{d+2}}\right)}. \tag{12.2.12}$$

Proof. Note that in Section 10.2 we have showed that all conditions of Theorem 9.3.8 are satisfied for the corresponding system $(U_\nu)_{\nu\in M_\mathbb{Q}}$. Furthermore, for $p\in\mathcal{P}$ we have that $s_p=\frac{(p-1)^2}{p^{d+2}}$ and thus

$$\rho(E_d) = 1-\prod_{p\in\mathcal{P}}\left(1-\frac{(p-1)^2}{p^{d+2}}\right).$$

Let us choose $\alpha = 1$. Observe that $I = \{0\} \subset \mathbb{Z}^{d+1}$, and thus we only exclude that a is zero. Recall, that

$$\ell_{a,H} = \mid \{p \in \mathcal{P} \mid p > H, a \in [-H, H[_I^{d+1} \cap U_p\} \mid .$$

Now, since for $x < H$ nonzero, the are no primes $p > H$ dividing x. Therefore, we get that Condition (12.1.1) is verified, since $\ell_{a,H} = 0$.

As a next step, we want to show that there exists a sequence $(v_p)_{p\in\mathcal{P}}$, such that for all $p < H$ it holds that $\mid U_p \cap [-H, H[_I^{d+1} \mid \leq v_p(2H)^{d+1}$.

The set $\mid U_p \cap [-H, H[^{d+1}$ can be bounded with Lemma 11.3.2 by

$$\begin{aligned}
&\mid \{(a_0, \dots, a_d) \in [-H, H[_I^{d+1} \mid p \mid a_i \ \forall i < d, p^2 \nmid a_0, p \nmid a_d\} \mid \\
&\leq \frac{(2H)^{d+1}(p-1)^2}{p^{d+2}} + O\left(\frac{H^d}{p^{d-1}} 2^{\omega(p)}\right) \\
&\leq \frac{c}{p^2}(2H)^{d+1},
\end{aligned}$$

for some absolute constant c. Thus, $(v_p)_{p\in\mathcal{P}}$ can be chosen to be $(\frac{c}{p^2})_{p\in\mathcal{P}}$, which also satisfies Condition (12.1.3).

For the variance, we choose $\beta = 1/2$. For $a \in [-H, H[_I^{d+1}$, we can observe again that

$$\sqrt{H}^{r_{a,H}} \leq \prod_{\substack{p\in\mathcal{P} \\ p>\sqrt{H},\ p\mid a_i \forall i<d}} p \leq \gcd(a_0, \dots, a_{d-1}) \leq \sqrt{H}^2.$$

This implies that $\sqrt{H}^{r_{a,H}} \leq \sqrt{H}^2$, which in turn gives $r_{a,H} \leq 2$, for H large enough. Thus, Condition (12.1.5) is satisfied.

Next, we prove that Conditions (12.1.6) and (12.1.7) are verified for all $pq \leq H$.

Note, that the set $\mid U_p \cap U_q \cap [-H, H[_I^{d+1} \mid$ is bounded with Lemma 11.3.2 by

$$\begin{aligned}
&\mid \{(a_0, \dots, a_d) \in [-H, H[_I^{d+1} \mid p \mid a_i \ \forall i < d, p^2 \nmid a_0, p \nmid a_d, \\
&\qquad q \mid a_i \ \forall i < d, q^2 \nmid a_0, q \nmid a_d\} \mid \\
&\leq \frac{(2H)^{d+1}(p-1)^2(q-1)^2}{p^{d+2}q^{d+2}} + O\left(\frac{H^d}{p^{d-1}q^{d-1}} 2^{\omega(pq)}\right) \\
&\leq \frac{c'^2}{p^2q^2}(2H)^{d+1},
\end{aligned}$$

for some absolute constant c'. Thus, $(v_p)_{p\in\mathcal{P}}$ can be chosen to be $(\frac{c'}{p^2})_{p\in\mathcal{P}}$, which also satisfies Condition (12.1.7). With Lemma 12.1.6, we can conclude.

□

The Monic Case The case of monic Eisenstein polynomials is easier than the non-monic case, thus we will only state the result.

Corollary 12.2.4. *Let us denote by M_d the set of monic Eisenstein polynomials of degree $d > 2$ over $\mathbb{Z}$. Then, the corresponding system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ is given as in Section 10.2,* i.e.*, $U_\infty = \emptyset$ and for $p \in \mathcal{P}$ denote by*

$$U_p = (p\mathbb{Z}_p \setminus p^2\mathbb{Z}_p) \times (p\mathbb{Z}_p)^{d-1}.$$

Then, the expected value and the variance of the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ exist and are given by

$$\mu = \sum_{\nu\in M_\mathbb{Q}} s_\nu = \sum_{p\in\mathcal{P}} \frac{p-1}{p^{d+1}}, \tag{12.2.13}$$

$$\sigma^2 = \mu - \sum_{\nu\in M_\mathbb{Q}} s_\nu^2 = \sum_{p\in\mathcal{P}} \frac{p-1}{p^{d+1}} - \sum_{p\in\mathcal{P}} \frac{(p-1)^2}{p^{2(d+1)}}. \tag{12.2.14}$$

And the average number and the variance of primes for which a monic Eisenstein polynomial is Eisenstein are given by

$$\mu_{M_d} = \frac{\mu}{1-\prod_{p\in\mathcal{P}}\left(1-\frac{p-1}{p^{d+1}}\right)}, \tag{12.2.15}$$

$$\sigma^2_{M_d} = \frac{\mu+\mu^2-\sum\limits_{p\in\mathcal{P}}\frac{(p-1)^2}{p^{2(d+1)}} - \mu^2\frac{1}{1-\prod_{p\in\mathcal{P}}\left(1-\frac{p-1}{p^{d+1}}\right)}}{1-\prod_{p\in\mathcal{P}}\left(1-\frac{p-1}{p^{d+1}}\right)}. \tag{12.2.16}$$

The proof is very similar to before, we use Lemma 11.3.7 to bound $\mid U_p\cap[-H,H[_I^d\mid$, as well as $\mid U_p\cap U_q\cap[-H,H[_I^d\mid$.

Rectangular Unimodular Matrices

As last example, we compute the expected number of common prime divisors of all maximal sized minors of a rectangular non-unimodular matrix. We will consider $n>1$, since the case $n=1$ corresponds to the coprime m-tuples.

Corollary 12.2.5. *Let $1<n<m$ be positive integers, and let us denote by R the set of rectangular unimodular matrices in $\mathbb{Z}^{n\times m}$. Then, the corresponding system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ is given as in Section 10.3,* i.e., *$U_\infty=\emptyset$ and for $p\in\mathcal{P}$ denote by U_p the set of all matrices in $\mathbb{Z}_p^{n\times m}$ whose n-minors are all divisible by p.*

Then, the expected value and the variance of the system $(U_\nu)_{\nu\in M_\mathbb{Q}}$ exist and are given by

$$\mu = \sum_{\nu\in M_\mathbb{Q}} s_\nu = \sum_{p\in\mathcal{P}}(1-\prod_{i=0}^{n-1}(1-\frac{1}{p^{m-i}})), \tag{12.2.17}$$

$$\sigma^2 = \mu - \sum_{\nu\in M_\mathbb{Q}} s_\nu^2 = \mu - \sum_{p\in\mathcal{P}}(1-\prod_{i=0}^{n-1}(1-\frac{1}{p^{m-i}}))^2. \tag{12.2.18}$$

And the average number and the variance of primes that divide all n-minors of a

rectangular non-unimodular matrix are given by

$$\mu_{R^C} = \frac{\mu}{1 - \prod_{i=0}^{n-1} \frac{1}{\zeta(m-i)}}, \tag{12.2.19}$$

$$\sigma^2_{R^C} = \frac{\mu + \mu^2 - \sum_{\nu \in M_{\mathbb{Q}}} s_\nu^2 - \mu^2 \frac{1}{1 - \prod_{i=0}^{n-1} \frac{1}{\zeta(m-i)}}}{1 - \prod_{i=0}^{n-1} \frac{1}{\zeta(m-i)}}, \tag{12.2.20}$$

where ζ denotes the Riemann zeta function.

Proof. In Section 10.3 we have showed that all conditions of Theorem 9.3.8 are satisfied for the corresponding system $(U_\nu)_{\nu \in M_{\mathbb{Q}}}$. Further, we have showed that for $p \in \mathcal{P}$, we have

$$s_p = (1 - \prod_{i=0}^{n-1} (1 - \frac{1}{p^{m-i}}))$$

and thus

$$\rho(R) = \rho(P^{-1}(\{\emptyset\})) = \prod_{i=0}^{n-1} \frac{1}{\zeta(m-i)}.$$

Using Lemma 12.1.6, we are left with proving that the additional assumptions on the system $(U_\nu)_{\nu \in M_{\mathbb{Q}}}$ of Theorem 12.1.8 are satisfied. For this let us choose again $\alpha = 1$. Recall, that

$$\ell_{A,H} =\mid \{p \in \mathcal{P} \mid p > H, A \in [-H, H[_I^d \cap U_p\} \mid .$$

Denote by f_i the function associating to $A \in \mathbb{Z}^{n \times m}$ some fixed n-minor. Then, we have for all $A \in [-H, H[^{nm}$ the inequality $f_i(A) \leq (2H)^n$ holds for all i. We exclude that $f_i(A)$ vanishes for all i, since then we land in I. Further, observe that

$$H^{\ell_{A,H}} \leq \prod_{\substack{p \in \mathcal{P} \\ p > H,\ A \in U_p \cap [-H, H[_I^{nm}}} p,$$

since all $p > H$ and there are $\ell_{A,H}$ many. In addition, we have that

$$\prod_{\substack{p \in \mathcal{P} \\ p > H,\ A \in U_p \cap [-H, H[_I^{nm}}} p \leq \gcd((f_i(A))_i) \leq (2H)^n.$$

Hence we get that $H^{\ell_{A,H}} \leq (2H)^n$, which implies that $\ell_{A,H} \leq n$ for H large enough. With this, Condition (12.1.1) is satisfied.

In order to verify Condition (12.1.2), we want to show that there exists a sequence $(v_p)_{p \in \mathcal{P}}$, such that for all $p < H$ we have that $\mid U_p \cap [-H, H[_I^{nm} \mid \leq v_p (2H)^{nm}$. The set of non-full rank matrices over $\mathbb{F}_p$ has size

$$p^{nm} - \prod_{i=0}^{n-1} (p^m - p^i) \leq 2^n p^{m(n-1)+n-1} = 2^n p^{(m+1)(n-1)}.$$

We can fix one non-full rank $n \times m$ matrix over $\mathbb{F}_p$, for which we have less or equal than $2^n p^{(m+1)(n-1)}$ choices. For this fixed matrix there are less than or equal $(\lceil \frac{2H}{p} \rceil)^{nm}$ lifts to $\mathbb{Z}^{n\times m} \cap [-H, H[^{nm}$. Hence, we have for $p < H$

$$\begin{aligned} \mid U_p \cap [-H, H[_I^{nm} \mid &\leq 2^n \left(\left\lceil \frac{2H}{p} \right\rceil \right)^{nm} p^{(m+1)(n-1)} \\ &\leq 2^n \left(\frac{2H}{p} + 1 \right)^{nm} p^{(m+1)(n-1)} \\ &\leq 2^n (3H)^{nm} p^{nm-m+n-1-nm} \\ &\leq 6^n H^{nm} \frac{1}{p^2}. \end{aligned}$$

Thus, $(v_p)_{p\in\mathcal{P}}$ can be chosen to be $(\frac{6^n}{p^2})_{p\in\mathcal{P}}$, which also satisfies Condition (12.1.3). Hence (12.2.17) follows and Lemma 12.1.6 implies (12.2.19).

For the variance we choose $\beta = 1/2$. Recall, that

$$r_{A,H} = \mid \{p \in \mathcal{P} \mid p > \sqrt{H}, A \in [-H, H[_I^d \cap U_p\} \mid,$$

and that for all $A \in [-H, H[^{nm}$ it holds that $f_i(A) \leq (\sqrt{2H})^{2n}$ for all i. We again exclude that $f_i(A)$ vanishes for all i. Since,

$$\sqrt{H}^{r_{A,H}} \leq \prod_{\substack{p\in\mathcal{P} \\ p>\sqrt{H},\ A\in U_p\cap[-H,H[_I^{nm}}} p \leq \gcd((f_i(A))_i) \leq (\sqrt{2H})^{2n},$$

we get that $\sqrt{H}^{r_{A,H}} \leq (\sqrt{2H})^{2n}$. Thus, for H large enough we have that $r_{A,H} \leq 2n$ and Condition (12.1.1) is satisfied.

Next, we prove that Conditions (12.1.6) and (12.1.7) are verified for all $p, q \leq \sqrt{H}$.

We compute

$$\begin{aligned} &|U_p \cap U_q \cap [-H, H[_I^{nm}| \\ =&|\{A \in [-H, H[_I^{nm} \mid \text{all } n\text{-minors of } A \text{ are divisible by } pq\}| \\ \leq& \left(\left\lceil \frac{2H}{pq} \right\rceil \right)^{nm} |\{B \in (\mathbb{Z}/pq\mathbb{Z})^{n\times m} \mid \text{all } n\text{-minors of } B \text{ are } 0\}| \\ =& \left(\left\lceil \frac{2H}{pq} \right\rceil \right)^{nm} |\{C \in (\mathbb{Z}/p\mathbb{Z})^{n\times m} \mid \text{all } n\text{-minors of } C \text{ are } 0\}| \\ &\qquad \times |\{D \in (\mathbb{Z}/q\mathbb{Z})^{n\times m} \mid \text{all } n\text{-minors of } D \text{ are } 0\}| \\ =& \left(\left\lceil \frac{2H}{pq} \right\rceil \right)^{nm} \left(p^{nm} - \prod_{i=0}^{n-1} (p^m - p^i) \right) \left(q^{nm} - \prod_{j=0}^{n-1} (q^m - q^j) \right) \\ \leq& 4^n \left(\left\lceil \frac{2H}{pq} \right\rceil \right)^{nm} (pq)^{(m+1)(n-1)}. \end{aligned}$$

For $p, q < \sqrt{H}$, this implies

$$\begin{aligned}|U_p \cap U_q \cap [-H, H[_I^{nm}| \leq & 4^n \left(\frac{4H}{pq}\right)^{nm} (pq)^{nm-m+n-1} \\ \leq & 4^{nm}(2H)^{nm}(pq)^{-m+n-1} \\ \leq & (2H)^{nm} \frac{2^{nm}}{p^2} \frac{2^{nm}}{q^2}.\end{aligned}$$

Thus, we can pick $\tilde{v}_p = \frac{2^{nm}}{p^2}$ and Conditions (12.1.6) and (12.1.7) are satisfied. □

Chapter 13

Conclusion and Future Work

In this part we applied the local to global principle on four sets over the integers, namely, the coprime pairs, the coprime m-tuples, the Eisenstein polynomials and the rectangular unimodular matrices. Note that all results, except for the rectangular unimodular matrices, are known, but they have not been proved through the local to global principle. This work thus forms a compendium of applications of the local to global principle.

We have further provided the idea of how to generalize the local to global principle to the ring of algebraic integers of some number field and given the respective results. Note that the results for the coprime pairs and coprime m-tuples are known, but again not via the local to global principle.

We have left out the densities of Eisenstein polynomials over the algebraic integers, as this is ongoing work with Simran Tinani. Hence the first future work project of this part of the thesis is to compute the densities of monic and non-monic Eisenstein polynomials over $\mathcal{O}_K$.

We have defined the mean and the variance corresponding to the natural density and, using methods of analytic number theory, we computed the mean and the variance for the four sets, namely the coprime pairs, the coprime m-tuples, the Eisenstein polynomials and the rectangular unimodular matrices. Note that only the result for Eisenstein polynomials was already known.

By doing so, we observed that the mean and the variance follow a certain pattern, which is coming from the local to global principle. We thus have provided an addendum to the local to global principle, which, with a few additional conditions, allows to compute the mean and the variance directly. We have then seen the four sets again as corollaries to the addendum and have therefore verified the mean and variance computations from before.

Since the local to global principle can also be applied on algebraic integers, a further generalization of the addendum could be to compute the mean and the variance corresponding to the density over algebraic integers. Another idea is to add higher

moments to the addendum.

Of course, there are many more sets for which one can compute the density through the local to global principle, (*e.g.* shifted Eisenstein polynomials, affine Eisenstein polynomials and many more), but for time and space reasons we restricted ourselves to the selected four sets. Clearly, a broader compendium of such computations could be of interest, nevertheless, we believe that with this selection we have given the reader a broad variety of difficulties for computing densities and expected values.

Appendices

Appendix A

A.1 Code-Based Cryptography

In this section we recall the framework of the McEliece system and the Niederreiter system.

The McEliece Framework

Key Generation Let $k \leq n$ be two positive integers and choose $\mathcal{C}$ an $[n, k]$ linear code over $\mathbb{F}_q$, that con correct up to t errors and has an efficient decoding algorithm $\mathcal{D}$. Let $\mathbf{G}$ be a $k \times n$ generator matrix of $\mathcal{C}$.

Choose randomly a matrix $\mathbf{S} \in \mathrm{GL}_k(\mathbb{F}_q)$ and a $n \times n$ permutation matrix $\mathbf{P}$. Compute $\mathbf{G}' = \mathbf{SGP}$.

The public key is then given by $(\mathbf{G}', t)$ and the private key is given by $(\mathbf{G}, \mathbf{S}, \mathbf{P})$.

Encryption Choose a message $\mathbf{m} \in \mathbb{F}_q^k$ and a random error vector $\mathbf{e} \in \mathbb{F}_q^n$ of Hamming weight less than or equal to t, *i.e.*, $\mathrm{wt}_H(\mathbf{e}) \leq t$. The cipher is then given by

$$\mathbf{c} = \mathbf{mG}' + \mathbf{e}.$$

Decryption We first take away the permutation matrix, by computing

$$\mathbf{cP}^{-1} = \mathbf{mSG} + \mathbf{eP}^{-1}.$$

Note that $\mathbf{eP}^{-1}$ has still Hamming weight less than or equal to t and the code generated by $\mathbf{SG}$ is the same as the code generated by $\mathbf{G}$, thus we can use the decoding algorithm $\mathcal{D}$ on $\mathbf{cP}^{-1}$ to get $\mathbf{m}$.

Equivalently, we can decode using the generator matrix $\mathbf{G}$ and obtain $\mathbf{mS}$ and by multiplying with $\mathbf{S}^{-1}$ we recover $\mathbf{m}$.

The Niederreiter Framework

Key Generation Let $k \leq n$ be two positive integers and choose $\mathcal{C}$ an $[n, k]$ linear code over $\mathbb{F}_q$, that con correct up to t errors and has an efficient decoding algorithm $\mathcal{D}$. Let $\mathbf{H}$ be a $(n-k) \times n$ parity-check matrix of $\mathcal{C}$.

Choose randomly a matrix $\mathbf{S} \in \mathrm{GL}_{n-k}(\mathbb{F}_q)$ and a $n \times n$ permutation matrix $\mathbf{P}$. Compute $\mathbf{H}' = \mathbf{SHP}$.

The public key is then given by $(\mathbf{H}', t)$ and the private key is given by $(\mathbf{H}, \mathbf{S}, \mathbf{P})$.

Encryption Choose a message $\mathbf{e} \in \mathbb{F}_q^k$ of Hamming weight less than or equal to t, *i.e.*, $\mathrm{wt}_H(\mathbf{e}) \leq t$. The cipher is then given by

$$\mathbf{c}^\top = \mathbf{H}'\mathbf{e}^\top.$$

Decryption We first take away the invertible matrix by computing

$$\mathbf{S}^{-1}\mathbf{c}^\top = \mathbf{HPe}^\top.$$

Note that $\mathbf{Pe}^\top$ has still Hamming weight less than or equal to t and thus we can use the decoding algorithm $\mathcal{D}$ on $\mathbf{S}^{-1}\mathbf{c}^\top$ to get $\mathbf{Pe}^\top$ and thus we can recover the message $\mathbf{e}$.

A.2 Information Set Decoding over $\mathbb{F}_2$ in the Hamming Metric

For the sake of completeness, we will provide in this section also the original algorithms of Prange, Lee-Brickell and Stern over the binary. However, for the complexity analysis we will use more modern techniques instead of providing the original cost.

A.2.1 Prange

Algorithm 14 Prange's Algorithm over $\mathbb{F}_2$ in the Hamming metric

Input: $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_2^{n-k}$, $t \in \mathbb{N}$.
Output: $\mathbf{e} \in \mathbb{F}_2^n$ with $\mathbf{eH}^\top = \mathbf{s}$ and $\mathrm{wt}_H(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size k and define $J = \{1, \ldots, n\} \setminus I$.
2: Compute $\mathbf{U} \in \mathbb{F}_2^{(n-k)\times(n-k)}$, such that

$$(\mathbf{UH})_I = \mathbf{A} \quad \text{and } (\mathbf{UH})_J = \mathrm{Id}_{n-k},$$

where $\mathbf{A} \in \mathbb{F}_2^{(n-k)\times k}$.
3: Compute $\mathbf{s}' = \mathbf{sU}^\top$.
4: **if** $\mathrm{wt}_H(\mathbf{s}') = t$ **then**
5: Return $\mathbf{e}$, such that $\mathbf{e}_I = \mathbf{0}_k$ and $\mathbf{e}_J = \mathbf{s}'$.
6: Start over with Step 1 and a new selection of I.

The average cost of Prange's original algorithm [93] is given as follows.

Theorem A.2.1. *Prange's algorithm over $\mathbb{F}_2$ requires on average*

$$\binom{n-k}{t}^{-1}\binom{n}{t}(n-k)^2(n+1)$$

binary operations.

Proof. One iteration of Algorithm 14 over $\mathbb{F}_2$ only consists in bringing $\mathbf{H}$ into systematic form and to apply the same row operations on the syndrome; thus, the cost can be assumed equal to that of computing $\mathbf{U}\begin{pmatrix}\mathbf{H} & \mathbf{s}^\top\end{pmatrix}$, *i.e.*,

$$(n-k)^2(n+1)$$

binary operations.

The success probability is given by having chosen the correct weight distribution of $\mathbf{e}$. As in the q-ary case this is given by

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

□

A.2.2 Lee-Brickell

Algorithm 15 Lee-Brickell's Algorithm over $\mathbb{F}_2$ in the Hamming metric

Input: $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_2^{n-k}$, $t \in \mathbb{N}$, $v < \min\{k, v\}$.
Output: $\mathbf{e} \in \mathbb{F}_2^n$ with $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$ and $\mathrm{wt}_H(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size k and define $J = \{1, \ldots, n\} \setminus I$.
2: Compute $\mathbf{U} \in \mathbb{F}_2^{(n-k)\times(n-k)}$, such that

$$(\mathbf{U}\mathbf{H})_I = \mathbf{A} \quad \text{and } (\mathbf{U}\mathbf{H})_J = \mathrm{Id}_{n-k},$$

where $\mathbf{A} \in \mathbb{F}_2^{(n-k)\times k}$.
3: Compute $\mathbf{s}' = \mathbf{s}\mathbf{U}^\top$.
4: **for** $\mathbf{e}_I \in \mathbb{F}_2^k$ with $\mathrm{wt}_H(\mathbf{e}_I) = v$ **do**
5: **if** $\mathrm{wt}_H(\mathbf{s}' + \mathbf{e}_I\mathbf{A}^\top) = t - v$ **then**
6: Return $\mathbf{e}$ such that $\mathbf{e}_I = \mathbf{e}_I$ and $\mathbf{e}_J = \mathbf{s}' + \mathbf{e}_I A^\top$.
7: Start over with Step 1 and a new selection of I.

In the following we give the complexity analysis of Lee-Brickell's algorithm over the binary [65]. Note that we will use the speed-up technique of early abort.

Theorem A.2.2. *Lee-Brickell's algorithm over $\mathbb{F}_2$ requires on average*

$$\binom{k}{v}^{-1}\binom{n-k}{t-v}^{-1}\binom{n}{t}\left((n-k)^2(n+1)+\binom{k}{v}\min\{n-k,2(t-v+1)\}v\right)$$

binary operations.

Proof. As a first step, we bring $\mathbf{H}$ into systematic form and apply the same row operations on the syndrome. This can again be assumed to be equal to that of computing $\mathbf{U}\begin{pmatrix}\mathbf{H} & \mathbf{s}^\top\end{pmatrix}$, hence a broad estimate for the cost is

$$(n-k)^2(n+1)$$

binary operations.

In the next step, we go through all $\mathbf{e}_I \in \mathbb{F}_2^k$ of Hamming weight v, and compute $\mathbf{s}\mathbf{U}^\top + \mathbf{e}_I\mathbf{A}^\top$. This would usually require

$$\binom{k}{v}(n-k)v$$

binary operations, since $\mathbf{e}_I$ has support size v. However, the algorithm only proceeds, if the weight of $\mathbf{s}\mathbf{U}^\top + \mathbf{e}_I\mathbf{A}^\top$ is $t-v$. Thus, we can apply the concept of early abort. Since we are over $\mathbb{F}_2$, and we assume that the resulting vector is uniformly distributed, we have that one entry of the resulting vector adds Hamming weight 1 to the weight of the full vector, with probability $\frac{1}{2}$. Thus, we have to compute on average $2(t-v+1)$ many entries of the resulting vector before we can abort. Computing one entry of the vector $\mathbf{s}\mathbf{U}^\top + \mathbf{e}_I\mathbf{A}^\top$ costs v binary operations. Thus, by applying early abort, we get that this step costs on average

$$2(t-v+1)v$$

binary operations.

The success probability is given as in the q-ary case, by

$$\binom{k}{v}\binom{n-k}{t-v}\binom{n}{t}^{-1}.$$

Then, the estimated overall cost of Lee-Brickell's ISD algorithm over $\mathbb{F}_2$ is given as in the claim. □

A.2.3 Stern

In the following we present Stern's algorithm over the binary [100].

We now provide a complexity estimate of Stern's algorithm in the Hamming metric using the techniques of early abort and intermediate sums.

Algorithm 16 Stern's Algorithm over $\mathbb{F}_2$ in the Hamming metric

Input: $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_2^{n-k}$, $t \in \mathbb{N}$, $m_1 + m_2 = k, \ell < n-k$ and $v < \min\{m_1, m_2, \lfloor \frac{t}{2} \rfloor\}$.
Output: $\mathbf{e} \in \mathbb{F}_2^n$ with $\mathbf{eH}^\top = \mathbf{s}$ and $\mathrm{wt}_H(\mathbf{e}) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size k and choose a zero-window $Z \subset \{1, \ldots, n\} \setminus I$ of size ℓ, and define $J = \{1, \ldots, n\} \setminus (I \cup Z)$.
2: Partition I into X of size m_1 and Y of size $m_2 = k - m_1$.
3: Compute $\mathbf{U} \in \mathbb{F}_2^{(n-k)\times(n-k)}$, such that

$$(\mathbf{UH})_I = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix}, \ (\mathbf{UH})_Z = \begin{pmatrix} \mathrm{Id}_\ell \\ \mathbf{0}_{(n-k-\ell)\times\ell} \end{pmatrix},$$
$$\text{and } (\mathbf{UH})_J = \begin{pmatrix} \mathbf{0}_{\ell\times(n-k-\ell)} \\ \mathrm{Id}_{n-k-\ell} \end{pmatrix},$$

where $\mathbf{A} \in \mathbb{F}_2^{\ell\times k}$ and $\mathbf{B} \in \mathbb{F}_2^{(n-k-\ell)\times k}$.
4: Compute $\mathbf{sU}^\top = \begin{pmatrix} \mathbf{s}_1 & \mathbf{s}_2 \end{pmatrix}$, where $\mathbf{s}_1 \in \mathbb{F}_2^\ell$ and $\mathbf{s}_2 \in \mathbb{F}_2^{n-k-\ell}$.
5: Compute the set S

$$S = \{(\mathbf{e}_X\mathbf{A}^\top, \mathbf{e}_X) \mid \mathbf{e}_X \in \mathbb{F}_2^k(X), \ \mathrm{wt}_H(\mathbf{e}_X) = v\}.$$

6: Compute the set T

$$T = \{(\mathbf{e}_Y\mathbf{A}^\top + \mathbf{s}_1, \mathbf{e}_Y) \mid \mathbf{e}_Y \in \mathbb{F}_2^k(Y), \ \mathrm{wt}_H(\mathbf{e}_Y) = v\}.$$

7: **for** $(\mathbf{a}, \mathbf{e}_X) \in S$ **do**
8: **for** $(\mathbf{a}, \mathbf{e}_Y) \in T$ **do**
9: **if** $\mathrm{wt}_H(\mathbf{s}_2 + (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top) = t - 2v$ **then**
10: Return $\mathbf{e}$, such that $\mathbf{e}_I = \mathbf{e}_X + \mathbf{e}_Y$, $\mathbf{e}_Z = \mathbf{0}_\ell$ and $\mathbf{e}_J = \mathbf{s}_2 + (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top)$.
11: Start over with Step 1 and a new selection of I.

Theorem A.2.3. *Stern's algorithm over $\mathbb{F}_2$ requires on average*

$$\binom{m_1}{v}^{-1}\binom{m_2}{v}^{-1}\binom{n-k-\ell}{t-2v}^{-1}\binom{n}{t}\Big((n-k)^2(n+1) + \ell\left(L(m_1, v) + L(m_2, v) + \binom{m_2}{v}\right) + \frac{\binom{m_1}{v}\binom{m_2}{v}}{2^\ell}\min\{n-k-\ell, 2(t-2v+1)\}2v\Big)$$

binary operations.

Proof. Again, we assume that bringing $\mathbf{H}$ into systematic form and performing the

same row operations on the syndrome can be estimated by

$$(n-k)^2(n+1)$$

binary operations.

In the next step, we compute the set S, using the technique of intermediate sums. We want to compute $\mathbf{e}_X\mathbf{A}^\top$ for all $\mathbf{e}_X \in \mathbb{F}_2^k(X)$ of Hamming weight v. Using intermediate sums, this costs

$$\ell L(m_1, v)$$

binary operations.

Similarly, we can build set T: we want to compute $\mathbf{e}_Y\mathbf{A}^\top + \mathbf{s}_1$, for all $\mathbf{e}_Y \in \mathbb{F}_2^k(Y)$ of Hamming weight v. Using intermediate sums, this costs

$$\ell\left(L(m_2, v) + \binom{m_2}{v}\right)$$

binary operations. The $\ell L(m_2, v)$ part comes from computing $\mathbf{e}_Y\mathbf{A}^\top$, whereas the $\ell\binom{m_2}{v}$ part comes from adding to each of the vectors $\mathbf{e}_Y\mathbf{A}^\top$ the vector $\mathbf{s}_1$.

In the next step, we want to check for collisions between S and T. Since, S consists of all $\mathbf{e}_X \in \mathbb{F}_2^k(X)$ of Hamming weight v, S is of size $\binom{m_1}{v}$ and similarly T is of size $\binom{m_2}{v}$. The resulting vectors $\mathbf{e}_X\mathbf{A}^\top$, respectively, $\mathbf{e}_Y\mathbf{A}^\top + \mathbf{s}_1$ live in $\mathbb{F}_2^\ell$, and we assume that they are uniformly distributed. Hence, we have to check on average

$$\frac{\binom{m_1}{v}\binom{m_2}{v}}{2^\ell}$$

many collisions. For each collision we have to compute $\mathbf{s}_2 + (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$, which would usually require

$$(n-k-\ell)2v$$

binary operations. However, the algorithm only proceeds, if the weight of $\mathbf{s}_2 + (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$ is $t-2v$, hence we can use the concept of early abort. Since, we are over $\mathbb{F}_2$, and we assume that the resulting vector is uniformly distributed, we have that one entry of the resulting vector adds Hamming weight 1 to the weight of the full vector, with probability $\frac{1}{2}$. Hence, we have to compute on average $2(t-2v+1)$ many entries of the resulting vector before we can abort. Computing one entry of the vector $\mathbf{s}_2 + (\mathbf{e}_X + \mathbf{e}_Y)\mathbf{B}^\top$ costs $2v$ binary operations. Thus, by applying early abort, we get that this step costs on average

$$2(t-2v+1)2v$$

binary operations.

The success probability is given by having chosen the correct weight distribution of $\mathbf{e}$; in this case, we require that v errors happen in the set X, v errors happen in the set Y, and the remaining $t-2v$ outside the information set and the zero-window, hence such a probability is given by

$$\binom{m_1}{v}\binom{m_2}{v}\binom{n-k-\ell}{t-2v}\binom{n}{t}^{-1}.$$

□

A.3 Quaternary Code-Based Cryptography

In this section we state a quaternary version of the McEliece and the Niederreiter cryptosystem, as it was presented in [55]. For the key generation one chooses a quaternary code $\mathcal{C}$ of length n and type $h = 4^{k_1}2^{k_2}$, which has an efficient decoding algorithm and is able to correct up to t errors.

Quaternary McEliece

Key Generation Let G be a $(k_1 + k_2) \times n$ generator matrix of $\mathcal{C}$ and choose an $n \times n$ permutation matrix $\mathbf{P}$. On this matrix we impose no further conditions, since the change of columns does not affect the $\mathbb{F}_2$-part of the message, whereas for the $(k_1+k_2)\times(k_1+k_2)$ invertible matrix S we need further conditions: in the classical case over finite fields, S is just a change of basis, but in the $\mathbb{Z}/4\mathbb{Z}$ case, changing the rows of the generator matrix affects the position of $\mathbb{F}_2$-part of the message. Since such a change hinders the constructor of the cryptosystem to tell where the $\mathbb{F}_2$-part of the message should be taken, we will restrict the choice of invertible matrices to the following form: let $\mathbf{S}_1$ and $\mathbf{S}_2$ be $k_1 \times k_1$, respectively $k_2 \times k_2$ invertible matrices over $\mathbb{Z}/4\mathbb{Z}$, then $\mathbf{S}$ is given by

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 \end{pmatrix}.$$

Compute $\mathbf{G}' = \mathbf{SGP}$ and publish $(k_1, k_2, \mathbf{G}', t)$.

Encryption For the encryption let $\mathbf{m} = (\mathbf{m}_1, \mathbf{m}_2)$, with $\mathbf{m}_1 \in (\mathbb{Z}/4\mathbb{Z})^{k_1}$ and $\mathbf{m}_2 \in \mathbb{F}_2^{k_2}$ be the message and choose an error vector $\mathbf{e} \in (\mathbb{Z}/4\mathbb{Z})^n$ of Lee weight $\mathrm{wt}_L(\mathbf{e}) \leq t$. The cipher is computed as

$$\mathbf{c} = \mathbf{m}\mathbf{G}' + \mathbf{e}.$$

Decryption For the decryption one computes

$$\mathbf{c}\mathbf{P}^{-1} = \mathbf{m}\mathbf{S}\mathbf{G} + \mathbf{e}\mathbf{P}^{-1}.$$

Since $\mathrm{wt}_L(\mathbf{e}\mathbf{P}^{-1}) \leq t$ and $\mathbf{SG}$ generates the same code as $\mathbf{G}$, we can use the decoding algorithm of the code to recover $\mathbf{mS}$ and hence the message $\mathbf{m}$.

Quaternary Niederreiter The quaternary version of the Niederreiter cryptosystem is done in a similar way by using the parity-check matrix $\mathbf{H}$ and by computing its syndromes for encryption. Since there is no restriction on the message space in the Niederreiter version, there will be no conditions needed on the permutation matrix and on the invertible matrix.

Key Generation Again, one chooses a quaternary code $\mathcal{C}$ of length n and type $h = 4^{k_1}2^{k_2}$, which has an efficient decoding algorithm and is able to correct up to t errors.

Let $\mathbf{H}$ be a $(n-k_1)\times n$ parity-check matrix of $\mathcal{C}$, choose an invertible $(n-k_1)\times(n-k_1)$ matrix $\mathbf{S}$, *i.e.* $\det(\mathbf{S}) \in (\mathbb{Z}/4\mathbb{Z})^\times$ and an $n \times n$ permutation matrix $\mathbf{P}$. Compute $\mathbf{H}' = \mathbf{S}^{-1}\mathbf{H}\mathbf{P}^\top$ and publish $(k_1, k_2, \mathbf{H}', t)$.

Encryption For the encryption let $\mathbf{e} \in (\mathbb{Z}/4\mathbb{Z})^n$ be the message of Lee weight $\mathrm{wt}_L(\mathbf{e}) \leq t$. The cipher is computed as

$$\mathbf{c} = \mathbf{H}'\mathbf{e}^\top.$$

Decryption For the decryption one computes

$$\mathbf{S}\mathbf{c} = \mathbf{H}\mathbf{P}^\top\mathbf{e}^\top.$$

Since $\mathrm{wt}_L(\mathbf{P}^\top\mathbf{e}^\top) \leq t$, we can use the decoding algorithm of the code to recover $\mathbf{P}^\top\mathbf{e}^\top$ and hence the message $\mathbf{e}$.

Key Size To determine the key size we need to count the number of non-prescribed entries of the public generator matrix. For this we assume that the generator matrix is published in quaternary systematic form as in (5.1.5).

This allows us to compute the size of the generator matrix in the form (5.1.5), or equivalently the size of the parity-check matrix in the form (5.1.6).

Theorem A.3.1. *The size of the public key, given by the non-prescribed parts of either the generator matrix* (5.1.5) *or the parity-check matrix* (5.1.6)*, is*

$$2(n-k_1-k_2)k_1 + (n-k_1-k_2)k_2 + k_1k_2 = k_1k_2 + (2k_1+k_2)(n-k_1-k_2)$$

bits.

Bibliography

[1] Morton Abramson. Restricted combinations and compositions. *Fibonacci Quart*, 14(5):439, 1976.

[2] Carlos Aguilar Melchor, Nicolas Aragon, Magali Bardet, Slim Bettaieb, Loic Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Adrien Hauteville, Ayoub Otmani, Olivier Ruatta, Jean-Pierre Tillich, and Gilles Zémor. ROLLO–Rank-Ouroboros, LAKE & LOCKER. second round submission to the NIST post-quantum cryptography call. *NIST PQC Round*, 2:4, 2019.

[3] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loic Bidoux, Olivier Blazy, Maxime Bros, Alain Couvreur, Jean-Christophe Deneuville, Philippe Gaborit, Adrien Hauteville, and Gilles Zémor. Rank quasi cyclic (RQC). second round submission to the NIST post-quantum cryptography call, 2019.

[4] Abdulrahman Al Jabri. A statistical decoding algorithm for general linear block codes. In *IMA International Conference on Cryptography and Coding*, pages 1–8. Springer, 2001.

[5] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, et al. Status report on the second round of the NIST post-quantum cryptography standardization process. *NIST, Tech. Rep., July*, 2020.

[6] Gianira N. Alfarano, Karan Khathuria, and Violetta Weger. On single server private information retrieval in a coding theory perspective. *arXiv preprint arXiv:2008.06417*, 2020.

[7] Nicolas Aragon, Philippe Gaborit, Adrien Hauteville, and Jean-Pierre Tillich. A new algorithm for solving the rank syndrome decoding problem. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2421–2425. IEEE, 2018.

[8] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[9] Alexei Ashikhmin and Alexander Barg. Minimal vectors in linear codes. *IEEE Transactions on Information Theory*, 44(5):2010–2017, 1998.

[10] Marco Baldi, Marco Bodrato, and Franco Chiaraluce. A new analysis of the McEliece cryptosystem based on QC-LDPC codes. In *International Conference on Security and Cryptography for Networks*, pages 246–262. Springer, 2008.

[11] Marco Baldi, Marco Bianchi, Franco Chiaraluce, Joachim Rosenthal, and Davide Schipani. A variant of the McEliece cryptosystem with increased public key security. In *Proceedings of the Seventh International Workshop on Coding and Cryptography*, pages 173–182. HAL-Inria, 2011.

[12] Marco Baldi, Marco Bianchi, Franco Chiaraluce, Joachim Rosenthal, and Davide Schipani. Method and apparatus for public-key cryptography based on error correcting codes, November 17 2015. US Patent 9,191,199.

[13] Marco Baldi, Franco Chiaraluce, Joachim Rosenthal, Paolo Santini, and Davide Schipani. Security of generalised Reed–Solomon code-based cryptosystems. *IET Information Security*, 13(4):404–410, 2019.

[14] Marco Baldi, Massimo Battaglioni, Franco Chiaraluce, Anna-Lena Horlemann-Trautmann, Edoardo Persichetti, Paolo Santini, and Violetta Weger. A new path to code-based signatures via identification schemes with restricted errors. *arXiv preprint arXiv:2008.06403*, 2020.

[15] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier Verbel. Algebraic attacks for solving the rank decoding and MinRank problems without Gröbner basis. *arXiv preprint arXiv:2002.08322*, 2020.

[16] Alexander Barg. Some new NP-complete coding problems. *Problemy Peredachi Informatsii*, 30(3):23–28, 1994.

[17] Anja Becker, Antoine Joux, Alexander May, and Alexander 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.

[18] Thierry P. Berger and Pierre Loidreau. How to mask the structure of codes for a cryptographic use. *Designs, Codes and Cryptography*, 35(1):63–79, 2005.

[19] Elwyn Berlekamp. *Algebraic coding theory*. World Scientific, 1968.

[20] Elwyn Berlekamp, Robert J. McEliece, and Henk Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Information Theory*, IT-24(3):384–386, 1978.

[21] Daniel J. Bernstein. Grover vs. McEliece. In *International Workshop on Post-Quantum Cryptography*, pages 73–80. Springer, 2010.

[22] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the McEliece cryptosystem. In *International Workshop on Post-Quantum Cryptography*, pages 31–46. Springer, 2008.

[23] Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. Post-quantum cryptography, 2009.

[24] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Wild McEliece. In *International Workshop on Selected Areas in Cryptography*, pages 143–158. Springer, 2010.

[25] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: ball-collision decoding. In *Annual Cryptology Conference*, pages 743–760. Springer, 2011.

[26] Daniel J. Bernstein, Tung Chou, Tanja Lange, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, and Wen Wang. Classic McEliece: conservative code-based cryptography. *NIST submissions*, 2017. URL https://classic.mceliece.org/nist.html.

[27] Jessalyn Bolkema, Heide Gluesing-Luerssen, Christine A. Kelley, Kristin E. Lauter, Beth Malmskog, and Joachim Rosenthal. Variations of the McEliece cryptosystem. In *Algebraic geometry for coding theory and cryptography*, pages 129–150. Springer, 2017.

[28] Martin Bright, Tim Daniel Browning, and Daniel Loughran. Failures of Weak Approximation in Families. *Compositio Mathematica*, 152(7):1435–1475, 2016.

[29] Anne Canteaut and Hervé Chabanne. *A further improvement of the work factor in an attempt at breaking McEliece's cryptosystem.* PhD thesis, INRIA, 1994.

[30] Anne Canteaut and Florent 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.

[31] Anne Canteaut and Nicolas 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.

[32] Davide Castelvecchi. IBM's quantum cloud computer goes commercial. *Nature News*, 543(7644):159, 2017.

[33] Ernesto Cesaro. Probabilité de certains faits arithméthiques. *Mathesis*, 4:150–151, 1884.

[34] Florent Chabaud. Asymptotic analysis of probabilistic algorithms for finding short codewords. In *Eurocode'92*, pages 175–183. Springer, 1993.

[35] Florent Chabaud and Jacques Stern. The cryptographic security of the syndrome decoding problem for rank distance codes. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 368–381. Springer, 1996.

[36] Lily Chen, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. *Report on post-quantum cryptography*, volume 12. US Department of Commerce, National Institute of Standards and Technology, 2016.

[37] Alain Couvreur, Philippe Gaborit, Valérie Gauthier-Umaña, Ayoub Otmani, and Jean-Pierre Tillich. Distinguisher-based attacks on public-key cryptosystems using Reed–Solomon codes. *Designs, Codes and Cryptography*, 73(2):641–666, 2014.

[38] Alain Couvreur, Ayoub Otmani, Jean-Pierre Tillich, and Valérie Gauthier-Umana. A polynomial-time attack on the BBCRS scheme. In *IACR International Workshop on Public Key Cryptography*, pages 175–193. Springer, 2015.

[39] Alain Couvreur, Ayoub Otmani, and Jean-Pierre Tillich. Polynomial time attack on wild McEliece over quadratic extensions. *IEEE Transactions on Information Theory*, 63(1):404–427, 2016.

[40] Alain Couvreur, Irene Márquez-Corbella, and Ruud Pellikaan. Cryptanalysis of McEliece cryptosystem based on algebraic geometry codes and their subcodes. *IEEE Transactions on Information Theory*, 63(8):5404–5418, 2017.

[41] Jintai Ding, Jason E. Gower, and Dieter S. Schmidt. *Multivariate public key cryptosystems*, volume 25. Springer Science & Business Media, 2006.

[42] Arturas Dubickas. Polynomials irreducible by Eisenstein's criterion. *Applicable Algebra in Engineering, Communication and Computing*, 14(2):127–132, 2003. ISSN 0938-1279. doi: 10.1007/s00200-003-0131-7. URL http://dx.doi.org/10.1007/s00200-003-0131-7.

[43] Il'ya Isaakovich Dumer. Two decoding algorithms for linear codes. *Problemy Peredachi Informatsii*, 25(1):24–32, 1989.

[44] Andrea Ferraguti and Giacomo Micheli. On the Mertens–Césaro theorem for number fields. *Bulletin of the Australian Mathematical Society*, 93(02):199–210, 2016.

[45] Matthieu Finiasz and Nicolas 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.

[46] Ernst Gabidulin. A brief survey of metrics in coding theory. *Mathematics of Distances and Applications*, 66, 2012.

[47] Philippe Gaborit, Olivier Ruatta, and Julien Schrek. On the complexity of the rank syndrome decoding problem. *IEEE Transactions on Information Theory*, 62(2):1006–1019, 2015.

[48] Georges Grekos. On various definitions of density (survey). *Tatra Mt. Math. Publ*, 31(17):17–27, 2005.

[49] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[50] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical review letters*, 79(2):325, 1997.

[51] Cheikh Thiécoumba Gueye, Jean Belo Klamti, and Shoichi Hirose. Generalization of BJMM-ISD using May-Ozerov nearest neighbor algorithm over an arbitrary finite field $\mathbb{F}_q$. In *International Conference on Codes, Cryptology, and Information Security*, pages 96–109. Springer, 2017.

[52] Randell Heyman and Igor E. Shparlinski. On the number of Eisenstein polynomials of bounded height. *Applicable Algebra in Engineering, Communication and Computing*, 24(2):149–156, 2013. ISSN 0938-1279. doi: 10.1007/s00200-013-0187-y. URL http://dx.doi.org/10.1007/s00200-013-0187-y.

[53] Shoichi 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.

[54] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. NTRU: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium*, pages 267–288. Springer, 1998.

[55] Anna-Lena Horlemann-Trautmann and Violetta Weger. Information set decoding in the Lee metric with applications to cryptography. *Advances in Mathematics of Communications*, 2019. ISSN 1930-5346. doi: 10.3934/amc.2020089. URL http://aimsciences.org//article/id/ab4b64c6-5f2f-4014-b0f2-eacc9058c7cf.

[56] Nick Howgrave-Graham and Antoine Joux. New generic algorithms for hard knapsacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 235–256. Springer, 2010.

[57] Carmelo Interlando, Karan Khathuria, Nicole Rohrer, Joachim Rosenthal, and Violetta Weger. Generalization of the ball-collision algorithm. *Journal of Algebra Combinatorics Discrete Structures and Applications*, 7:195 – 207, 2020. doi: 10.13069/jacodesmath.729477.

[58] Fedor Ivanov, Grigory Kabatiansky, Eugeny Krouk, and Nikita Rumenko. A new code-based cryptosystem. In *Code-Based Cryptography Workshop*, pages 41–49. Springer, 2020.

[59] Heeralal Janwa and Oscar Moreno. McEliece public key cryptosystems using algebraic-geometric codes. *Designs, Codes and Cryptography*, 8(3):293–307, 1996.

[60] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International Workshop on Post-Quantum Cryptography*, pages 19–34. Springer, 2011.

[61] Karan Khathuria, Joachim Rosenthal, and Violetta Weger. Encryption scheme based on expanded Reed–Solomon codes. *Advances in Mathematics of Communications*, 2019.

[62] Karan Khaturia, Joachim Rosenthal, and Violetta Weger. Weight two masking of the Reed-Solomon structure in conjunction with list decoding. In *Proceedings of 23rd International Symposium on Mathematical Theory of Networks and Systems*, pages 309—314. Hong Kong University of Science and Technology, 2018.

[63] Evgenii Avramovich Kruk. Decoding complexity bound for linear block codes. *Problemy Peredachi Informatsii*, 25(3):103–107, 1989.

[64] Grégory Landais and Jean-Pierre Tillich. An efficient attack of a McEliece cryptosystem variant based on convolutional codes. In *International Workshop on Post-Quantum Cryptography*, pages 102–117. Springer, 2013.

[65] Pil Joong Lee and Ernest 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.

[66] Yongwoo Lee, Jinkyu Cho, Young-Sik Kim, and Jong-Seon No. Cryptanalysis of the Ivanov-Kabatiansky-Krouk-Rumenko cryptosystems. *IEEE Communications Letters*, 2020.

[67] Jeffrey S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transactions on Information Theory*, 34(5): 1354–1359, 1988.

[68] Yuan Xing Li, Robert H. Deng, and Xin Mei Wang. On the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–273, 1994.

[69] Carl Löndahl and Thomas Johansson. A new version of McEliece PKC based on convolutional codes. In *International Conference on Information and Communications Security*, pages 461–470. Springer, 2012.

[70] Shilin Ma, Kevin McGown, Devon Rhodes, and Mathias Wanner. On the number of primes for which a polynomial is Eisenstein. *arXiv preprint arXiv:1901.09014*, 2019.

[71] Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 203–228. Springer, 2015.

[72] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\mathcal{O}(2^{0.054n})$. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 107–124. Springer, 2011.

[73] Gérard Maze, Joachim Rosenthal, and Urs Wagner. Natural density of rectangular unimodular integer matrices. *Linear Algebra and its Applications*, 434(5): 1319–1324, 2011.

[74] Robert J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. Technical report, DSN Progress report, Jet Propulsion Laboratory, Pasadena, 1978.

[75] Franz Mertens. Ueber einige Asymptotische Gesetze der Zahlentheorie. *Journal für die reine und angewandte Mathematik*, 77:289–338, 1874.

[76] Alexander Meurer. *A coding-theoretic approach to cryptanalysis*. PhD thesis, Ruhr University Bochum, 2012.

[77] Giacomo Micheli. *Densities over global fields, arithmetic of subfield preserving maps and applications to cryptography*. PhD thesis, University of Zurich, 2015.

[78] Giacomo Micheli and Violetta Weger. On rectangular unimodular matrices over the algebraic integers. *SIAM Journal on Discrete Mathematics*, 33(1):425–437, 2019.

[79] Giacomo Micheli, Severin Schraven, and Violetta Weger. Local to global principle for expected values. *arXiv preprint arXiv:2008.06235*, 2020.

[80] Victor S. Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[81] Lorenz Minder and Amin Shokrollahi. Cryptanalysis of the Sidelnikov cryptosystem. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 347–360. Springer, 2007.

[82] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo SLM Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In *2013 IEEE international symposium on information theory*, pages 2069–2073. IEEE, 2013.

[83] Robert Niebuhr, Edoardo Persichetti, Pierre-Louis Cayrel, Stanislav Bulygin, and Johannes Buchmann. On lower bounds for information set decoding over $\mathbb{F}_q$ and on the effect of partial knowledge. *International journal of information and Coding Theory*, 4(1):47–78, 2017.

[84] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory 15*, 1(6):159–166, 1986.

[85] James E. Nymann. On the probability that k positive integers are relatively prime. *Journal of number theory*, 4(5):469–473, 1972.

[86] Ayoub Otmani, Jean-Pierre Tillich, and Léonard Dallot. Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes. *Mathematics in Computer Science*, 3(2):129–140, 2010.

[87] Alexei V. Ourivski and Thomas Johansson. New technique for decoding codes in the rank metric and its cryptography applications. *Problems of Information Transmission*, 38(3):237–246, 2002.

[88] Raphael Overbeck and Nicolas Sendrier. Code-based cryptography. In *Post-quantum cryptography*, pages 95–145. Springer, 2009.

[89] Chris Peikert. A decade of lattice cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.

[90] Christiane Peters. Information-set decoding for linear codes over $\mathbb{F}_q$. In *International Workshop on Post-Quantum Cryptography*, pages 81–94. Springer, 2010.

[91] Bjorn Poonen and Michael Stoll. The Cassels-Tate pairing on polarized Abelian varieties. *Annals of Mathematics*, 150(3):1109–1149, 1999.

[92] Bjorn Poonen and Michael Stoll. A local-global principle for densities. In Scott D. Ahlgren, George E. Andrews, and K. Ono, editors, *Topics in Number Theory*, volume 467 of *Mathematics and Its Applications*, pages 241–244. Springer US, 1999. ISBN 978-1-4613-7988-1.

[93] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, pages 5–9, 1962.

[94] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. Cryptographic communications system and method, September 20 1983. US Patent 4,405,829.

[95] Ron M. Roth. Introduction to coding theory. *IET Communications*, 47, 2006.

[96] Keisuke Shiromoto. Singleton bounds for codes over finite rings. *Journal of Algebraic Combinatorics*, 12(1):95–99, 2000.

[97] Peter W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[98] Vladimir M. Sidelnikov. A public-key cryptosystem based on binary Reed-Muller codes. *Discrete Mathematics and Applications*, 4(3):191–208, 1994.

[99] Vladimir M. Sidelnikov and Sergey O. Shestakov. On insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 2(4):439–444, 1992.

[100] Jacques Stern. A method for finding codewords of small weight. In *International Colloquium on Coding Theory and Applications*, pages 106–113. Springer, 1988.

[101] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 8.4)*, 2018. `https://www.sagemath.org`.

[102] Ekedahl Torsten. An infinite version of the chinese remainder theorem. *Commentarii mathematici Universitatis Sancti Pauli= Rikkyo Daigaku sugaku zasshi*, 40(1):53–59, 1991.

[103] Johan van Tilburg. On the McEliece public-key cryptosystem. In *Conference on the Theory and Application of Cryptography*, pages 119–131. Springer, 1988.

[104] Violetta Weger, Massimo Battaglioni, Paolo Santini, Franco Chiaraluce, Marco Baldi, and Edoardo Persichetti. Information set decoding of Lee-metric codes over finite rings. *arXiv preprint arXiv:2001.08425*, 2020.

[105] Violetta Weger, Massimo Battaglioni, Paolo Santini, Anna-Lena Horlemann-Trautmann, and Edoardo Persichetti. On the hardness of the Lee syndrome decoding problem. *arXiv preprint arXiv:2002.12785*, 2020.

[106] Christian Wieschebrink. Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. In *International Workshop on Post-Quantum Cryptography*, pages 61–72. Springer, 2010.