A Hybrid Method for Lattice Basis Reduction and Applications

A HYBRID METHOD FOR LATTICE BASIS REDUCTION AND APPLICATIONS

BY

ZHAOFEI TIAN, M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTING AND SOFTWARE

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

PHD OF COMPUTER SCIENCE

PhD of Computer Science (2017) McMaster University

(Computing and Software) Hamilton, Ontario, Canada

TITLE:	A Hybrid Method for Lattice Basis Reduction and Applications
AUTHOR:	Zhaofei Tian M.Sc., (Computing and Software) McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Sanzheng Qiao
NUMBER OF PAGES:	x, 84

To my family and my parents

Abstract

Lattice reduction aided techniques have been successfully applied to a wide range of applications. Efficient and robust lattice basis reduction algorithms are valuable. In this thesis, we present an $O(n^4 \log B)$ hybrid Jacobi method for lattice basis reduction, where n is the dimension of the lattice and B is the maximum length of the input lattice basis vectors. Building upon a generic Jacobi method for lattice basis reduction, we integrate the size reduction into the algorithm to improve its performance. To ensure the convergence and the efficiency of the algorithm, we introduce a parameter to the Lagrange reduction. To improve the quality of the computed bases, we impose a condition on the size reduction, delay the structure restoration, and include a postprocessing in the hybrid method.

Our experiments on random matrices show that the proposed algorithm produces better reduced bases than the well-known LLL algorithm and BKZ 2.0 algorithm, measured by both the orthogonality defect and the condition number of the basis matrix. Moreover, our hybrid method consistently runs faster than the LLL algorithm, although

they have the same theoretical complexity. We have also investigated two potential applications of the hybrid method. The application simulations show that the hybrid method can improve the stability of the communication channels for Multi-Input Multi-Output systems, and can partially discover the plain text when attacking the GGH cryptosystem.

Acknowledgements

I extend my sincere gratitude and appreciation to many people who made this PhD's thesis possible.

First and foremost, it is an honor for me to offer the sincerest thank to my supervisor, Dr. Sanzheng Qiao, for kindly providing guidance with his patience and knowledge throughout the development of my study and preparation of this thesis. One simply could not wish for a better or friendlier supervisor.

I would like to show my grateful appreciation to Dr. Antoine Deza and Dr. Michael Soltys for their agreement to be committee members and for their careful reading of, and helpful suggestions on, my thesis.

Last but not the least, I am indebted to my family and also all whose direct and indirect support helped and encouraged me completing my thesis in time.

Contents

List of Figures

Chapter 1

Introduction

Lattice technique plays an increasingly important role in a wide range of applications in mathematics, engineering and computer science. It has been successfully applied to numerous fields including cryptography, signal processing, wireless communication, materials science, solid-state physics, integer linear programming and number theory [21, 44, 55, 67, 90, 103]. In cryptography, lattice technique is deeply involved in designing lattice-based cryptography systems, such as GGH and NTRU, and attacking many other public-key cryptography systems, for example, RSA and knapsack [13, 27, 41]. In signal processing and wireless communication, lattice technique can be used to optimize the communication channels for a variety of applications such as global positioning systems, frequency estimation, multi-input multi-output (MIMO) systems, and many data decoding systems [42, 95, 105, 106].

A lattice consists of periodic arrangement of discrete points. Lattice reduction is to find improved representations of a given lattice, depending on the notions of reduction. Various notions of lattice reduction have been introduced, such as the Lagrange reduction, the Minkowski reduction, the Hermite reduction, the Hermite-Korkine-Zolotareff (HKZ) reduction, the Blockwise Korkine-Zolotareff (BKZ) reduction, the Gaussian reduction, the Seysen reduction, and the Lenstra-Lenstra-Lovász (LLL) reduction [49, 52, 68, 85].

A major lattice-related problem is to find a shortest non-zero vector in a lattice. The problem is known to be NP-complete in high dimensions [3, 22]. The Minkowski reduction or the HKZ reduction requires the search for shortest lattice vectors. Thus, the

algorithms for the Minkowski reduction or the HKZ reduction are non-polynomial, for example, the Kannan's algorithm, the Helfrich's algorithm [22, 43, 48, 49, 108], and the algorithm introduced by W. Zhang and S. Qiao [108].

There are also polynomial time algorithms for lattice reduction, such as algorithms for the Seysen reduction [53, 83]. In 1982, A. Lenstra, W. Lenstra, and L. Lovász presented a polynomial time lattice reduction algorithm of complexity $O(n^4 \log B)$, where n is the dimension of the lattice and B is the maximum Euclidean length of the basis vectors to be reduced [54]. This algorithm has been widely used and known as the LLL algorithm, because it is very efficient and produces good results in practice [75, 100].

There are improvements on the LLL algorithm. For example, the deep insertion LLL algorithm [71, 75] and the Blockwise Korkine-Zolotarev reduction algorithm presented by C. Schnorr and M. Euchner [82]. The deep insertion LLL, whose complexity is unknown, is a floating-point variant of the original LLL algorithm. The BKZ reduction algorithm can be viewed as another variant of the LLL algorithm, implemented in NTL using floating-point operations [86], and further improved to BKZ 2.0 by Y. Chen and P. Nguyen in 2011 [20]. It has been proven that the BKZ admits a polynomial time complexity bound with respect to the call times of the enumeration subroutine for a fixed block size [40]. Since the enumeration subroutine is exponential to the size of the block, there is no polynomial time implementation of the BKZ reduction algorithm in high dimension lattices. Despite that the complexities of the BKZ reduction algorithms are non-polynomial, BKZ 2.0 is a popular technique in many fields because it finds relatively

high quality bases with flexible block sizes in relatively short time in practice [28, 40, 70].

The complexities of some well-known lattice reduction algorithms remain unknown. For example, the complexity of the algorithms for Gaussian reduction is unknown in high dimensions [72, 84]. The complexity of the LLL algorithm is also unclear when the Lovász condition parameter δ equals 1 [54, 75, 100].

The above notions are introduced to describe the properties of lattice reduction in terms of the lengths of lattice basis vectors. In 2012, S. Qiao proposed a generic Jacobi method [78] for lattice basis reduction based on improving the orthogonality between the basis vectors. Despite its unknown complexity, empirically it runs faster than the LLL algorithm. Then in 2012, an $O(n^4 \log B)$ quasi-Jacobi method was proposed [94], whose complexity is the same as that of the original LLL algorithm. However, the condition numbers of the basis matrices, which highly influence the performance of the applications in signal processing and communication [6, 45, 102], produced by the Jacobi method and the quasi-Jacobi method are not as small as those produced by the LLL algorithm. In 2013, an enhanced Jacobi method is presented [95]. It computes better reduced basis than the LLL algorithm measured by either orthogonality defect or condition number. However, its convergence is unproven. By integrating the size reduction into the Jacobi method, a polynomial time enhanced Jacobi method [96] is introduced. Whereas, its complexity is $O(n^5 \log B)$, worse than that of the original LLL algorithm.

In this thesis, we present an $O(n^4 \log B)$ hybrid method for lattice basis reduction, which improves both the output quality and the complexity of the enhanced Jacobi

method. To ensure fast convergence, we modify the condition for the Lagrange reduction embedded in the generic Jacobi method, and push shorter vectors to the front. To improve the conditioning the computed lattice basis matrices, we integrate the size reduction into the algorithm. To lower the complexity, we delay the restoration of the upper triangular structure. To further enhance our algorithm, especially the quality of the computed basis matrix, we also include a postprocessing.

Our experiments on random matrices show that the hybrid method produces better reduced bases than the well-known LLL algorithm and the BKZ 2.0 algorithm, measured by both the orthogonality defect and the condition number of the computed basis matrix. Moreover, our algorithm requires less cpu time than the LLL algorithm, although they have the same complexity. Our simulations of the GGH cryptosystem show that the hybrid method discovered more information from the plain text than the LLL algorithm and the BKZ 2.0 algorithm when attacking the GGH cryptosystem. In the simulations of MIMO systems, the communication channels improved by the new hybrid method resulted lower bit error rate than the channels improved by the LLL algorithm and the BKZ 2.0 algorithm.

1.1 Thesis Outline and Notations

The rest of the thesis is organized as follows. In Chapter 2, we briefly introduce the background knowledge of the lattice and basis. We also review some lattice-based algorithms, including the Gaussian algorithm, the Lagrange algorithm and the generic Jacobi method.

The three algorithms introduced in Chapter 2 can be regarded as the basis for the conditional Jacobi method presented in Chapter 3. We also show the complexity of the conditional Jacobi method in the chapter.

In Chapter 4, after introducing a commonly used lattice reduction technique called the size reduction, we propose the main algorithm of this thesis, a hybrid Jacobi method for lattice basis reduction and perform its complexity analysis.

Chapter 5 shows some applications of the proposed algorithm. We show that the hybrid Jacobi method can benefit applications in cryptography and signal processing. We also demonstrate the experimental results of the comparison among the hybrid method, the widely-used LLL algorithm and the BKZ 2.0 algorithm.

The thesis concludes in Chapter 6, where we propose potential improvements and future work.

Notations: We choose column-version representation for matrices and vectors. Matrices and vectors are denoted respectively by uppercase and lowercase boldface letters. For example, a matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ represents $[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$. The determinant and transpose of a square matrix $\mathbf{A}$ are denoted by $\det(\mathbf{A})$ and $\mathbf{A}^T$, respectively. We use $\mathbf{A}(a:b, c:d)$

to denote a submatrix of $\mathbf{A}$ with elements from rows a to b and from columns c to d of $\mathbf{A}$. The symbol ":" in subscript denotes a complete row or column of a matrix, for example, $\mathbf{A}_{1,:}$ is the first row of a matrix $\mathbf{A}$. The identity matrix of order n is denoted by $\mathbf{I}_n$. The length of a vector $\mathbf{v}$ is measured by the Euclidean norm, $\|\mathbf{v}\|_2$ or simply $\|\mathbf{v}\|$. For the sake of simplicity, when we say that a vector is shorter than another vector, we mean the Euclidean length of a vector is less than that of another vector, when there is no confusion.

1.2 Thesis Contributions

This thesis presents a novel polynomial time algorithm for lattice basis reduction, the hybrid Jacobi method. The original LLL algorithm proposed in 1982 is the first polynomial time lattice reduction algorithm. Other polynomial time lattice reduction algorithms have been introduced since 1982. To our best knowledge, most of the polynomial time lattice reduction algorithms are the variants of the LLL algorithm. They improve the original LLL algorithm either on efficiency or output quality. The LLL algorithm and its variants are designed to shorten the lengths of basis vectors. In contrast, the hybrid method in this thesis focuses on improving the orthogonality of the basis vectors. Thus, the hybrid method proposed in this thesis is essentially different from the LLL algorithm and its variants.

Our hybrid Jacobi method has the same computational complexity, $O(n^4 \log B)$, as the LLL algorithm, where n is the dimension of the lattice and B is the Euclidean length

of the longest basis vector. Our experimental results show that the hybrid method produces better reduced bases than the LLL algorithm and the BKZ 2.0 algorithm, measured by both the orthogonality defect and the condition number of the basis matrices. Despite the same theoretical complexity, experimentally, the hybrid method is approximately twice as fast as the LLL algorithm.

The thesis also illustrates two potential applications in signal processing and cryptography. In the signal processing simulations, the hybrid method can improve the bit error rate for MIMO systems. In our 8×8 MIMO system, the communication channel improved by the hybrid method results lower bit error rate than the LLL algorithm in both Zero-Forcing (ZF) and Minimum Mean Squared Error (MMSE) estimations. For lattice-based cryptography, when attacking a GGH cryptosystem, the hybrid method can discover more information from the plain text than the widely used LLL algorithm and the BKZ 2.0 algorithm.

Chapter 2

Preliminaries

In this chapter, we first give some background knowledge of lattice and bases. Then, we review three lattice basis reduction algorithms: Gaussian algorithm, Lagrange algorithm and a generic Jacobi algorithm.

2.1 Basic Concepts of Lattice Theory

A lattice is a set of periodically distributed discrete points in Euclidean space, defined as follows.

Definition 2.1.1 (Lattice and Basis [27, 70])**.** Let $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$ $(m \geq n)$ be of full-column rank. The *lattice* L generated by $\mathbf{A}$ is the infinite set of linear combinations of column vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ with integer coefficients, denoted by

$$L(\mathbf{A}) = \{ \mathbf{A}\mathbf{z} \mid \mathbf{z} \in \mathbb{Z}^n \},$$

where $\mathbb{Z}^n$ is the set of all integer n-vectors. The vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ form a *basis* for lattice L. We call integers n and m the *dimension* and *rank* of $L(\mathbf{A})$, respectively. When $n = m$, $L(\mathbf{A})$ is called a *full rank* lattice.

We call matrix $\mathbf{A}$ a generator matrix or a basis matrix for the lattice $L(\mathbf{A})$. There are infinitely many bases for a lattice of dimension at least two [44]. Those bases share the property that the volumes of the parallelepipeds they generate are equal [44, 59]. We call the volume of those parallelepipeds the *determinant* of the lattice L, denoted by $\det(L)$,

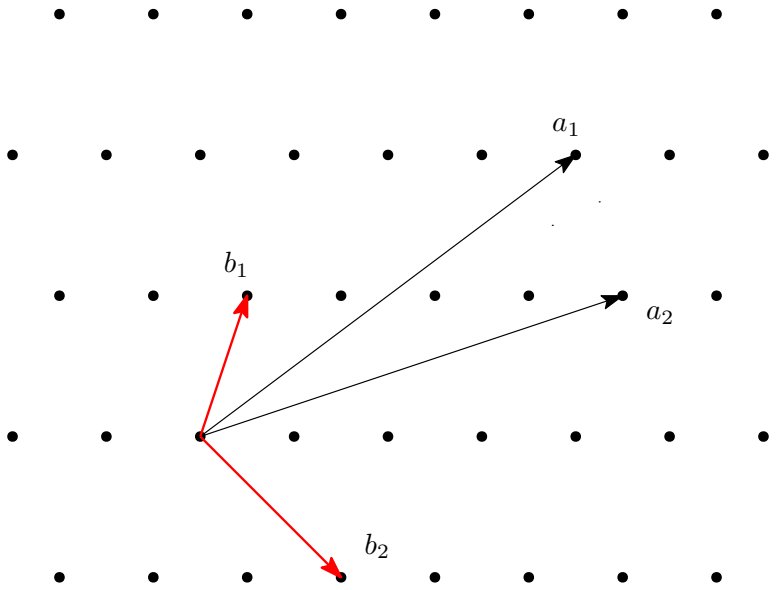


Figure 2.1: Two bases $\{\mathbf{a}_1, \mathbf{a}_2\}$ and $\{\mathbf{b}_1, \mathbf{b}_2\}$ for a same lattice.

which equals $\sqrt{\det(\mathbf{A}^\mathbf{T}\mathbf{A})}$ in terms of the basis matrix. Figure 2.1 shows two bases $\{\mathbf{a}_1, \mathbf{a}_2\}$ and $\{\mathbf{b}_1, \mathbf{b}_2\}$ for a two dimensional lattice L.

If $\mathbf{A}$ and $\mathbf{B}$ are two basis matrices for a same lattice L, then they are related by $\mathbf{B} = \mathbf{AZ}$, for some integer matrix $\mathbf{Z}$, called unimodular matrix, whose determinant $\det(\mathbf{Z})$ equals ± 1. Consequently, $\det(\mathbf{A}^\mathbf{T}\mathbf{A}) = \det(\mathbf{B}^\mathbf{T}\mathbf{B})$. Indeed, since $\mathbf{B}$ is also a basis matrix for L, each column of $\mathbf{A}$ can be represented as an integer linear combination of the columns of $\mathbf{B}$. Hence, there exists another integer coefficient matrix $\bar{\mathbf{Z}}$ such that $\mathbf{A} = \mathbf{B}\bar{\mathbf{Z}}$. Obviously, $\bar{\mathbf{Z}}$ is the inverse of $\mathbf{Z}$. Since $\det(\mathbf{A}^\mathbf{T}\mathbf{A}) = \det(\mathbf{B}^\mathbf{T}\mathbf{B})$, we know that $\det(\mathbf{Z}) = \pm 1$. To compute a new basis for a lattice, we only need to find a unimodular matrix and multiply it by the given basis matrix.

Various metrics are used to measure the quality of a lattice basis, for example, the orthogonality defect [44], the Hermite factor and the condition number [37]. The Hermite factor, popularized by Gama and Nguyen [29], assesses the length of the shortest vector in a basis. The orthogonality defect measures the geometric mean of the lengths of the basis vectors, defined as follows.

Definition 2.1.2 (Orthogonality defect [44])**.** Given a basis matrix $\mathbf{A}$ for a lattice L, the orthogonality defect $\delta(\mathbf{A})$ of $\mathbf{A}$ is defined by

$$\delta^n(\mathbf{A}) = \frac{\Pi_i \|\mathbf{a}_i\|_2}{\sqrt{\det(\mathbf{A}^T\mathbf{A})}}. \tag{2.1.1}$$

The orthogonality defect is also called the *Hadamard Ratio* [44, 47, 64]. From the Hadamard's Inequality [62], $\delta(\mathbf{A}) \geq 1$, where the equality holds if and only if the columns $\mathbf{a}_i$ are orthogonal to each other. The closer $\delta(\mathbf{A})$ is to 1, the smaller the geometric mean of the lengths of the columns is, and the columns are considered being more orthogonal to each other.

The Hermite factor is another commonly used criterion for the quality of a basis. Since it is similar to the orthogonality defect, in that they both measure the basis vector lengths, we do not include the Hermite factor as a measurement in our experiments.

Different from the orthogonality defect, the condition number describes the non-singularity of a matrix [37]. Therefore, in MIMO detection, the condition number of the channel matrix characterizes the quality and stability of the signal transformation

channel. The condition number used in the thesis is defined as follows.

Definition 2.1.3 (Condition number [37])**.** Given a matrix **A**, we define the condition number of **A** as

$$\kappa(\mathbf{A}) = \frac{\sigma_{max}}{\sigma_{min}}, \tag{2.1.2}$$

where σ_{max} and σ_{min} are respectively the maximum and minimum singular values of **A**.

Let **A** be a channel matrix of a MIMO system, then $\kappa(\mathbf{A})$ measures the impact of the channel realization on the noise influence of signal decoding schemes [6, 104]. A channel matrix with a smaller $\kappa(\mathbf{A})$ achieves better performance as the impact of noise enhancement is reduced.

For example, let

$$\mathbf{A} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

be a basis matrix and

$$\mathbf{Z} = \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix}.$$

We can verify that $|\det(\mathbf{Z})| = 1$, thus $\mathbf{Z}$ is a unimodular matrix. So

$$\mathbf{B} = \mathbf{AZ} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 0 \end{bmatrix}$$

is another basis matrix for the lattice $L(\mathbf{A})$. For the two basis matrices, we have $\delta(\mathbf{A}) \approx 2.1138$ and $\delta(\mathbf{B}) \approx 1.0670$. Based on the orthogonality defect, we can say that $\mathbf{B}$ is a better basis than $\mathbf{A}$ for the lattice. On the other hand, their condition numbers are approximately $\kappa(\mathbf{A}) \approx 12.3022$ and $\kappa(\mathbf{B}) \approx 2.1121$. Based on the condition numbers, we may say that $\mathbf{B}$ is better than $\mathbf{A}$ for the lattice. However, it should be noted that a smaller orthogonality defect does not always imply a smaller condition number. For example, considering

$$\mathbf{C} = \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{D} = \begin{bmatrix} 2 & 0 & 1 & -1 & -1 \\ 0 & 2 & 1 & -1 & -1 \\ 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & -1 & -1 \end{bmatrix},$$

we have $\delta(\mathbf{C}) \approx 1.1746$ and $\delta(\mathbf{D}) \approx 1.2282$, whereas $\kappa(\mathbf{C}) \approx 4.2656$ and $\kappa(\mathbf{D}) \approx 4.0872$.

The lattice reduction problem is to find a unimodular matrix $\mathbf{Z}$ for a given lattice basis matrix $\mathbf{A}$, such that $\mathbf{AZ}$ is *reduced*, or "better", with respect to some measurement.

2.2 Gaussian Algorithm

The Gaussian elimination [32, 37] method is used for solving systems of linear equations. Likewise, a similar Gaussian algorithm [61] acts as a fundamental technique to shorten the length of one of the two given vectors.

The Gaussian algorithm is a greedy algorithm [16]. Given two m-entry vectors $\mathbf{a}_1$ and $\mathbf{a}_2$, it finds an integer scalar q that locally minimizes $\|\mathbf{a}_1 - q\mathbf{a}_2\|_2$ and replaces vector $\mathbf{a}_1$ with $\widehat{\mathbf{a}}_1 = \mathbf{a}_1 - q\mathbf{a}_2$. Thus the length of $\mathbf{a}_1$ is reduced. Specifically, without loss of generality, we assume that $\|\mathbf{a}_1\| \geq \|\mathbf{a}_2\|$. The integer q that locally minimizes $\|\mathbf{a}_1 - q\mathbf{a}_2\|_2$ is given by $q = \lfloor \mathbf{a}_1^T\mathbf{a}_2/\|\mathbf{a}_2\|^2 \rceil$, the integer closest to $\mathbf{a}_1^T\mathbf{a}_2/\|\mathbf{a}_2\|^2$, where $\lfloor\cdot\rceil$ represents the nearest integer rounding. Intuitively, $\widehat{\mathbf{a}}_1$ is a lattice vector closest to

$$\mathbf{a}_1 - \frac{\mathbf{a}_1^T\mathbf{a}_2}{\|\mathbf{a}_2\|^2}\mathbf{a}_2,$$

which is orthogonal to $\mathbf{a}_2$. Thus

$$\widehat{\mathbf{a}}_1^T\mathbf{a}_2 = (\mathbf{a}_1 - q\mathbf{a}_2)^T\mathbf{a}_2 = \mathbf{a}_1^T\mathbf{a}_2 - q\|\mathbf{a}_2\|^2.$$

Since $q = \lfloor \mathbf{a}_1^T\mathbf{a}_2/\|\mathbf{a}_2\|^2 \rceil$, that is, $|\mathbf{a}_1^T\mathbf{a}_2/\|\mathbf{a}_2\|^2 - q| \leq 1/2$, we have $|\mathbf{a}_1^T\mathbf{a}_2 - q\|\mathbf{a}_2\|^2| \leq \|\mathbf{a}_2\|^2/2$. It then follows that

$$|\widehat{\mathbf{a}}_1^T\mathbf{a}_2| \leq \frac{1}{2}\|\mathbf{a}_2\|^2.$$

In general, we have the following procedure.

Procedure Gauss($\mathbf{a}_1, \mathbf{a}_2$)

Input : Two vectors $\mathbf{a}_1, \mathbf{a}_2$

Output: overwritten $\mathbf{a}_1$ and $\mathbf{a}_2$, so that $|\mathbf{a}_1^T \mathbf{a}_2| \leq \frac{1}{2}\max(\|\mathbf{a}_1\|_2^2, \|\mathbf{a}_2\|_2^2)$

```
if ‖a1‖2 > ‖a2‖2 then
    s = 2, l = 1 ;
else
    s = 1, l = 2 ;
q = ⌊a1^T a2 / ‖as‖2^2⌉ ;
al ← al − q as ;
```

As shown in the above Procedure `Gauss`, it first finds the shorter one $\mathbf{a}_s$ of the two given vectors, then shortens the longer vector $\mathbf{a}_l$ using an integral scalar q of the shorter vector $\mathbf{a}_s$. The output vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ satisfy $|\mathbf{a}_1^T \mathbf{a}_2| \leq \frac{1}{2}\max(\|\mathbf{a}_1\|_2^2, \|\mathbf{a}_2\|_2^2)$.

2.3 Lagrange Reduction Algorithm

Based on the key idea of the Gaussian algorithm, we have the following notion of reduced basis for a two dimensional lattice.

Definition 2.3.1 (Lagrange reduced [78, 100])**.** An $m \times 2$, $m \geq 2$, lattice basis matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2]$ is said to be *Lagrange reduced* (short as *L-reduced*), if

$$|\mathbf{a}_1^T \mathbf{a}_2| \leq \frac{1}{2}\|\mathbf{a}_1\|_2^2 \quad \text{and} \quad \|\mathbf{a}_1\|_2 \leq \|\mathbf{a}_2\|_2. \tag{2.3.1}$$

Denoting θ the angle between $\mathbf{a}_1$ and $\mathbf{a}_2$, we have $|\cos(\theta)| = |\mathbf{a}_1^T \mathbf{a}_2| / (\|\mathbf{a}_1\|_2 \|\mathbf{a}_2\|_2) \leq$

$|\mathbf{a}_1^T \mathbf{a}_2| / \|\mathbf{a}_1\|_2^2 \leq 1/2$, which implies that $\theta \in [\pi/3,\ 2\pi/3]$. Thus, we may say that $\mathbf{a}_1$ and $\mathbf{a}_2$ are close to being orthogonal to each other. In dimension two, a Lagrange reduced basis is Minkowski reduced [27, 72], which is the strongest among all notions of lattice reduction.

Algorithm 1: Lagrange Algorithm `Lagrange`$(\mathbf{a}_1, \mathbf{a}_2)$

Input : Two basis vectors $\mathbf{a}_1, \mathbf{a}_2$
Output: Overwritten $\mathbf{a}_1, \mathbf{a}_2$, so that $[\mathbf{a}_1, \mathbf{a}_2]$ is Lagrange-reduced

1 **if** $\|\mathbf{a}_1\|_2 < \|\mathbf{a}_2\|_2$ **then**
2 Swap $\mathbf{a}_1$ and $\mathbf{a}_2$;
3 **repeat**
4 $[\mathbf{a}_1\ \mathbf{a}_2] \leftarrow$ `Gauss`$(\mathbf{a}_1\ \mathbf{a}_2)$;
5 Swap $\mathbf{a}_1$ and $\mathbf{a}_2$;
6 **until** $\|\mathbf{a}_1\|_2 \leq \|\mathbf{a}_2\|_2$;

The Lagrange reduction Algorithm 1 [72, 94, 99], computes an L-reduced basis. In each Lagrange iteration, lines 4 to 5, we reduce the length of the longer one of $\mathbf{a}_1$ and $\mathbf{a}_2$ by calling procedure `Gauss`$(\mathbf{a}_1, \mathbf{a}_2)$.

Let $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ be a basis matrix for lattice $L(\mathbf{A})$, and $\mathbf{A}' = [\mathbf{a}'_1, \mathbf{a}'_2, \ldots, \mathbf{a}'_n]$ be a reduced basis matrix for the same lattice produced by a lattice reduction procedure. We define the ratio

$$\tau = \frac{\|\mathbf{a}'_1\|_2 \, \|\mathbf{a}'_2\|_2 \cdots \|\mathbf{a}'_n\|_2}{\|\mathbf{a}_1\|_2 \, \|\mathbf{a}_2\|_2 \cdots \|\mathbf{a}_n\|_2}$$

the *reduction factor* of the procedure.

Consider the Lagrange reduction algorithm. It is proved that the reduction factor of

a single Lagrange iteration is less than or equal to $1/\sqrt{3}$, except that in the first or the last iteration the reduction factor can be arbitrarily close to 1 [72, 94]. Thus, we can prove that the Lagrange algorithm terminates in polynomial time with respect to the maximum length of the input vectors [27, 70].

2.4 A Generic Jacobi Method

Nguyen introduced a greedy algorithm for lattice basis reduction which generalizes the two-dimensional Lagrange reduction algorithm to dimensions higher than two in 2009 [72]. The greedy algorithm is an iterative algorithm. It is proved that up to dimension four, the greedy algorithm can compute a Minkowski reduced basis and also terminates in polynomial time w.r.t. the big-lengths of the input basis vectors. Nevertheless, for lattices of dimension greater than four, the greedy algorithm is not optimal and exponential, since the computation is dominated by solving the closest vector problem.

A Jacobi method has a two dimensional workhorse. In the Jacobi method for the symmetric eigenvalue problem proposed by Jacobi in 1846, the workhorse is two dimensional eigenvalue decomposition [37]. In 2012, using the two dimensional Lagrange reduction, S. Qiao presented a generic Jacobi method for lattice basis reduction [78]. It repeatedly applies the Lagrange algorithm to every pair of the basis vectors for an n dimensional lattice, until every pair is L-reduced. The generic Jacobi method produces a reduced basis defined as follows.

Definition 2.4.1 (Jacobi reduced)**.** An $m \times n$ basis matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ is said to be *Jacobi reduced*, if

$$|\mathbf{a}_i^T \mathbf{a}_j| \leq \frac{1}{2}\|\mathbf{a}_i\|_2^2 \quad \text{and} \quad \|\mathbf{a}_i\|_2 \leq \|\mathbf{a}_j\|_2, \tag{2.4.1}$$

for all $1 \leq i < j \leq n$.

Algorithm 2 shows the row-cyclic version of the generic Jacobi method [78].

Algorithm 2: Generic Jacobi Method

Input : A basis matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$

Output: An overwritten reduced basis matrix $\mathbf{A}$ defined by Definition 2.4.1

```
1 while not all pairs (a_i, a_j) satisfy (2.4.1) do
2     for i = 1 to n − 1 do
3         for j = i + 1 to n do
4             [a_i, a_j] ← Lagrange(a_i, a_j) ;
```

Despite that experiments have shown that Algorithm 2 typically terminates within 10 sweeps of the `while` loop for random basis matrices of dimension less than 300, its convergence and complexity remain unknown.

Chapter 3

A Conditional Jacobi Method

To ensure the termination of the generic Jacobi method presented in Section 2.4, in this section, we present a conditional Jacobi method using the Lagrange iteration in Section 2 with a modified condition and show that its time complexity is $O(n^4 \log B)$ for an integer basis of full rank, where n is the dimension of the input lattice basis, and B is the maximum length of the input basis vectors.

3.1 A Conditional Lagrange Reduction

Consider the Lagrange reduction algorithm in Section 2, we know that the reduction factor τ can be arbitrarily close to 1 in the first or the last iteration [72, 94]. To ensure that the reduction factor of every Lagrange iteration is strictly smaller than 1, that is, the basis vector length is strictly reduced, we modify the Lagrange reduced definition (2.3.1) by introducing a condition with a parameter ω.

Definition 3.1.1 (ω-Lagrange reduced)**.** We say that a two dimensional lattice basis matrix $\mathbf{A}_{m\times 2} = [\mathbf{a}_1, \mathbf{a}_2]$ is *ω-Lagrange reduced* (short as *ωL reduced*), if

$$\left| \left\lfloor \frac{\mathbf{a}_1^T \mathbf{a}_2}{\|\mathbf{a}_s\|_2^2} \right\rceil \right| \leq 1, \tag{3.1a}$$

and

$$\omega \|\mathbf{a}_l\|_2 < \|\mathbf{a}_l - \zeta \mathbf{a}_s\|_2, \tag{3.1b}$$

where $1/\sqrt{3} \leq \omega < 1$, $\zeta = \pm 1$ denotes the sign of $\mathbf{a}_1^T \mathbf{a}_2$, $\mathbf{a}_s$ and $\mathbf{a}_l$ represent the shorter and

the longer of $\mathbf{a}_1$ and $\mathbf{a}_2$, respectively.

Since the reduction factors of the Lagrange iterations, except the first or the last iteration, are less than or equal to $1/\sqrt{3}$ [94], to ensure the strict length reduction, we only need to inspect the first and the last iterations. For these two iterations, only two values ± 1 of the integral scalar q on line 5 of Procedure `Gauss` can cause the reduction factor τ of the iteration to be greater than $1/\sqrt{3}$ [64, 94]. Note that $\mathbf{a}_l - \zeta\,\mathbf{a}_s$ in the condition (3.1b) is the same as $\mathbf{a}_l - q\mathbf{a}_s$ in line 6 of Procedure `Gauss` when $q = \pm 1$, where $\mathbf{a}_l$ is updated. The condition (3.1b) ensures that the reduction factor of every Lagrange iteration satisfying the condition (3.1b) is at most $\omega < 1$.

Let $\mathbf{A}_{m\times n}$ be a basis matrix, the gram matrix $\mathbf{G}_{n\times n} = \mathbf{A}^T\mathbf{A}$ is used for computational efficiency in [78]. Applying the QR decomposition [37], we can decompose $\mathbf{A}$ into two matrices, the orthonormal matrix $\mathbf{Q}_{m\times n}$, that is, the columns $\mathbf{q}_i$ are orthogonal to each other and $\|\mathbf{q}_i\| = 1$ (for all $1 \le i \le n$), and an upper triangular matrix $\mathbf{R}_{n\times n}$. Then, we have $\mathbf{G} = (\mathbf{R}^T\mathbf{Q}^T)\mathbf{Q}\mathbf{R} = \mathbf{R}^T\mathbf{R}$. In addition, we have $g_{ij} = \mathbf{a}_i^T\mathbf{a}_j = \mathbf{r}_i^T\mathbf{r}_j$ and $g_{jj} = \|\mathbf{a}_j\|_2^2 = \|\mathbf{r}_j\|_2^2 \ge 0$, for all $1 \le i, j \le n$. Thus, condition (2.4.1) is equivalent to

$$|g_{ij}| \le \frac{1}{2}g_{ii} \quad \text{and} \quad g_{ii} \le g_{jj}, \tag{3.1.1}$$

for all $1 \le i < j \le n$.

Procedure `LagrangeIT` performs a Lagrange iteration on the gram matrix. Given a basis matrix $\mathbf{A}$ of dimension n, the corresponding gram matrix $\mathbf{G}$, the optional upper

triangular matrix **R** in the QR-decomposition of **A**, and an initial unimodular matrix **Z**, Procedure `LagrangeIT`($\mathbf{G}, \mathbf{Z}, \mathbf{R}, i, j$) performs one iteration of the Lagrange reduction algorithm on the ith and jth basis vectors using **G**. Also, the unimodular **Z** and the optional **R** is updated accordingly if **R** is present. From the procedure, we can see that the larger of the output g_{ii} and g_{jj} equals the smaller of the input g_{ii} and g_{jj}.

Procedure LagrangeIT($\mathbf{G}, \mathbf{Z}, \mathbf{R}, i, j$)

Input : The matrices **G**, **Z** , optional **R** and a pair of indices (i, j), $i < j$
Output: The input matrices are updated, so that one Lagrange iteration is performed on the ith and jth ($i < j$) basis vectors

1 **if** $g_{ii} > g_{jj}$ **then**
2 $\quad s = j, l = i$;
3 **else**
4 $\quad s = i, l = j$;
5 $q = \left\lfloor \frac{g_{ij}}{g_{ss}} \right\rceil$;
6 Set $\mathbf{Z}_{ij} = \mathbf{I}_n$ except $z_{sl} = -q$;
7 $\mathbf{G} \leftarrow \mathbf{Z}_{ij}^T \mathbf{G} \mathbf{Z}_{ij}$;
8 $\mathbf{Z} \leftarrow \mathbf{Z}\mathbf{Z}_{ij}$;
9 **if** **R** *is present* **then**
10 $\quad \mathbf{R} \leftarrow \mathbf{R}\mathbf{Z}_{ij}$;

The procedure `LagrangeIT`($\mathbf{G}, \mathbf{Z}, \mathbf{R}, i, j$) costs $O(n)$ operations (additions or multiplications), since it operates on two rows and two columns of **G**, two columns of **Z**, and two columns of **R** if **R** is present. Note that matrix **R** is no longer upper triangular after applying $\mathbf{Z}_{ij}$. We will introduce a procedure to restore the upper triangular structure of **R** in Section 4.2.

Algorithm 3 performs the generic Jacobi method using the gram matrix **G**. It is an improvement over Algorithm 2.

Algorithm 3: Gram Jacobi method

Input : A basis matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$
Output: A unimodular matrix **Z**, so that **AZ** is reduced by Definition 2.4.1

```
1 G = A^T A, Z = I_n ;
2 while not all elements g_ij satisfy (3.1.1) do
3     for i = 1 to n − 1 do
4         for j = i + 1 to n do
5             if condition (3.1.1) is not satisfied then
6                 [G, Z] ← LagrangeIT(G, Z, i, j) ;
7                 Swap the ith and jth columns in Z ;
8                 Swap the ith and jth columns, and the ith and jth rows in G ;
```

3.2 A Conditional Jacobi Method

Based on the definition of ω-Lagrange reduction, we can define an ω-reduced basis for n-dimensional lattices.

Definition 3.2.1 (ω-Jacobi-reduced)**.** We say that an n-dimensional basis matrix $\mathbf{A}_{m\times n} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ is *ω-Jacobi-reduced*, short as *ω-reduced*, if each pair of basis vectors $(\mathbf{a}_i, \mathbf{a}_j)$ satisfies

$$\left| \left\lfloor \frac{\mathbf{a}_i^T \mathbf{a}_j}{\|\mathbf{a}_s\|_2^2} \right\rceil \right| \leq 1, \tag{3.2a}$$

and

$$\omega\|\mathbf{a}_l\|_2 \;<\; \|\mathbf{a}_l - \zeta\,\mathbf{a}_s\|_2, \tag{3.2b}$$

for all $1 \le i < j \le n$, where $1/\sqrt{3} \le \omega < 1$, $\zeta = \pm 1$ denotes the sign of $\mathbf{a}_i^T\mathbf{a}_j$, $\mathbf{a}_s$ and $\mathbf{a}_l$ are the shorter vector and the longer vector of $\mathbf{a}_i$ and $\mathbf{a}_j$, respectively.

In terms of the gram matrix $\mathbf{G}$, we have $|\zeta\,\mathbf{a}_l^T\mathbf{a}_s| = |g_{ij}|$ for subscripts i, j, l and s defined above. Then, the above inequalities (3.2a) and (3.2b) are equivalent to

$$\left|\left\lfloor \frac{g_{ij}}{g_{ss}} \right\rceil\right| \;\le\; 1, \tag{3.3a}$$

and

$$\omega^2 g_{ll} \;<\; g_{ii} + g_{jj} - 2|g_{ij}|. \tag{3.3b}$$

Introducing the above ω-reduced conditions into the generic Jacobi method, we have the following Algorithm 4, the conditional Jacobi method [96].

In Algorithm 4, line 5 checks the ω-reduced conditions on g_{ij}, ensuring that the reduction factor of every Lagrange iteration is less than or equal to ω. The situation is twofold. Firstly, for the Lagrange iterations other than the first or the last, the integral scalar q in line 4 of procedure `LagrangeIT` satisfies $|q| \ge 2$ [94]. Thus, condition (3.3a) does not hold. Under this situation, it is proved that the reduction factor $\tau \le 1/\sqrt{3}$, implying $\tau \le \omega$. Secondly, in the first or the last Lagrange iterations, $|q| = 1$ or $q = 0$ [94]. Then, the condition (3.3b) holds if and only if $\tau > \omega$. Hence, if `LagrangeIT`$(\mathbf{G}, \mathbf{Z}, i, j)$ is

Algorithm 4: Conditional Jacobi method

Input : A basis matrix $\mathbf{A}$ and ω $(1/\sqrt{3} \le \omega < 1)$
Output: A unimodular matrix $\mathbf{Z}$, so that $\mathbf{AZ}$ is ω-reduced

```
1 G = A^T A, Z = I_n ;
2 while not all elements g_ij satisfy (3.3a) and (3.3b) do
3     for i = 1 to n − 1 do
4         for j = i + 1 to n do
5             if g_ij does not satisfy (3.3a) and (3.3b) then
6                 [G, Z] ← LagrangeIT(G, Z, i, j) ;
7                 Swap the ith and jth columns in Z ;
8                 Swap the ith and jth columns, and the ith and jth rows in G ;
```

executed in this situation, we also have $\tau \le \omega$. Therefore, the reduction factor τ of each Lagrange iteration in line 6 is less than or equal to $\omega < 1$. When the algorithm terminates, the computed basis is ω-reduced defined by (3.2a) and (3.2b).

The generic Jacobi method Algorithm 2 is a special case of Algorithm 4 when $\omega = 1$.

Theorem 3.2.2. *Given an integer basis matrix $\mathbf{A}_{m\times n}$, the time complexity of the conditional Jacobi method is $O(mn^2+n^4\log B)$, where n is the dimension of the input lattice basis, and B is the maximum length of the input basis vectors. In particular, if $\mathbf{A}$ is a full rank basis matrix, then the time complexity of the conditional Jacobi method is $O(n^4\log B)$.*

Proof. Calculating $\mathbf{G}$ costs $O(mn^2)$ in line 1. Denote $D = \prod_{i=1}^{n} g_{ii} = \prod_{i=1}^{n} \|\mathbf{a}_i\|_2^2$, and let B be the maximum Euclidean length of the input basis vectors, then B^{2n} is an upper bound for D.

Let λ_1 be the first Minkowski minima of the lattice $L(\mathbf{A})$ [43, 68], which is the length of a shortest nonzero lattice vector. Then, D has a lower bound $\prod_{i=1}^{n}(\lambda_1^2) = \lambda_1^{2n}$. Each iteration on line 6 of Algorithm 4 reduces g_{ll} for some l by a factor of $\tau^2 \leq \omega^2 < 1$, while keeping g_{kk}, $k \neq l$, unchanged. Thus, each iteration reduces D by a factor of at least $\omega^2 < 1$. Therefore, after a maximum number $\log_{1/\omega^2} \frac{B^{2n}}{\lambda_1^{2n}}$ of iterations, we reduce D to the lower bound λ_1^{2n}. Since $\mathbf{A}$ is an integer matrix, λ_1 has a trivial lower bound 1. Hence, the algorithm terminates after performing the Lagrange iteration at most $O(n \log B)$ times. Since the complexity of `LagrangeIT`$(\mathbf{G}, \mathbf{Z}, i, j)$ is $O(n)$, the complexity of Algorithm 4, the conditional Jacobi method, is $O(mn^2 + n^4 \log B)$. In particular, if $m = n$, the complexity of the algorithm is $O(n^4 \log B)$, the same as the well-known LLL algorithm [54]. □

Chapter 4

A Hybrid Jacobi Method for Lattice Basis Reduction

The conditional Jacobi method introduced in Section 3 has a time complexity of $O(n^4 \log B)$ for a full rank basis matrix, where n is the dimension of the input lattice basis, and B is the maximum length of the input basis vectors. The Lagrange reduction technique used in the conditional Jacobi method is efficient at improving the orthogonality defect of a basis matrix. However, it is not effective to reduce the lengths of the basis vectors. Size reduction, on the other hand, is an effective approach to shorten the lengths of the basis vectors.

In this chapter, we present a hybrid Jacobi method, which integrates the size reduction into the conditional Jacobi method. By using the partial size reduction technique, we show that the time complexity of the hybrid Jacobi method remains $O(n^4 \log B)$ for a full rank basis matrix.

4.1 Size Reduction

Given an n-dimensional lattice L and its generator matrix $\mathbf{A}$. Let $\mathbf{R}$ be the upper triangular matrix in the QR decompositon of $\mathbf{A}$. We define the size reduction of a basis matrix as the following:

Definition 4.1.1 (Size Reduced [70])**.** Let $\mathbf{A}$ be an n dimensional basis matrix and $\mathbf{A} = \mathbf{QR}$ be its QR decomposition, then we say that $\mathbf{A}$ is *size reduced*, if the upper triangular matrix $\mathbf{R}$ satisfies

$$|r_{i,j}| \leq \frac{1}{2}|r_{i,i}|, \tag{4.1.1}$$

for all $1 \leq i < j \leq n$.

Size reduction is a widely used condition for lattice reduction algorithms, for example, the LLL algorithm [54, 75] and Schnorr's algorithm [81]. However, unlike the Lagrange reduction, size reduction cannot ensure a strict length reduction for the basis vectors, i.e., the reduction factor τ may greater than 1 after applying the size reduction.

The generic Jacobi method, which is based on the Lagrange algorithm, brings basis vectors closer to being orthogonal to each other. However, it is ineffective in reducing the lengths of the basis vectors, especially if we measure the quality of a basis matrix by the condition number of matrix. The size reduction, which is required by most lattice reductions, can effectively shorten the basis vectors and hence improve the condition number of a basis matrix [105].

For example, consider the following basis matrix [84],

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3] = \begin{bmatrix} 5 & 3 & 2 \\ 0 & 8 & -8 \\ 0 & 0 & 2 \end{bmatrix}.$$

We can verify that $\mathbf{A}$ satisfies Definition 2.4.1 (i.e., $\mathbf{A}$ is Jacobi reduced) and its condition number is approximately 10.0768. After applying size reduction on $\mathbf{A}$, $\mathbf{a}_3$ is shortened to $[0\ 0\ 2]^T$ and the condition number of the modified basis matrix is improved to approximately 4.3961.

To improve the condition number of the basis matrix produced by the conditional Jacobi method, we integrate the size reduction into Algorithm 4. Since each Lagrange iteration modifies only one column of the basis matrix, we introduce a notion of partial size reduction, which is a generalization of the size reduction described in Definition 4.1.1.

Definition 4.1.2 (Partial Size Reduction)**.** We say that a basis matrix **A** is (i, j)-*partially size reduced*, $1 \leq i < j \leq n$, if the upper triangular matrix **R** in the QR decomposition of **A** satisfies

$$|r_{k,j}| \leq \frac{1}{2}|r_{k,k}|, \tag{4.1.2}$$

for all $1 \leq k \leq i$.

From the above definition, (i, j)-partially size reduced means that the subvector $\mathbf{r}_{1:i,j}$ is size reduced. Partially size reduced is a generalization of size reduced in that if **A** is $(i, i+1)$-partially size reduced for all i, $1 \leq i < n$, then **A** is size reduced.

Let **A** be a basis matrix and **G** and **R** be the gram matrix and the upper triangular matrix in the QR decomposition of **A**, respectively. Let **Z** be an initial unimodular matrix. The following procedure `PSizeReduce` applies the (i, j)-partial size reduction on **R**. The operations on **R** are accumulated in the unimodular matrix **Z**.

We can see that there are at most $O(n)$ vector operations in Procedure `PSizeReduce`. So the time complexity of the procedure is $O(n^2)$ for an $m \times n$ basis matrix, noting that the matrices **G** and **R** are of size $n \times n$.

Procedure PSizeReduce($\mathbf{G}, \mathbf{Z}, \mathbf{R}, i, j$)

Input : $\mathbf{G}, \mathbf{Z}, \mathbf{R}$ and indices $(i,\ j)$, $i < j$
Output: Updated $\mathbf{G}, \mathbf{Z}$ and $\mathbf{R}$, so that $\mathbf{R}$ is $(i,\ j)$-partially size reduced

$\mathbf{Z}_{ij} = \mathbf{I}_n$;
for $k \leftarrow i$ **downto** 1 **do**
    if $|r_{kj}| > |r_{kk}|/2$ **then**
        $q = \lfloor r_{kj}/r_{kk} \rceil$;
        Set $\mathbf{Z}_s = \mathbf{I}_n$ except $z_{kj} = -q$;
        $\mathbf{R} \leftarrow \mathbf{R}\mathbf{Z}_s$;
        $\mathbf{Z}_{ij} \leftarrow \mathbf{Z}_{ij}\mathbf{Z}_s$;
$\mathbf{Z} \leftarrow \mathbf{Z}\mathbf{Z}_{ij}$;
$\mathbf{G} \leftarrow \mathbf{Z}_{ij}^T \mathbf{G} \mathbf{Z}_{ij}$;

4.2 Restoring the Upper Triangular Structure

Let

$$\mathbf{M} = \begin{bmatrix} r_{1,1} & r_{1,2} \\ r_{2,1} & r_{2,2} \end{bmatrix}, \quad \text{where} \quad r_{1,1}^2 + r_{2,1}^2 > 0. \tag{4.2.1}$$

To triangulate matrix $\mathbf{M}$, we construct a plane rotation matrix [37, 57, 58]

$$\mathbf{P}_2 = \begin{bmatrix} c & s \\ s & -c \end{bmatrix}, \quad \text{where} \quad c = \frac{r_{1,1}}{\sqrt{r_{1,1}^2 + r_{2,1}^2}} \text{ and } s = \frac{r_{2,1}}{\sqrt{r_{1,1}^2 + r_{2,1}^2}},$$

so that

$$\mathbf{P}_2 \mathbf{M} = \begin{bmatrix} r'_{1,1} & r'_{1,2} \\ 0 & r'_{2,2} \end{bmatrix} \tag{4.2.2}$$

to annihilate the $(2,1)$-entry. In the following hybrid Jacobi method for lattice basis reduction, when $\mathbf{R}$ is not upper triangular after a vector operation, it is necessary to restore $\mathbf{R}$ back to upper triangular structure. In the thesis, we only consider either adding a scaled column $\mathbf{r}_j$ to another column $\mathbf{r}_i$, $i < j$, or swapping two columns $\mathbf{r}_j$ and $\mathbf{r}_i$, $i < j$, meaning that $\mathbf{R}$ is upper triangular except only one column. For example, if $\mathbf{R}$ is present in procedure `LagrangeIT`, then after applying $\mathbf{Z}_{ij}$ to $\mathbf{R}$, the entries $r_{k,i}$, $k = i+1, ..., j$, of $\mathbf{R}$ become nonzero, as shown in Figure 4.1, where "x" represents a nonzero entry.

$$
\mathbf{R} = \begin{array}{c} \\ i \\ i+1 \\ \\ j-1 \\ j \\ \\ \\ \\ \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} & & i & i+1 & & j-1 & j & & \end{array} \\
\begin{pmatrix}
x & \cdots & x & x & \cdots & x & x & \cdots & x \\
 & \ddots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\
 & & x & x & \cdots & x & x & \cdots & x \\
 & & x & x & \cdots & x & x & \cdots & x \\
 & & \vdots & & \ddots & \vdots & \vdots & \cdots & \vdots \\
 & & x & & & x & x & \cdots & x \\
 & & x & & & & x & \cdots & x \\
 & & & & & & & \ddots & \vdots \\
 & & & & & & & & x
\end{pmatrix}
\end{array}
$$

Figure 4.1: The non-triangular structure of $\mathbf{R}$

Given a matrix $\mathbf{R}$ of the structure shown in Figure 4.1, the following Procedure `RestoreR` restores the upper triangular structure of $\mathbf{R}$ using the plane rotations.

In `RestoreR(`$\mathbf{R}, i, j$`)`, we eliminate the elements $r_{k,i}$, $k = i+1, ..., j$ from $k = j$ down to $k = i+1$ in the first `for` loop by the plane rotations. To find the n-dimensional plane rotation matrix $\mathbf{P}_n$ in line 2, we firstly take the following 2-by-2 matrix $\mathbf{M}_2$ as the target

matrix $\mathbf{M}$ in (4.2.1)

$$\mathbf{M}_2 = \begin{bmatrix} r_{k-1,i} & r_{k-1,j} \\ r_{k,i} & r_{k,j} \end{bmatrix}.$$

Secondly, we find a 2-dimensional plane rotation matrix $\mathbf{P}_2$ that eliminates $r_{k,i}$ in $\mathbf{M}_2$. Lastly, we construct the n-dimensional matrix $\mathbf{P}_n$ by initiating an n-dimensional identity matrix $\mathbf{I}_n$ and replacing its $(k-1,i)$, $(k-1,j)$, (k,i) and (k,j) entries with the four entries $p_{1,1}$, $p_{1,2}$, $p_{2,1}$ and $p_{2,2}$ in $\mathbf{P}_2$, respectively. After the first `for` loop, a sequence of non-zero subdiagonal elements $r_{k+1,k}$, $k = i+1,...,j-1$ are created as shown in Figure 4.2). Hence, the second `for` loop is necessary to eliminate the non-zero entries on the subdiagonal. Likewise, to eliminate the element $r_{k+1,k}$, we find the n-dimensional plane rotation matrix $\mathbf{P}_n$ in line 5 starting from the following 2-by-2 matrix

$$\mathbf{M}_2 = \begin{bmatrix} r_{k,k} & r_{k,k+1} \\ r_{k+1,k} & r_{k+1,k+1} \end{bmatrix}.$$

In line 2 of Procedure `RestoreR`, the n-dimensional plane rotation matrix $\mathbf{P}_n$ is orthogonal, i.e., $\mathbf{P}_n^T\mathbf{P}_n = 1$. Let $\mathbf{G}_n$, $\mathbf{A}_n$, $\mathbf{Q}_n$ and $\mathbf{R}_n$ be the updated Gram matrix, the basis matrix, the orthogonal matrix and the upper triangular matrix after applying the plane

$$
\mathbf{R} = \begin{array}{c} \\ \\ i \\ i+1 \\ \\ j-1 \\ j \\ \\ \\ \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} & & i & i+1 & & j-1 & j & & \end{array} \\
\begin{pmatrix}
x & \cdots & x & x & \cdots & x & x & \cdots & x \\
 & \ddots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\
 & & x & x & \cdots & x & x & \cdots & x \\
 & & & x & \cdots & x & x & \cdots & x \\
 & & & x & \ddots & \vdots & \vdots & \cdots & \vdots \\
 & & & & \ddots & x & x & \cdots & x \\
 & & & & & x & x & \cdots & x \\
 & & & & & & & \ddots & \vdots \\
 & & & & & & & & x
\end{pmatrix}
\end{array}
$$

Figure 4.2: The structure of **R** after the first `for` loop in Procedure `RestoreR`

rotation matrix $\mathbf{P}_n$, respectively, i.e., $\mathbf{R}_n = \mathbf{P}_n\mathbf{R}$ and $\mathbf{G}_n = \mathbf{A}_n^T\mathbf{A}_n$, then we have

$$
\begin{aligned}
\mathbf{G}_n &= \mathbf{A}_n^T\mathbf{A}_n \\
&= \mathbf{R}_n^T\mathbf{Q}_n^T\mathbf{Q}_n\mathbf{R}_n \\
&= \mathbf{R}^T\mathbf{P}_n^T\mathbf{Q}_n^T\mathbf{Q}_n\mathbf{P}_n\mathbf{R} \\
&= \mathbf{R}^T\mathbf{P}_n^T\mathbf{P}_n\mathbf{R} \\
&= \mathbf{R}^T\mathbf{R} \\
&= \mathbf{G}.
\end{aligned}
$$

Hence, applying the plane rotation matrix $\mathbf{P}_n$ on **R** doesn't change matrix **G**. Likewise, the matrix $\mathbf{P}_n$ in line 5 doesn't change **G**. Therefore, when we apply Procedure

`RestoreR` to triangulate a non-triangular matrix **R** of the structure in Figure 4.1, the matrices **G** and **Z** remain unchanged.

Procedure RestoreR($\mathbf{R}, i, j$)

Input : **R** and indices i, j ($i < j$)
Output: Triangulated **R**

1 **for** $k = j$ **downto** $i + 1$ **do**
2 Find a plane rotation $\mathbf{P}_n$ to eliminate $r_{k,i}$ using $r_{k-1,i}$, $r_{k-1,j}$ and $r_{k,j}$;
3 $\mathbf{R} \leftarrow \mathbf{P}_n\mathbf{R}$;
4 **for** $k = i + 1$ **to** $j - 1$ **do**
5 Find a plane rotation $\mathbf{P}_n$ to eliminate $r_{k+1,k}$ using $r_{k,k}$, $r_{k,k+1}$ and $r_{k+1,k+1}$;
6 $\mathbf{R} \leftarrow \mathbf{P}_n\mathbf{R}$;

4.3 A Hybrid Jacobi Method for Lattice Basis Reduction

Generally, the generic Jacobi method, Algorithm 2, computes basis matrices with smaller orthogonality defect than those computed by the LLL algorithm[78]. However, the complexity of the generic Jacobi method is unknown. The conditional Jacobi method, Algorithm 4, improves the generic Jacobi method with a time complexity w.r.t. the dimension of the lattice. However, the conditional Jacobi method is not as efficient in reducing the lengths of the basis vectors as the LLL algorithm [97]. An enhanced Jacobi method [95] is then proposed for MIMO decoding applications, which integrates the generic Jacobi method with the size reduction technique to improve the condition number of the computed basis matrix. Since the size reduction does not always reduce basis vector lengths,

the convergence of the enhanced Jacobi method is unknown.

In this section, we introduce a hybrid Jacobi method for lattice basis reduction short as the hybrid method. It integrates the size reduction into the conditional Jacobi method. The hybrid Jacobi method ensures the convergence. We show that the time complexity of the hybrid method is $O(n^4 \log B)$, the same as the conditional Jacobi method, where n is the dimension of the input basis, and B is the maximum length of the input basis vectors.

As shown in Algorithm 5, three techniques are used in our hybrid method. First, we introduce a condition for the partial size reduction to ensure that the basis vector lengths decrease. Specifically, after applying `PSizeReduce` on g_{ii} in line 14, if g_{ii} is not reduced, we roll back the procedure. Second, to make the length reduction of basis vectors more effective, after each inner `j`-loop, we push the shorter basis vector to the front in line 8 to 13. Third, we add a postprocessing procedure, lines 15 to 27, to the hybrid Jacobi method without changing the complexity. Unlike the main `while` loop in Algorithm 5, both the Lagrange iterations and the partial size reductions are unconditional in the postprocessing part. We can run the postprocessing multiple times to gain higher quality results. Based on our experiences, adding a postprocessing part can improve the condition number of the computed basis matrix.

For the complexity of Algorithm 5, we give the following theorem:

Theorem 4.3.1. *Let* $\mathbf{A}_{m\times n} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$ *be an integer lattice basis matrix, and* $B = \max_{1\le i\le n} \|\mathbf{a}_i\|_2$*, then, the number of arithmetic operations required by Algorithm 5, the*

Algorithm 5: Hybrid Jacobi method

Input : A basis matrix $\mathbf{A}$ and a reduction factor ω $(1/\sqrt{3} \le \omega < 1)$
Output: A unimodular matrix $\mathbf{Z}$ such that $\mathbf{AZ}$ is reduced

$\mathbf{G} = \mathbf{A}^T\mathbf{A}$, $\mathbf{Z} = \mathbf{I}_n$, get $\mathbf{R}$ from the QR decomposition of $\mathbf{A}$;
while *not all elements* g_{ij} *satisfy (3.3a) and (3.3b)* **do**
    for $i = 1$ **to** n **do**
        for $j = i+1$ **to** n **do**
            if g_{ij} *doesn't satisfy (3.3a) and (3.3b)* **then**
                $[\mathbf{G}, \mathbf{Z}, \mathbf{R}] \leftarrow$ `LagrangeIT`$(\mathbf{G}, \mathbf{Z}, \mathbf{R}, i, j)$;
        `RestoreR`$(\mathbf{R}, i, n)$;
        Find an index k $(i \le k \le n)$, s.t. $g_{kk} = \min_{l=i}^{n} g_{ll}$;
        if $k \neq i$ **then**
            Swap the ith and kth columns in $\mathbf{Z}$;
            Swap the ith and kth columns in $\mathbf{R}$;
            Swap the ith and kth columns, and the ith and kth rows in $\mathbf{G}$;
            `RestoreR`$(\mathbf{R}, i, k)$;
        Apply `PSizeReduce`$(\mathbf{G}, \mathbf{Z}, \mathbf{R}, i-1, i)$ only if g_{ii} is reduced after the appplication ;
for $i = 1$ **to** n **do**
    for $j = i+1$ **to** n **do**
        $[\mathbf{G}, \mathbf{Z}, \mathbf{R}] \leftarrow$ `LagrangeIT`$(\mathbf{G}, \mathbf{Z}, \mathbf{R}, i, j)$;
    `RestoreR`$(\mathbf{R}, i, n)$;
    Find an index k $(i \le k \le n)$, s.t. $g_{kk} = \min_{l=i}^{n} g_{ll}$;
    if $k \neq i$ **then**
        Swap the ith and kth columns in $\mathbf{Z}$;
        Swap the ith and kth columns in $\mathbf{R}$;
        Swap the ith and kth columns, and the ith and kth rows in $\mathbf{G}$;
        `RestoreR`$(\mathbf{R}, i, k)$;
    $[\mathbf{G}, \mathbf{Z}, \mathbf{R}] \leftarrow$ `PSizeReduce`$(\mathbf{G}, \mathbf{Z}, \mathbf{R}, i-1, i)$;
    for $j = i+1$ **to** n **do**
        $[\mathbf{G}, \mathbf{Z}, \mathbf{R}] \leftarrow$ `PSizeReduce`$(\mathbf{G}, \mathbf{Z}, \mathbf{R}, i, j)$;

hybrid Jacobi method for lattice basis reduction, is $O(mn^2 + n^4 \log B)$. *In particular, if* $\mathbf{A}$ *is full rank, then the complexity of Algorithm 5 is* $O(n^4 \log B)$.

Proof. Similar to the proof of Theorem 3.2.2, computing $\mathbf{G}$ and $\mathbf{R}$ takes $O(mn^2)$ arithmetic operations. The costs of `PSizeReduce`, `LagrangeIT` and `RestoreR` are $O(n^2)$. Similar as Theorem 3.2.2, the complexity of the main `while` loop is $O(n^4 \log B)$. The cost of the postprocessing is $O(n^4)$. Thus, adding a postprocessing does not compromise the complexity of our algorithm. The time complexity of the hybrid Jacobi method is $O(mn^2 + n^4 \log B)$. When $m \leq n^2$, the complexity is $O(n^4 \log B)$. In particular, if $m = n$, i.e., $\mathbf{A}$ is full rank, the complexity of the algorithm is $O(n^4 \log B)$. □

Chapter 5

Applications

In this chapter, we present our experimental results of the proposed hybrid method and compare it with two widely used algorithms, the LLL algorithm [54], a well-known polynomial time lattice reduction algorithm, and the Blockwise-Korkine-Zolotarev algorithm, BKZ 2.0. The BKZ algorithm was originally introduced by Schnorr [81] in 1987, then a practical implementation [82] was introduced in 1994. Chen and Nguyen improved the BKZ implementation to BKZ 2.0 [20], which is commonly used, because it calculates reasonably high quality basis vectors within acceptable time in practice. We adopt the BKZ 2.0 in this thesis.

We first show the experimental results of the quality of the computed bases by the algorithms. Then, we show the performance of the three algorithms by simulating two applications in signal processing and cryptography. The experiments demonstrate the potential practical applications of our method.

5.1 Quality of the Computed Bases

In this section, we show the output quality and the efficiency of the hybrid method, comparing with the other two algorithms. We measure the quality of the computed basis matrices by two metrics, the orthogonality defect $\delta(\mathbf{A})$ defined in (2.1.1) and the condition number defined in (2.1.3).

For many lattice based cryptography systems, the orthogonality defect is a popular criterion for measuring the quality of the involved lattice basis. For example, the GGH

cryptosystem, the knapsack cryptosystem and the RLWE (Ring Learning with Errors) version of the classic Diffie-Hellman cryptography. The difficulty of breaking these lattice based cryptography is equivalent to solving known hard problems on lattices, such as finding a shortest vector in a lattice of dimension at least 3 [27, 39, 63, 66]. Different from the applications in cryptography, the performance of applications in signal processing and communications are highly influenced by the condition numbers of the channel matrices, for example, the MIMO detectors [6, 45, 102]. Thus, we also choose the condition number as the second metric for comparing the quality of the computed lattice bases.

The efficiency of the algorithms is measured by the cpu time that an algorithm takes to reduce a given basis matrix.

The hybrid method, the LLL algorithm and the BKZ 2.0 are implemented in the 64 bit `MATLAB 2014b` running on a Mac OS X server with 2.8`GHz` 4 core processor. We adopt the vector-operation version [60] of the LLL algorithm to achieve high efficiency, and set the parameter ω in the LLL algorithm to 0.99 to get high quality outputs [20, 82]. The block size β in the BKZ algorithm highly affects the output quality of the computed basis. In the experiments, we adopt the two block sizes recommended by Nguyen [20]: a small block size 20 and a medium block size 40. In the cases when the dimension of the lattice is less than the block size, we use the dimension as the block size. We set the reduction factors ω, $1/\sqrt{3} \leq \omega < 1$, in the hybrid method to $1/\sqrt{3}$. From our experience, the hybrid method is insensitive to the value of ω. The postprocessing of the hybrid method is run

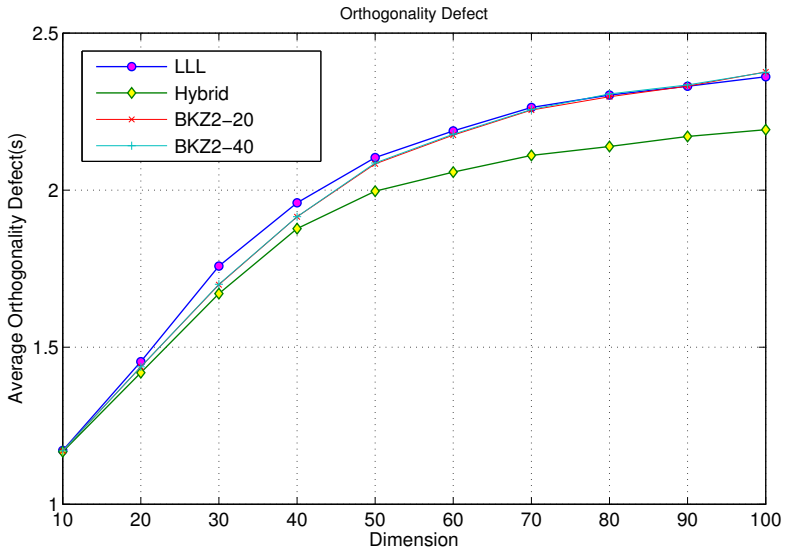
twice. We compare the basis matrices of dimensions up to 100, starting from dimension 10 with interval 10.

To obtain the average of each dimension, we generated 1000 matrices with uniformly distributed random entries between 0 and 1. The results shown in Figure 5.1 (a)-(b) and Figure 5.2 are the measurements for the chosen dimensions.

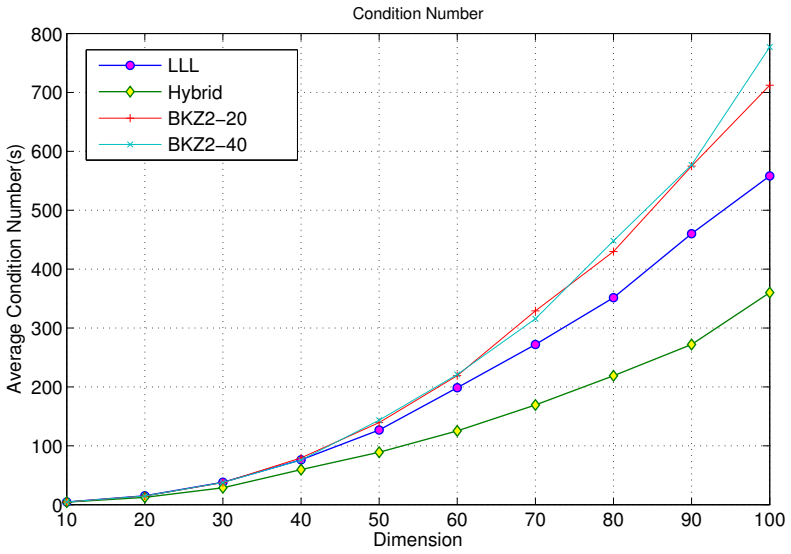
Figure 5.1 (a) and (b) illustrate the quality of the bases computed by the three algorithms, measured by the orthogonality defect and the condition number. Both measurements are the smaller, the better. The figures show that the hybrid method outperforms the LLL algorithm and the BKZ 2.0 on output quality with respect to the two measurements for random matrices. The difference becomes more significant as the dimension increases.

Figure 5.2 shows the cpu times, in the logarithm of seconds, of the three algorithms. The hybrid method is faster than both the LLL algorithm and the BKZ 2.0. As expected, both hybrid method and the LLL algorithm are much faster than the BKZ 2.0.

We also found that the hybrid method did not perform well for extremely ill-conditioned basis matrices. For extremely ill-conditioned bases, the BKZ 2.0 computed the best basis matrices among the three methods.



(a) The orthogonality defects of computed bases



(b) The condition numbers of computed bases

Figure 5.1: The quality of the bases computed by the hybrid method, the LLL algorithm and the BKZ 2.0

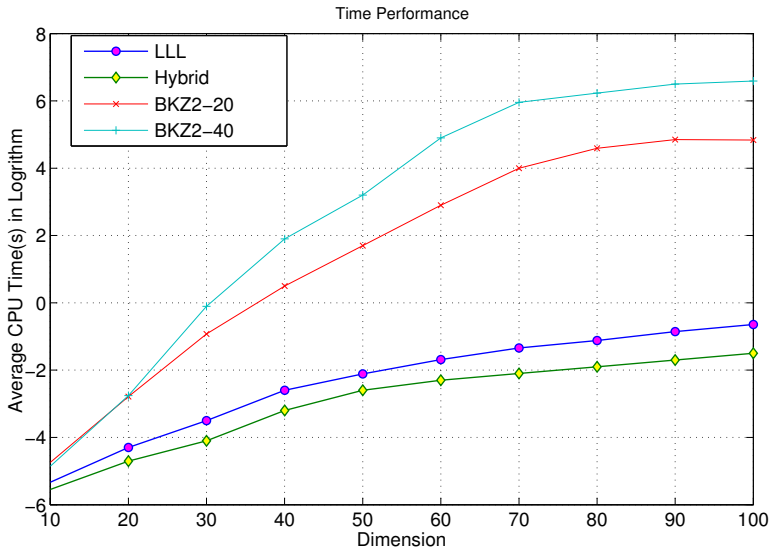


Figure 5.2: The time performance (in logarithm of seconds) of the hybrid method, the LLL algorithm and the BKZ 2.0

5.2 Multiple Input Multiple Output System

In signal processing, the performance of radio communications highly depends on the antenna system. Multiple antenna technologies are designed to achieve high transmission rate, high signal reliability and long range communications. Therefore, multiple-input multiple-output has become an essential element of wireless communication standards for wireless LANs, 3G and 4G mobile-phone networks, such as IEEE 802.11n, IEEE 802.11ac, HSPA+, WiMAX and LTE [10, 33, 92, 98].

The lattice reduction technique has been successfully introduced to numerous applications in signal processing for decades. When being applied to MIMO systems, the lattice reduction algorithms can improve the quality of the communication links and hence improve the accuracy and efficiency of signal transmission. For example, the LLL algorithm and other lattice reduction algorithms are widely adopted in MIMO systems because of their relatively low computational complexities. It has been proven that the lattice reduction algorithms such as the LLL and the BKZ 2.0 can improve the performance of MIMO systems with respect to high spectral efficiency in data transmission, high accuracy for signal detection and the maximum receive diversity over fading channels [8, 15, 31, 91, 107].

In wireless communication, there are more than one transmit antennas and more than one receive antennas in a MIMO system. The multiple antenna structure multiplies the capacity of a radio link to exploit multipath propagation [10]. Suppose that a MIMO system consists of n transmitting antennas `Tx` and m receiving antennas `Rx`, as shown

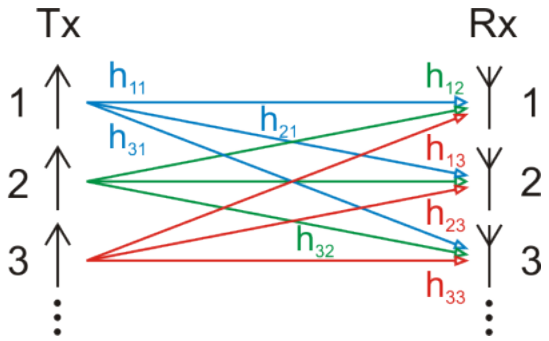


Figure 5.3: A general MIMO channel model

in Figure 5.3. The communication link of the system is the channel matrix between antennas, denoted by **H**, which consists of all mn communication paths between the antennas. The independent elements in the signal vector **x** can then be transmitted simultaneously through the transmitting antennas and the receiving antennas. Once the receiver gets a signal vector **y** from the receiving antennas Rx, it decodes the received signal vector into a transmit signal vector simultaneously. Practically, the receive signal stream also includes particular additional noise, called fading, being introduced during signal transmission [98].

We can model a MIMO system as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}, \tag{5.2.1}$$

where **H** denotes an m-by-n channel matrix, $\mathbf{z}_{m\times 1}$ is the additional noise vector, and $\mathbf{y}_{m\times 1}$ and $\mathbf{x}_{n\times 1}$ represent the receive vector and the transmit vector, respectively [36, 38,

98] .

In our MIMO simulation, we choose the instantaneous channel state information (CSI) model for the communication link, indicating that the complex channel matrix **H** is clearly known. According to the sequential Monte Carlo sampling method [1, 23, 25], we can generate a random binary stream as the transmit signals, e.g., `10010111010...` . The communication link does not accept the binary stream directly. We need to modulate these discrete signals into continuous symbols. To minimize the bit error rate (BER) during the signal transmission, we adopt the Quadrature phase-shift keying (QPSK) digital modulation scheme, also known as 4-PSK or 4-QAM [36]. QPSK modulation scheme maps every two source signal bits onto one complex symbol using Gray coding [12], shown in Figure 5.4. For example, two bits "`01`" are mapped into $-\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i$.

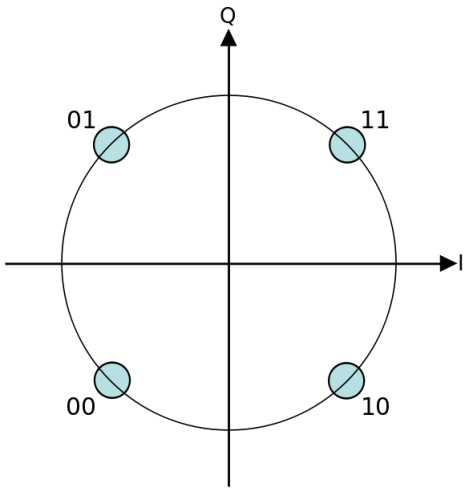


Figure 5.4: Constellation diagram for QPSK with Gray coding

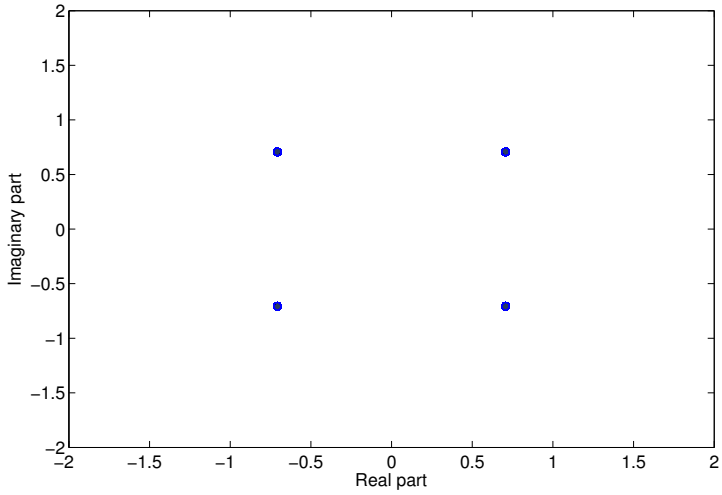
In wireless communications, fading is an unavoidable deviation of the attenuation

that affects a signal over propagation media. In our MIMO simulation, we assume that the priorities of the transmitting antennas are the same, i.e., none of the line of sight signal is much stronger than the others. Thus, we use the Rayleigh fading model for the effect of a propagation environment. The tropospheric and ionospheric signal propagation on radio signals are two typical cases of the Rayleigh fading [77, 87]. To implement Rayleigh fading in the phasor domain, we choose the noise $\mathbf{n}$ as the additive white Gaussian noise (AWGN). Hence, the resultant phasor's magnitude is a Rayleigh distributed random variable, i.e., the phase is uniformly distributed with zero mean between 0 and 2π radians [76, 79, 87]. The signal-to-noise ratio(SNR) is defined as the ratio of the power of a signal and the power of background noise:

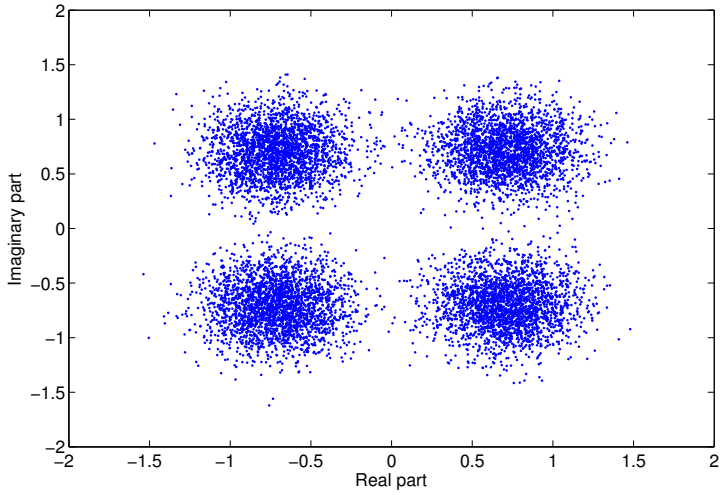
$$\mathtt{SNR} = \frac{E_{\mathbf{x}\in\mathscr{A}}(\|\mathbf{Hx}\|_2^2)}{E(\|\mathbf{n}\|_2^2)}, \tag{5.2.2}$$

where the transmitting signals $\mathbf{x}$ are assumed to be uniformly distributed in the finite set of modulation alphabet $\mathscr{A}$ [77, 108].

There are two typical detection methods in MIMO systems. The zero forcing (ZF) detection compensates the delays of receiving signals from a specific source when using as a time-domain equalizer, and it maximizes the transmitted signal capacity of the given communication channel in spatial domain. The other method is the minimum mean-square-error (MMSE) precoding, which is more commonly used since it minimizes the expected or the mean value of the square of the error [9, 80, 89].



(a) Transmit symbols mapped by the QPSK scheme



(b) Receive symbols with AWGN by the MMSE decoding at SNR of 20 dB

Figure 5.5: Example of transmit and receive symbols using QPSK scheme

If perfect complex channel gain matrix (CSI) is available at the receiver, the zero-forcing estimation of the transmitted symbol vector can be written as

$$\hat{\mathbf{x}} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{y}, \tag{5.2.3}$$

where $\mathbf{H}^H$ is the conjugate transpose of the channel matrix $\mathbf{H}$ and $\mathbf{y}$ is the received vector [98, 101]. The ZF estimator offers significant computational complexity reduction with tolerable performance degradation. The MMSE estimator, on the other hand, provides the best estimation mean square error among the set of all commonly used linear estimators. In the case that the noise $\mathbf{n}$ is additive white Gaussian noise, the MMSE estimation formula can be simplified as

$$\hat{\mathbf{x}} = (\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}_m)^{-1}\mathbf{H}^H\mathbf{y}, \tag{5.2.4}$$

where σ^2 is the variance of the noise, m is the number of the receiving antennas and $\mathbf{I}_m$ is the m-dimensional identity matrix [50, 98].

After computing the estimation vector $\hat{\mathbf{x}}$ of the transmit signal $\mathbf{x}$ from the received signal vector $\mathbf{y}$ by either ZF estimator (5.2.3) or MMSE estimator (5.2.4), the hard decision of determining the transmit bits is straightforward. We calculate a source bit by the sign of the real or imaginary part of the corresponding complex entry $\hat{x}_i$ in $\hat{\mathbf{x}}$ ($1 \le i \le m$), called QPSK demodulation [36, 98]. That is, the source bit is 1 if the corresponding part of the complex number is positive, and 0 if the part is negative.

We give an example of a 2×2 MIMO system. In the example, we show the whole signal processing circle that includes signal mapping, symbol transmission with AWGN, signal decoding and the hard decision of computing the transmit symbols. Suppose a 2×2 channel matrix is

$$\mathbf{H} = \begin{bmatrix} -0.99367 + 1.81176i & -0.88471 + 2.85039i \\ -0.54034 - 0.42801i & -2.32167 - 0.86847i \end{bmatrix}.$$

Assume zero forcing detection method is applied. We send four transmit signal bits "1100" to the receiver. By the QPSK constellation diagram, Figure 5.4, we first map the four transmit bits into two complex numbers $\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i$ and $-\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i$, corresponding to bits "11" and "00", respectively. Thus, the transmit vector is

$$\mathbf{x} = \begin{bmatrix} \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i & -\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i \end{bmatrix}^T.$$

By MIMO model formula (5.2.1), we get the following receive signal

$$\mathbf{y} = \begin{bmatrix} 0.84347 - 0.46420i & 1.45678 + 0.76190i \end{bmatrix}^T,$$

in which a noise

$$\mathbf{w} = \begin{bmatrix} 0.18609 + 0.34727i & 0.50864 - 0.80914i \end{bmatrix}^T,$$

is included. Then, we can compute the estimation $\hat{\mathbf{x}}$ of the source signal vector $\mathbf{x}$ by the ZF estimator (5.2.3):

$$\hat{\mathbf{x}} = \begin{bmatrix} 0.96973 - 0.46620i \\ -0.95459 - 0.04137i \end{bmatrix}, \quad \text{corresponding to} \quad \begin{bmatrix} \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i \\ -\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i \end{bmatrix}.$$

Notice that we determine the transmit bit by the sign of the estimation of a transmit signal. The signs of the four parts in the estimation vector $\hat{\mathbf{x}}$ are $+$, $-$, $-$ and $-$, respectively. Hence, the decoded receive bits are "1000", indicating that the second transmit bit is incorrect after the transmission.

Now, applying a lattice reduction algorithm, say the complex LLL algorithm [30], to the channel matrix $\mathbf{H}$, we obtain the complex unimodular matrix

$$\mathbf{Z} = \begin{bmatrix} -2 & -1+2i \\ 1 & 1-i \end{bmatrix}$$

and the reduced channel matrix

$$\mathbf{HZ} = \begin{bmatrix} 1.10263 - 0.77313i & -0.66418 - 0.06400i \\ -1.24099 - 0.01245i & -1.79378 + 0.80054i \end{bmatrix}.$$

Note that $\text{cond}(\mathbf{H}) \approx 6.2552$ while $\text{cond}(\mathbf{HZ}) \approx 1.9751$. Applying the ZF estimator to the

reduced channel matrix **HZ** and the receive signal **y**, we obtain

$$\begin{bmatrix} \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i \\ -\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}i \end{bmatrix}.$$

Premultiplying the above vector with the complex unimodular **Z** to transform it back to the original system, the transmit signal **x** is correctly decoded. This example illustrates how lattice reduction can improve the MIMO detection performance.

Since the LTE-Advanced standard (3GPP Release 11) supports maximum 8×8 MIMO [93], we use 8 transmit antennas and 8 receive antennas in our MIMO simulation. The entries of the channel matrices $\mathbf{H}_{8\times 8}$ are random Gaussian distributed complex numbers of zero-mean and unit variance. The signal noise ratio `SNR` varies from 2 dB to 20 dB. We generate $1,000$ random channel matrices in each `SNR`. For each channel matrix, we transmit $1,000,000$ random binary bits to the receiver.

Figure 5.6 shows our MIMO simulation results of the hybrid method, the LLL algorithm and the BKZ 2.0. The figure shows the average BER of the experiments in logarithm. We set the block size β of the BKZ 2.0 algorithm to 16 in our MIMO experiments. The simulation results show that our hybrid algorithm performs better than the LLL algorithm and the BKZ 2.0 with respect to BER in both ZF and MMSE estimation. The LLL algorithm and the BKZ 2.0 have almost the same BER and our hybrid algorithm improves around 0.5 dB.

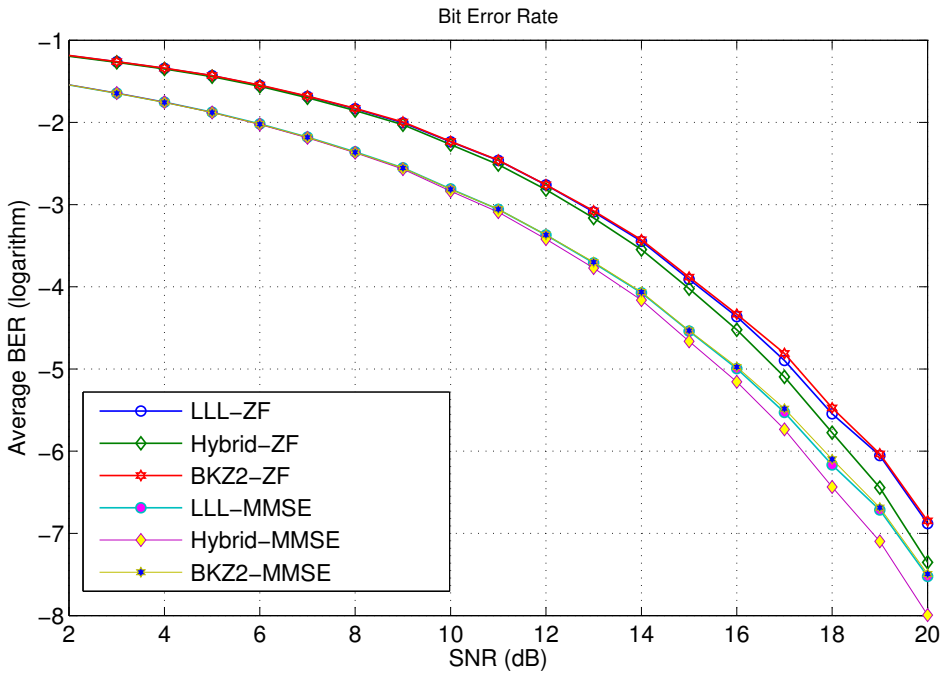


Figure 5.6: The BER performance (in logarithm) of ZF decoding and MMSE decoding for an 8×8 complex-valued MIMO system

5.3 Lattice Based Cryptography

Lattice-based strategies have successfully involved in many applications in security, such as GGH encryption scheme and NTRU signature scheme. People also use lattice-reduction technique to attack traditional cryptography systems like RSA and knapsack [13, 24, 27, 63]. Comparing with the non-lattice cryptographic schemes, such as RSA and ElGamal, lattice-based cryptographic schemes show advantages in simplicity, efficiency and average-case hardness. Moreover, lattice-based cryptography is believed to be secure against quantum computers [11, 56, 66].

Most lattice-based cryptosystems are constructed on the presumed hardness of lattice problems: the shortest vector problem (SVP) and the closest vector problem (CVP). For example, the security of Ajtai-Dwork (AD) cryptosystem is based on solving SVP, and the Goldreich-Goldwasser-Halevi (GGH) cryptosystem is based on the hardness of solving CVP [4, 17]. It has been proved that SVP is NP-hard under randomized lattices, and solving CVP is slightly harder than solving SVP [2, 3, 35, 51].

The GGH cryptosystem can be regarded as a lattice analogue of the McEliece cryptosystem based on the algebraic coding theory. It presents an intuitive encryption scheme of designing a closest vector problem on the given lattice [17, 63].

Designing a GGH cryptosystem includes finding a private key, constructing a public key, and choosing a proper bias vector, shown in the following steps [27, 44]:

1. Private key generation

To generate a private key for a GGH cryptosystem, we can compute a "good" basis matrix $\mathbf{V}_{m\times n}$ for a lattice L that consists nearly orthogonal basis vectors, i.e., the orthogonality defect $\delta(\mathbf{V})$ is close to 1. Then, we keep the basis matrix $\mathbf{V}$ as the private key of the constructed GGH cryptosystem.

2. Public key generation

A public key $\mathbf{W}_{m\times n}$ is a "bad" basis for the lattice $L(\mathbf{V})$, i.e., $\delta(\mathbf{W})$ is very large. To compute a basis matrix with large orthogonality defect for $L(\mathbf{V})$, we can first compute a unimodular matrix $\mathbf{Z}$, and compute

$$\mathbf{W} = \mathbf{V}\mathbf{Z}. \tag{5.3.1}$$

Then, we check the orthogonality defect $\delta(\mathbf{W})$. If $\delta(\mathbf{W})$ is not large enough, we repeat the above iteration by computing a new unimodular matrix $\mathbf{Z}$ and multiplying it with $\mathbf{W}$ computed in (5.3.1) of the previous iteration, until $\delta(\mathbf{W})$ satisfies our expectation. After that, we publish $\mathbf{W}$ as the public key of the GGH cryptosystem.

There are several practical methods to compute the public key $\mathbf{W}$, improving the original method by either efficiency or the size (in bits) of the public key. For example, S. Qiao introduced an efficient method to compute $\mathbf{W}$ within one iteration by constructing the unimodular matrix $\mathbf{Z}$ from the eigenvalues of an ill-conditioned matrix. The size of the public key $\mathbf{W}$ computed by iteratively finding and multiplying $\mathbf{Z}$ may be large. Alternately, Micciancio proposed a Hermite Normal Form

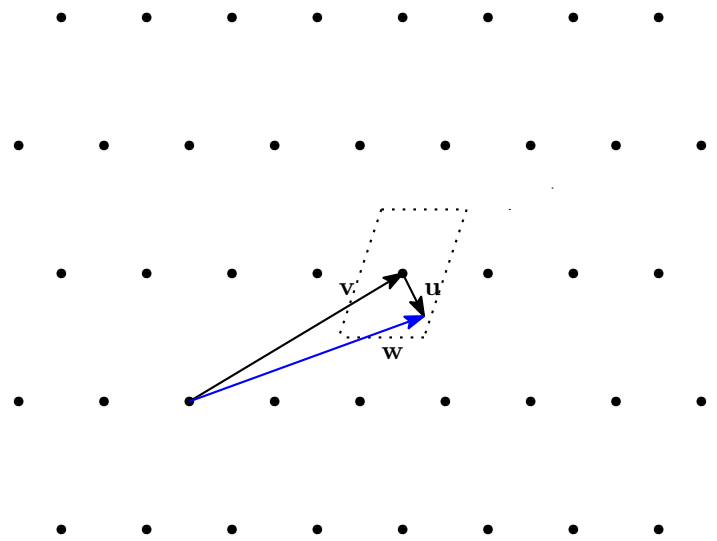


Figure 5.7: Encrypt a plain text vector **p** to the cipher text vector **e**

(HNF) method to compute the public key **W** with relatively small size by computing the HNF of **V** [65].

3. Encryption process

   The encryption process is to construct a non-lattice vector, such that we can easily find its correct closest vector in the lattice using the private key **V**. Let $\mathbf{W}_{m\times n}$ be the public key of the GGH cryptosystem, and let **p** be an $n \times 1$ plain text vector. We encrypt the plain text **p** by

$$\mathbf{e} = \mathbf{W}\mathbf{p} + \mathbf{r}, \tag{5.3.2}$$

   where **e** is the encrypted $m \times 1$ cipher text vector, and **r** is an $m \times 1$ bias vector, shown as Figure 5.7. In the original paper of the GGH cryptography, each entry of

the bias vector $\mathbf{r}$ is randomly set to either $+\sigma$ or $-\sigma$, with equal probability 50%, where σ is a security parameter [34]. Nguyen proved that the choice of either $+\sigma$ or $-\sigma$ was insecure [69, 70, 74]. Thus in (5.3.2), we use Nguyen's improvement to define $\mathbf{r}$,

$$\mathbf{r} = (r_1, r_2, \ldots, r_m)^T, \tag{5.3.3}$$

where r_i is uniformly distributed between $[-\sigma, \sigma]$, for all $1 \leq i \leq m$ [69, 100]. Geometrically, the Euclidean length of $\mathbf{r}$ should be less than a half of the shortest distance between any two adjacent points in the lattice L. Hence, the vector $\mathbf{Wp}$ is the closest vector in the lattice to the vector $\mathbf{e}$. The criteria for choosing $\mathbf{r}$ highly influence the security of the GGH cryptosystem, for example, the range of the security parameter σ, or the form of $\mathbf{r}$ [65, 73]. In practice, we can consider sending the cipher text $\mathbf{e}$ together with its hashed value to ensure the correctness of the cipher text.

4. Decryption process

   The theoritical security of GGH is that it is hard to find the closest vector in a lattice to a given non-lattice point.

   The decryption process includes two steps. Firstly, we find the closest lattice vector $\mathbf{u}$ of the cipher text vector $\mathbf{e}$ using the private key $\mathbf{V}$. We compute $\mathbf{u}$ by Babai's algorithm [7]

$$\mathbf{u} = \mathbf{V}[\mathbf{V}^{-1}\mathbf{e}]. \tag{5.3.4}$$

Since $\mathbf{u} = \mathbf{W}\mathbf{p}$, we can then compute the plain text $\mathbf{p}$ by the public key

$$\mathbf{p} = \mathbf{W}^{-1}\mathbf{u}. \tag{5.3.5}$$

If we apply the public key $\mathbf{W}$ directly on (5.3.4), the closest vector $\mathbf{u}$ cannot be correctly discovered. Thus, the plain text $\mathbf{p}$ is safe [34, 70].

Known attacks on GGH cryptosystem involve with the two aspects of the original definition. The first attacking method uses the behavior of the bias vector $\mathbf{r}$ in the original definition that each entry of $\mathbf{r}$ is either $+\sigma$ or $-\sigma$. We can avoid this attacking method by defining $\mathbf{r}$ as (5.3.3). Nguyen presents another flaw in the design of the GGH scheme, by which we can reduce the decryption to a special closest vector problem using lattice reduction technique, such that decrypting the cipher text is much easier than the general CVP problem. Nguyen shows that the GGH cryptosystem is insecure for dimension up to 350 by attacking it using the LLL algorithm [65, 69]. In our GGH simulation, we attack the GGH cryptosystem using the lattice reduction technique pointed out by Nguyen, i.e., reduce the public key $\mathbf{W}$ and then try to find the closet lattice vector to $\mathbf{e}$ using the reduced $\mathbf{W}$.

By the lattice invariant of Minkowski's theorem, for an n-dimensional lattice L, there exists a Hermite's constant γ_n, such that we have $\lambda_1 \leq \sqrt{\gamma_n}(\det L)^{1/n}$, where λ_1 is the length of a shortest vector in the lattice [19, 44]. Since the Hermite's constant γ_n is directly related to the lattice dimension n, an approximation factor α is then introduced

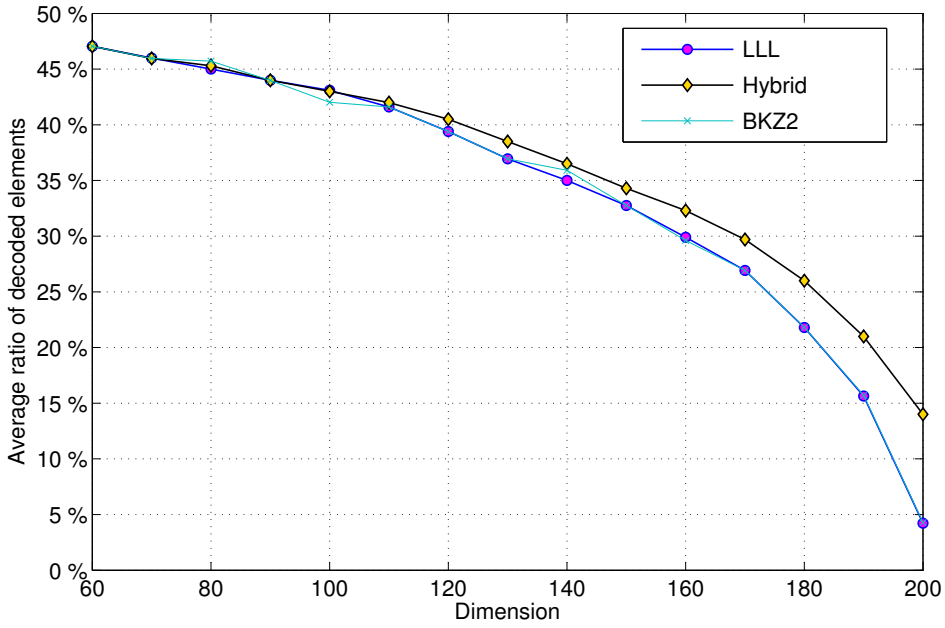


Figure 5.8: The performance of attacking a GGH cryptosystem by lattice reduction algorithms

to evaluate the approximation of γ_n that a lattice reduction algorithm can achieve. The LLL algorithm and the BKZ algorithm can roughly achieve φ^n for some φ, which are still exponential to the dimension of the lattice. In practice, the LLL algorithm and the BKZ algorithm can achieve $\varphi \approx 1.022$ and $\varphi \approx 1.012$, respectively [29, 74]. Therefore, in our experiments, we can safely use 1^n as a loose lower bound for γ_n. Furthermore, we use the approximation $\gamma_n = 1$ to estimate the above safety lower bound of λ_1, such that the length of the bias vector $\mathbf{r}$ satisfies $\|\mathbf{r}\|_2 < \lambda_1/2 \leq \frac{1}{4}(\det L)^{1/n}$ after choosing the parameter σ, i.e., to ensure that the lattice point closest to $\mathbf{e}$ is $\mathbf{Wp}$.

We attack the GGH cryptosystem using our hybrid method, the LLL algorithm and the BKZ 2.0 with block size 20 in the simulation. The dimension of the GGH cryptosystem we constructed varies from 60 to 200 with interval 10. We set the maximum of the orthogonality defect of the public key $\mathbf{W}$ to 1.0×10^4. Practically, the dimension of a GGH cryptosystem can be as large as 350 to 400 to achieve high security, and the orthogonality defect of $\mathbf{W}$ is usually much larger than our simulation choice 1.0×10^4. In such case, the numbers of the decode elements are too small to be distinguished among the three algorithms. For each dimension, we generated 1000 GGH instances; for each GGH instance, we encrypted and decrypted 1000 integral plain text vector. For every generated GGH instance, we used the lattice reduction algorithms to reduce the public key $\mathbf{W}$, and decrypt the cipher text $\mathbf{e}$ by (5.3.5) using our reduced public keys.

In practice, revealling the full cipher text may not be necessary. Preventing partial decryption of cipher texts (or other information leakage) is also critical for a cryptosystem. Decrypting partial information from the cipher text can be a valuable attack [14, 18, 26]. For example, the Japanese purple cipher was first and partially broken in August, 1940 by U.S. Navy cryptanalysts in Hawaii. The U.S. army then revealed the location of impending attack on Midway Island. Despite that only fewer than 15% of Japanese messages is broken, the broken Japanese purple cipher messages showed significant impact on World War II [5, 88]. Therefore, in our experiments, we compare the average rate of the successfully decrypted plain text elements in the encrypted vector by the three algorithms against the same GGH cryptosystem.

Figure 5.8 shows the experimental results of attacking the GGH cryptosystem. The average decryption rate of attacking simulations indicates that our hybrid method can discover more information than the LLL algorithm and the BKZ 2.0. We remark that the bias vector $\mathbf{r}$ in our simulation is relatively small comparing with the selection of $\mathbf{r}$ in practice. Thus, the average rate of the decrypted elements in our experiment are relatively higher than the GGH cryptosystem in practice. This high success rate can be decreased by making the security parameter σ bigger.

Chapter 6

Conclusion and Future Work

The main contribution of this thesis is the polynomial time hybrid method for lattice basis reduction. We showed that its complexity is $O(n^4 \log B)$ for integer basis matrices, where n is the dimension of the lattice and B is the maximum length of the input basis vectors. The hybrid method has the same complexity as the well-known LLL algorithm.

In this thesis, we first described the generic Jacobi method (the Gaussian algorithm) with unknown convergence. To ensure convergence, we introduced a parameter ω into the condition for the Lagrange reduction in the generic Jacobi method. We showed that the complexity of the conditional Jacobi method is $O(n^4 \log B)$, the same as the complexity of the LLL algorithm. To improve the quality, especially the condition number, of the computed bases, we proposed a hybrid method by integrating the conditional size reduction into the conditional Jacobi method. We proved that the complexity of the hybrid method is the same as that of the conditional Jacobi method, that is, $O(n^4 \log B)$.

We compared our hybrid method with the widely used LLL algorithm and the BKZ 2.0 algorithm in three-part experiments. In the first part, we tested the three algorithms on random matrices. Our experimental results show that the hybrid method consistently produced bases with smaller orthogonality defect and smaller condition number than the bases computed by the LLL algorithm and the BKZ 2.0. The difference between our algorithm and the other two methods grows as the problem size increases. Despite that the worst case complexity of our hybrid method is the same as that of the LLL algorithm, our algorithm ran faster than the LLL algorithm in our experiments. In the second part, we simulated a communication system consisting of 8 transmit antennas and

8 receive antennas, which is the maximum number of antennas that the LTE standard supports for MIMO systems. Our experiments show that the communication channels improved by our proposed hybrid method performed lower bit error rate than both the LLL algorithm and the BKZ 2.0. Lastly, we simulated the attacks against the GGH cryptosystem. Our experiments show that after reducing the public key matrices by the three algorithms, the hybrid method discovered more plain texts than the the LLL algorithm and the BKZ 2.0 in the GGH attack simulations.

6.1 Future Work

1. Derive theoretical boundary of the length of the shortest vector in the computed basis

   In the hybrid method, after introducing a fixed parameter $1/\sqrt{3} \leq \omega < 1$ into equation (3.2b), each iteration reduces the length of a vector with a reduction factor τ, which is small than or equal to ω. Let $\mathbf{a}_1$ be the shortest vector in the basis computed by the hybrid method, the theoretical upper bound of the length of $\mathbf{a}_1$ is still unknown.

2. Parallel implementation

   The Jacobi method is inherently parallel. Jeremic and Qiao give an parallel implementation of the generic Jacobi method using GPU [46]. Our future work includes parallel implementations of the hybrid method.

3. Improve performance on extremely ill-conditioned basis matrices

   Experimental results show that the proposed hybrid method does not perform very well on extremely ill-conditioned basis matrices.

Bibliography

[1] On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[2] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108. ACM, 1996.

[3] Miklós Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 10–19. ACM, 1998.

[4] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 284–293. ACM, 1997.

[5] NSA analysts' modern-day attempt to duplicate solving the Red and Purple ciphers. Red and Purple: A Story Retold. *Cryptologic Quarterly Article (NSA)*, Vol.

3(Nos. 3-4):63–80, Fall/Winter 1984-1985.

[6] Harold Artes, D. Seethaler, and F. Hlawatsch. Efficient detection algorithms for MIMO channels: a geometrical approach to approximate ML detection. *Signal Processing, IEEE Transactions on*, 51(11):2808–2820, Nov 2003.

[7] L. Babai. On Lovasz lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.

[8] L. Bai and J. Choi. *Low Complexity MIMO Detection*. Springer New York, 2012.

[9] Dushyantha A. Basnayaka, Peter J. Smith, and Phillipa A. Martin. Performance Analysis of Macrodiversity MIMO Systems with MMSE and ZF Receivers in Flat Rayleigh Fading. *Wireless Communications, IEEE Transactions on*, 12(5):2240–2251, May 2013.

[10] Lars T Berger, Andreas Schwager, Pascal Pagani, and Daniel M Schneider, editors. *MIMO Power Line Communications: Narrow and Broadband Standards, EMC, and Advanced Processing*. CRC Press, 2014.

[11] Daniel J. Bernstein. Introduction to post-quantum cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 1–14. Springer Berlin Heidelberg, 2009.

[12] Girish S. Bhat and Carla D. Savage. Balanced gray codes. *Electr. J. Comb.*, 3(1), 1996.

[13] Dan Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, 46:203–213, 1999.

[14] Dan Boneh, Xavier Boyen, and Shai Halevi. Chosen Ciphertext Secure Public Key Threshold Encryption Without Random Oracles. In David Pointcheval, editor, *Topics in Cryptology – CT-RSA 2006: The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2005. Proceedings*, pages 226–243, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[15] L. Bruderer, C. Studer, M. Wenk, D. Seethaler, and A. Burg. VLSI implementation of a low-complexity LLL lattice reduction algorithm for MIMO detection. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 3745–3748, May 2010.

[16] D. Buell. *Algorithmic Number Theory: 6th International Symposium, ANTS-VI, Burlington, VT, USA, June 13-18, 2004, Proceedings*. Lecture Notes in Computer Science. Springer, 2004.

[17] Anne Canteaut and Nicolas Sendrier. Cryptanalysis of the original McEliece cryptosystem. In Kazuo Ohta and Dingyi Pei, editors, *Advances in Cryptology — ASIACRYPT'98*, volume 1514 of *Lecture Notes in Computer Science*, pages 187–199. Springer Berlin Heidelberg, 1998.

[18] Zhenfu Cao. *New Directions of Modern Cryptography*. CRC Press, 2013.

[19] John William Scott Cassels. *An introduction to the geometry of numbers*. Springer Science & Business Media, 2012.

[20] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In DongHoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20. Springer Berlin Heidelberg, 2011.

[21] H. Cohen. *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics. Springer, 1993.

[22] Aharonov Dorit and Regev Oded. Lattice problems in NP and co-NP. *Journal of the ACM (JACM)*, 52:749–765, September 2005.

[23] A. Doucet, A. Smith, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Information Science and Statistics. Springer New York, 2013.

[24] Léo Ducas and PhongQ. Nguyen. Learning a zonotope and more: Cryptanalysis of NTRUSign countermeasures. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 433–450. Springer Berlin Heidelberg, 2012.

[25] B. Farhang-Boroujeny, Haidong Zhu, and Zhenning Shi. Markov chain Monte Carlo algorithms for CDMA and MIMO communication systems. *Signal Processing, IEEE Transactions on*, 54(5):1896–1909, May 2006.

[26] Pierre-Alain Fouque, Guillaume Poupard, and Jacques Stern. Sharing Decryption in the Context of Voting or Lotteries. In Yair Frankel, editor, *Financial Cryptography: 4th International Conference, FC 2000 Anguilla, British West Indies, February 20–24, 2000 Proceedings*, pages 90–104, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[27] Steven D. Galbraith. *Mathematics of public key cryptography*. Cambridge University Press, 2012.

[28] Nicolas Gama, Nick Howgrave-Graham, Henrik Koy, and Phong Q. Nguyen. Rankin's constant and blockwise lattice reduction. In *CRYPTO*, pages 112–130, 2006.

[29] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In *Advances in Cryptology – Proc. Eurocrypt '08*, Lecture Notes in Computer Science. Springer, 2008.

[30] Ying Hung Gan, Cong Ling, and Wai Ho Mow. Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection. *IEEE Transactions on Signal Processing*, 57(7):2701–2710, July 2009.

[31] Ying Hung Gan and Wai Ho Mow. Complex lattice reduction algorithms for low-complexity MIMO detection. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 5, pages 5 pp.–2957, Dec 2005.

[32] C.F. Gauss and H. Maser. *Untersuchungen über höhere Arithmetik*. AMS Chelsea Publishing Series. AMS Chelsea Publishing, 1889.

[33] Amitava Ghosh, Rapeepat Ratasuk, Bishwarup Mondal, Nitin Mangalvedhe, and Tim Thomas. LTE-advanced: Next-generation Wireless Broadband Technology. *Wireless Commun.*, 17(3):10–22, June 2010.

[34] O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. Technical report, Cambridge, MA, USA, 1996.

[35] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and J-P Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.

[36] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, New York, NY, USA, 2005.

[37] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[38] Jerry R. Hampton. *Introduction to MIMO Communications*. Cambridge University Press, 2013.

[39] Daewan Han, Myung-Hwan Kim, and Yongjin Yeom. Cryptanalysis of the Paeng-Jung-Ha cryptosystem from PKC 2003. In *Proceedings of the 10th international*

conference on Practice and theory in public-key cryptography, PKC'07, pages 107–117, Berlin, Heidelberg, 2007. Springer-Verlag.

[40] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Analyzing Blockwise Lattice Algorithms Using Dynamical Systems. In Phillip Rogaway, editor, *Proceedings of the International Cryptology Conference on Advances in Cryptology*, CRYPTO'11, pages 447–464. Springer Berlin Heidelberg, 2011.

[41] Guillaume Hanrot and Damien Stehlé. Improved analysis of Kannan's shortest lattice vector algorithm. In *Proceedings of the International Cryptology Conference on Advances in Cryptology*, CRYPTO'07, pages 170–186, Berlin, Heidelberg, 2007. Springer-Verlag.

[42] Babak Hassibi and Haris Vikalo. On the sphere-decoding algorithm I. expected complexity. *IEEE Trans. Sig. Proc*, pages 2806–2818, 2005.

[43] Bettina Helfrich. Algorithms to construct Minkowski reduced and Hermite reduced lattice bases. *Theoretical Computer Science*, 41:125 – 139, 1985.

[44] J. Hoffstein, J.C. Pipher, and J.H. Silverman. *An introduction to mathematical cryptography*. Undergraduate texts in mathematics. Springer, 2008.

[45] J. Jalden, D. Seethaler, and G. Matz. Worst- and average-case complexity of LLL

lattice reduction in MIMO wireless systems. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 2685–2688, March 2008.

[46] Filip Jeremic and Sanzheng Qiao. A parallel Jacobi-type lattice basis reduction algorithm. *International Journal of Numerical Analysis and Modeling, Series B*, 5(1-2):1–12, 2014.

[47] M. Jünger, T.M. Liebling, D. Naddef, G.L. Nemhauser, W.R. Pulleyblank, G. Reinelt, G. Rinaldi, and L.A. Wolsey. *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*. Springer, 2009.

[48] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, STOC '83, pages 193–206, New York, NY, USA, 1983. ACM.

[49] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12:415–440, August 1987.

[50] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[51] Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM (JACM)*, 52(5):789–808, 2005.

[52] A. Korkine and G. Zolotareff. Sur les formes quadratiques. *Mathematische Annalen*, 6:366–389, 1873.

[53] Brian A. LaMacchia. Basis reduction algorithms and subset sum problems. Technical report, Cambridge, MA, USA, 1991.

[54] A.K. Lenstra, Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.

[55] H.W. Lenstra. *Integer programming with a fixed number of variables*. Report. Department of Mathematics. University of Amsterdam. Department, Univ., 1981.

[56] Christoph Ludwig. *A Faster Lattice Reduction Method Using Quantum Search*, pages 199–208. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[57] Franklin T. Luk and Sanzheng Qiao. Numerical properties of the LLL method. In *Proceedings of SPIE, the International Society for Optical Engineering, 26-27 August 2007*, volume 6697, pages 669703–669703–7, San Diego, California, USA.

[58] Franklin T. Luk and Sanzheng Qiao. Conditioning properties of the LLL algorithm. In *Mathematics for Signal and Information Processing*, volume 7444, pages 7444–17. Proc. of SPIE, 2009.

[59] Franklin T. Luk, Sanzheng Qiao, and Wen Zhang. A lattice basis reduction algorithm. Technical report, Institute for Computational Mathematics Hong Kong Baptist University, 2010.

[60] Franklin T. Luk and Daniel M. Tracy. An improved LLL algorithm. *Linear Algebra and its Applications*, 428(2-3):441 – 452, 2008.

[61] J. Martinet. *Perfect Lattices in Euclidean Spaces*. Springer-Verlag, Berlin, 2003.

[62] V. Mazya and T.O. Shaposhnikova. *Jacques Hadamard: A Universal Mathematician*. History of mathematics. American Mathematical Society, 1999.

[63] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996.

[64] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*. Milken Institute Series on Financial Innovation and Economic Growth. Springer US, 2002.

[65] Daniele Micciancio. Improving lattice based cryptosystems using the Hermite normal form. In *Cryptography and Lattices*, pages 126–145. Springer, 2001.

[66] Daniele Micciancio. Lattice-based cryptography. pages 713–715, 2011.

[67] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.

[68] H. Minkowski. Discontinuity region for arithmetical equivalence. *J. reine Angew*, (129):220–274, 1905.

[69] Phong Nguyen. *Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97*, pages 288–304. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[70] Phong Nguyen. Lattice reduction algorithms: Theory and practice. In Kenneth Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 2–6. Springer Berlin / Heidelberg, 2011.

[71] Phong Q. Nguyen and Damien Stehlé. *LLL on the Average*, pages 238–256. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[72] Phong Q. Nguyen and Damien Stehlé. Low-dimensional lattice basis reduction revisited. *ACM Trans. Algorithms*, 5(4):46:1–46:48, November 2009.

[73] Phong Q Nguyen and Jacques Stern. Lattice Reduction in Cryptology: An Update. In *Algorithmic number theory*, pages 85–112. Springer, 2000.

[74] Phong Q Nguyen and Jacques Stern. The two faces of lattices in cryptology. In *Cryptography and lattices*, pages 146–180. Springer, 2001.

[75] Phong Q. Nguyen and Brigitte Valle. *The LLL Algorithm: Survey and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[76] A. Papoulis and S.U. Pillai. *Probability, random variables, and stochastic processes*. McGraw-Hill electrical and electronic engineering series. McGraw-Hill, 2002.

[77] J.G. Proakis and M. Salehi. *Digital Communications*. McGraw-Hill International Edition. McGraw-Hill, 2008.

[78] Sanzheng Qiao. A Jacobi method for lattice basis reduction. In *Proceedings of 2012 Spring World Congress on Engineering and Technology (SCET2012)*, Vol.2. IEEE, pages 649–652, Xi'an China, May 2012.

[79] Gregory G Raleigh and John M Cioffi. Spatio-temporal Coding for Wireless Communication. *Communications, IEEE Transactions on*, 46(3):357–366, 1998.

[80] H. Sampath, Petre Stoica, and A. Paulraj. Generalized linear precoder and decoder design for MIMO channels using the weighted MMSE criterion. *Communications, IEEE Transactions on*, 49(12):2198–2206, Dec 2001.

[81] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, August 1987.

[82] C. P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 1994.

[83] D. Seethaler, G. Matz, and F. Hlawatsch. Low-complexity mimo data detection

using seysen's lattice reduction algorithm. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 3, pages III–53–III–56, April 2007.

[84] Igor A. Semaev. A 3-dimensional lattice reduction algorithm. In *Cryptography and Lattices International Conference (CaLC) 2001 Providence, RI, USA, March 29–30, 2001*, pages 181–193, Berlin, Heidelberg. Springer Berlin Heidelberg.

[85] M. Seysen. Simultaneous reduction of a lattice basis and its reciprocal basis. *Combinatorica*, 13(3):363–376, 1993.

[86] V. Shoup. NTL: A library for doing number theory, Feb. 2013.

[87] Bernard Sklar and Communications Engineering Services. Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization. *IEEE Communications Magazine*, pages 90–100, 1997.

[88] M. Smith. *The Emperor's Codes: The Breaking of Japan's Secret Ciphers*. Arcade Pub., 2001.

[89] Q.H. Spencer, A.L. Swindlehurst, and M. Haardt. Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels. *Signal Processing, IEEE Transactions on*, 52(2):461–471, Feb 2004.

[90] M. Taherzadeh, A. Mobasher, and A.K. Khandani. LLL reduction achieves the

receive diversity in MIMO decoding. *Information Theory, IEEE Transactions on*, 53(12):4801–4805, Dec. 2007.

[91] M. Taherzadeh, A. Mobasher, and A.K. Khandani. LLL Reduction Achieves the Receive Diversity in MIMO Decoding. *Information Theory, IEEE Transactions on*, 53(12):4801–4805, Dec 2007.

[92] Agilent Technologies and Moray Rumney. *LTE and the Evolution to 4G Wireless: Design and Measurement Challenges*. Wiley Publishing, 2nd edition, 2013.

[93] The 3rd Generation Partnership Project. LTE-Advanced standard (3GPP Release 11). `http://www.3gpp.org/specifications/releases/69-release-11`, March 2013.

[94] Zhaofei Tian and Sanzheng Qiao. A complexity analysis of a Jacobi method for lattice basis reduction. In *Proceedings of the Fifth International C* Conference on Computer Science and Software Engineering*, C3S2E '12, pages 53–60, New York, NY, USA, 2012. ACM.

[95] Zhaofei Tian and Sanzheng Qiao. An enhanced Jacobi method for lattice-reduction-aided MIMO detection. In *Proceedings of IEEE China Summit and International Conference on Signal and Information Processing*, pages 39–43, Beijing, China, June 8-10 2013. IEEE.

[96] Zhaofei Tian and Sanzheng Qiao. A hybrid method for lattice basis reduction.

Technical Report CAS-14-01-SQ, Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada, L8S 4K1, Jan. 2014.

[97] Zhaofei Tian, Wen Zhang, and Sanzheng Qiao. A polynomial time Jacobi method for lattice basis reduction. Technical Report CAS-12-04-SQ, Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada, L8S 4K1, Dec. 2012.

[98] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, New York, NY, USA, 2005.

[99] Brigitte Vallée and Antonio Vera. Lattice reduction in two dimensions: analyses under realistic probabilistic models. *Discrete Mathematics and Theoretical Computer Science*, DMTCS Proceedings vol. AH, 2007 Conference on Analysis of Algorithms (AofA 07), Jan 2007.

[100] Brigitte Vallée and Antonio Vera. Probabilistic analyses of lattice reduction algorithms. In *The LLL Algorithm*, Information Security and Cryptography, pages 71–143. Springer Berlin Heidelberg, 2010.

[101] Cheng Wang, E.K.S. Au, R.D. Murch, Wai Ho Mow, R.S. Cheng, and V. Lau. On the Performance of the MIMO Zero-Forcing Receiver in the Presence of Channel Estimation Error. *Wireless Communications, IEEE Transactions on*, 6(3):805–810, March 2007.

[102] D. Wubben, R. Bohnke, V. Kuhn, and K. D Kammeyer. MMSE-based lattice-reduction for near-ML detection of MIMO systems. In *Smart Antennas, 2004. ITG Workshop on*, pages 106–113, March 2004.

[103] D. Wubben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer. Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice reduction. In *Communications, 2004 IEEE International Conference on*, volume 2, pages 798 – 802 Vol.2, June 2004.

[104] D. Wubben and D. Seethaler. On the Performance of Lattice Reduction Schemes for MIMO Data Detection. In *2007 Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers*, pages 1534–1538, Nov 2007.

[105] D. Wübben, D. Seethaler, J. Jalden, and G. Matz. Lattice reduction: A survey with applications in wireless communications. *IEEE Signal Processing Magazine*, 28(3):70–91, May 2011.

[106] P. Xu, C. Shi, and J. Liu. Integer estimation methods for GPS ambiguity resolution: an applications oriented review and improvement. *Survey Review*, 44:59–71, Jan. 2012.

[107] Wen Zhang, Sanzheng Qiao, and Yimin Wei. A Diagonal Lattice Reduction Algorithm for MIMO Detection. *Signal Processing Letters, IEEE,* 19(5):311–314, May 2012.

[108] Wen Zhang, Sanzheng Qiao, and Yimin Wei. HKZ and Minkowski reduction algorithms for Lattice-Reduction-Aided MIMO detection. *IEEE Transactions on Signal Processing*, 60(11):5963–5976, 2012.