Bridging Theory and Application: A Journey from Minkowski's Theorem to GGH Cryptosystems in Lattice Theory

Danzhe Chen

Claremont McKenna College

Follow this and additional works at: https://scholarship.claremont.edu/cmc_theses

Part of the Number Theory Commons

Recommended Citation

https://scholarship.claremont.edu/cmc_theses/3681

This Open Access Senior Thesis is brought to you by Scholarship@Claremont. It has been accepted for inclusion in this collection by an authorized administrator. For more information, please contact scholarship@claremont.edu.
Senior Thesis

Bridging Theory and Application: A Journey from
Minkowski’s Theorem to GGH Cryptosystems in Lattice Theory

Submitted to
Professor Lenny Fukshansky

By
Danzhe Chen

Spring 2024
April 22, 2024

Department of Mathematics
Claremont McKenna College
Abstract

This thesis provides a comprehensive exploration of lattice theory, emphasizing its dual significance in both theoretical mathematics and practical applications, particularly within computational complexity and cryptography. The study begins with an in-depth examination of the fundamental properties of lattices and progresses to intricate lattice-based problems such as the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). These problems are analyzed for their computational depth and linked to the Subset Sum Problem (SSP) to highlight their critical roles in understanding computational hardness. The narrative then transitions to the practical applications of these theories in cryptography, evaluating the shift from traditional cryptosystems like RSA to sophisticated lattice-based alternatives, including the GGH cryptoscheme. This shift is particularly relevant in the context of emerging quantum computing threats, where lattice-based cryptosystems offer a promising frontier for secure communications. Through detailed analysis, this thesis not only advances the academic discussion on lattices but also underscores their crucial impact on the evolution of cryptographic methods, bridging the gap between abstract mathematical concepts and their real-world cryptographic applications.
Contents

1 Introduction 4

2 Introduction to Lattices 5
   2.1 Basic Definitions and Properties ........................................... 5
   2.2 Successive minima ................................................................. 8
   2.3 Computational lattice problems ............................................... 18

3 Computational Complexity 21
   3.1 Basic Notion ................................................................. 21
   3.2 Cook-Levin Theorem .......................................................... 22
   3.3 Knapsack and SSP problems ................................................. 24

4 Lattice Problems in Computational Complexity 26
   4.1 SVP and CVP ................................................................. 26
   4.2 CVP is NP-complete ............................................................ 27
   4.3 Gauss-Lagrange algorithm in dimension 2 ................................. 28
   4.4 The LLL approximation algorithm ......................................... 30

5 Lattice Cryptography Applications 34
   5.1 Introduction to asymmetric cryptography .................................. 34
   5.2 The Rivest-Shamir-Adleman (RSA) Cryptosystem ......................... 35
   5.3 The Goldreich-Goldwasser-Halevi (GGH) cryptosystem .................. 37
   5.4 Conclusion ................................................................. 38
Acknowledgments

This thesis is affectionately dedicated to my family, friends, professors, and all who have supported me, both passively and actively, through this journey.

First and foremost, immense gratitude is extended to my thesis advisor, Prof. Lenny Fukshansky. Thank you, Professor, for eagerly agreeing to guide me through this intellectual journey and for your unwavering support and encouragement. While I may not be able to unravel the high-dimension optimal coverings anytime soon, your introduction to such a captivating topic has consistently illuminated the beauty and elegance of mathematics for me.

Second, I owe a profound debt of gratitude to my parents. It is unequivocally true that without your nurturing and backing in all my choices, none of this would have been achievable. Thank you for everything you have done in raising me and adventuring with me, sharing in both the challenges and joys along the way.

Third, my heartfelt thanks go out to all my friends. No matter where you are now, our conversations, our intersections, and the times we spent together have significantly shaped who I am, motivating me to appreciate both the common ground and differences among humans. You hold a special place in my heart.
1 Introduction

What is a lattice? You can start to imagine a regular grid consisting of many intersection points and then expand that to be infinite and n-dimensional. The seemingly simple geometric structure, however, has great depth that had trapped the most brilliant mathematicians for centuries. I am not one of them, but the beauty of lattices got me as well. The study of lattices occupies a unique place at the intersection of mathematics and computer science, embodied by both the abstract beauty of mathematical structures and practical applications in cryptography. This thesis embarks on an exploratory journey into the world of lattices, aiming to unveil their theoretical significance and application relevance, particularly in the realms of computational mathematics and digital security.

In Chapter 1, I will start with basic definitions and properties of lattices. Then I will introduce successive minima as a significant way to measure the “tightness” of a lattice and how Minkowski’s Theorem approaches it from the perspective of geometric number theory. It follows some of the most intriguing problems in lattice theory to provide a segue into their applications in computational complexity and cryptography. In Chapter 2, the fundamental notion of computational complexity will be explained first. I will prove the NP-completeness of integer knapsack and its variant, the Subset Sum Problem (SSP), from the Boolean Satisfiability Problem (SAT). SSP is instrumental in solving complex lattice-based problems.

In Chapter 3, with all the prior knowledge set up, I will introduce several lattice problems and focus on two central problems which are the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). The NP-completeness of CVP will be attained by SSP. Despite computational hardness, certain algorithms have been developed to approximate solutions to SVP and CVP. Two major algorithms will be introduced, the Gauss-Lagrange Algorithm and Lenstra-Lenstra-Lovász (LLL) Approximation Algorithm. The algorithms open up a window for practical implications of lattice problems, especially in cryptography. In Chapter 4, I will provide an overview of how cryptography helps keep the confidentiality and the role of computational hardness in achieving that. Some famous cryptoschemes such as RSA will be emphasized. And finally, I will explain the use of lattice theory in producing the GGH cryptoscheme, one of the most special existing cryptoschemes.

The exposition in this thesis is heavily based on Prof. Fukshansky’s lecture notes [Fuk22] and the book Complexity of Lattice Problems: A Cryptographic Perspective [MG02]. These resources have provided invaluable insights and foundational concepts that have shaped the structure and content of this work, enabling a thorough exploration of lattice theory and its applications in cryptography.
# 2 Introduction to Lattices

## 2.1 Basic Definitions and Properties

**DEFINITION 3.1.1.** $\mathbb{R}^n$ is defined as the $n$-dimensional Euclidean space. A lattice $\Lambda$ of rank $r$ in $\mathbb{R}^n$ is the set

$$\Lambda = \text{span}_\mathbb{Z}(a_1, \ldots, a_r) = \left\{ \sum_{i=1}^{r} x_i a_i : x_i \in \mathbb{Z} \text{ for all } 1 \leq i \leq r \right\}$$

of all possible integral combinations of the linear independent vectors $a_1, \ldots, a_r$ in $\mathbb{R}^n$ ($1 \leq r \leq n$). The set $a_1, \ldots, a_r$ is called a *basis* for $\Lambda$. And the integers $r$ and $n$ are defined to be the *rank* and *dimension* of the lattice.

Let us consider an example. One most intuitive example of a lattice would be the set of all points with integer coordinates $\mathbb{Z}^n$:

$$\mathbb{Z}^n = \{ x = (x_1, \ldots, x_n) : x_i \in \mathbb{Z} \text{ for all } 1 \leq i \leq n \}. $$

And a basis for this lattice can also be provided, which could be the set of standard basis vectors $e_1, \ldots, e_n$, where

$$e_i = \left( 0, \ 0, \ldots, \ 1, \ 0, \ldots, 0 \right)^T$$

with 1 in the $i$-th position and 0 elsewhere. However, this is not the only basis for the integer lattice. We can easily find another basis satisfying the condition, which is the set of all vectors

$$e_i + e_{i+1}, \text{ for } 1 \leq i \leq n, \text{ and } e_1 + e_n$$

Applying matrix notation, the lattice $\Lambda$ can be rewritten in the form of matrix-vector multiplication:

$$y \in \Lambda, \text{ and } y = \sum_{i=1}^{r} x_i a_i = A x, \text{ where } x = (x_1, \ldots, x_r)^T \in \mathbb{Z}^r,$$

and $A = [a_1, \ldots, a_n] \in \mathbb{R}^{n \times r}$ is a basis matrix with rank $r$.

Here we have built up the definition and notation for the lattice. Next we introduce the definition of sublattices of a lattice.

**DEFINITION 3.1.2.** If $\Lambda$ and $\Omega$ are both lattices in $\mathbb{R}^n$, and $\Omega \subseteq \Lambda$, then we say that $\Omega$ is a sublattice of $\Lambda$.

**LEMMA 3.1.3.** If $\Lambda$ and $\Omega$ are both lattices in $\mathbb{R}^n$, let $\Omega$ be a sublattice of $\Lambda$. There exists a positive integer $D$ such that $D \Lambda \subseteq \Omega$. 
Proof. As stated in Definition 3.1.2, $\Lambda$ and $\Omega$ are both lattices of rank $n$ in $\mathbb{R}^n$. Now assume $a_1, \ldots, a_n$ be a basis for $\Omega$ and $b_1, \ldots, b_n$ be a basis for $\Lambda$. Then by Definition 3.1.1

$$\text{span}\{a_1, \ldots, a_n\} = \text{span}\{b_1, \ldots, b_n\} = \mathbb{R}^n.$$ 

Since $\Omega \subseteq \Lambda$, there exist integers $u_{11}, \ldots, u_{nn}$ such that

\begin{align*}
a_1 & = u_{11}b_1 + \cdots + u_{1n}b_n \\
& \vdots \\
a_n & = u_{n1}b_1 + \cdots + u_{nn}b_n.
\end{align*}

By means of the elimination method, we can try to solve this linear system for $b_1, \ldots, b_n$ in terms of $a_1, \ldots, a_n$. After rearranging the variables, we find that there must exist rational numbers $\frac{p_{ij}}{q_{ij}}, \ldots, \frac{p_{nn}}{q_{nn}}$ for each $p_{ij}, q_{ij} \in \mathbb{Z}$, such that

\begin{align*}
b_1 & = \frac{p_{11}}{q_{11}}a_1 + \cdots + \frac{p_{1n}}{q_{1n}}a_n \\
& \vdots \\
b_n & = \frac{p_{n1}}{q_{n1}}a_1 + \cdots + \frac{p_{nn}}{q_{nn}}a_n.
\end{align*}

Let $D$ be $q_{11} \times \cdots \times q_{nn}$, and the linear equations can be transformed into a new system

\begin{align*}
Db_1 &= \frac{Dp_{11}}{q_{11}}a_1 + \cdots + \frac{Dp_{1n}}{q_{1n}}a_n \\
& \vdots \\
Db_n &= \frac{Dp_{n1}}{q_{n1}}a_1 + \cdots + \frac{Dp_{nn}}{q_{nn}}a_n
\end{align*}

Since $D/q_{ij} \in \mathbb{Z}$ for each $1 \leq i, j \leq n$, all the vectors in the system are in $\Omega$. Therefore $\text{span}_\mathbb{Z}\{Db_1, \ldots, Db_n\} \subseteq \Omega$. Thus,

$$\text{span}_\mathbb{Z}\{Db_1, \ldots, Db_n\} = D\text{span}_\mathbb{Z}\{b_1, \ldots, b_n\} = D\Lambda,$$

which completes the proof. \qed
It is noticeable that $D$ in this case is also the quotient of the determinant of $\Omega$ divided by the determinant of $\Lambda$. We will give a formal definition of the determinant of lattices. To do that, we need to introduce the definition of the fundamental parallelootope.

**DEFINITION 3.1.4.** Let $\Lambda$ be a lattice, and $a_1, \ldots, a_n$ be a basis for $\Lambda$. Then the set

$$\mathcal{F} = \left\{ \sum_{i=1}^{n} t_i a_i : 0 \leq t_i < 1, \forall 1 \leq i \leq n \right\},$$

is called a fundamental parallelootope of $\Lambda$ with respect to the basis $a_1, \ldots, a_n$.

**DEFINITION 3.1.5.** The determinant of a lattice $\Lambda$, denoted $\det(\Lambda)$, is the $n$-dimensional volume of the fundamental parallelootope $\mathcal{F}$ spanned by the basis vectors.

The determinant is one of the most important concepts for our later analysis. We will illustrate why in the proof of the following theorem.

**THEOREM 3.1.6.** Let $\Lambda$ be a lattice of rank $n$ in $\mathbb{R}^n$, and let $A$ be a basis matrix for $\Lambda$. Then $B$ is another basis matrix for $\Lambda$ if and only if there exists an $n \times n$ integral matrix $U$ with determinant $\pm 1$ such that

$$B = AU.$$

*Proof.* First suppose that $B$ is a basis matrix. Notice that, since $A$ is a basis matrix, for every $1 \leq i \leq n$ the $i$-th column vector $b_i$ of $B$ can be expressed as

$$b_i = \sum_{j=1}^{n} u_{ij} a_j,$$

where $a_1, \ldots, a_n$ are column vectors of $A$, and $u_{i1}, \ldots, u_{ij}$ are integers for all $1 \leq j \leq n$. Thus $B = AU$ can be obtained, where $U = (u_{ij})_{1 \leq i,j \leq n}$ is an $n \times n$ matrix with integer entries. In addition, since $B$ is also a basis matrix, we have that, for every $1 \leq i \leq n$,

$$a_i = \sum_{j=1}^{n} w_{ij} b_j,$$

where $w_{i1}, \ldots, w_{ij}$ are also integers for all $1 \leq j \leq N$. Hence $A = BW$, where $W = (w_{ij})_{1 \leq i,j \leq n}$ is also an $n \times n$ matrix with integer entries. Then we combine the results

$$B = AU = BWU,$$

which indicates that $WU = I_n$, the $n \times n$ identity matrix. Therefore

$$\det(WU) = \det(W) \det(U) = \det(I_n) = 1,$$
but det$(U)$, det$(W) \in \mathbb{Z}$ due to $U$ and $W$ are integral matrices. This means that

$$\det(U) = \det(W) = \pm 1.$$ 

Next we prove the other direction of the theorem. Assume that $B = UA$ for some integral $n \times n$ matrix $U$ with det$(U) = \pm 1$. This means that det$(B) = \pm \det(A) \neq 0$, hence column vectors of $B$ are linearly independent. Also, $U$ is invertible over $\mathbb{Z}$, showing that $U^{-1} = (u_{ij})_{1 \leq i,j \leq n}$ is also an integral matrix, hence $A = U^{-1}B$. This means that column vectors of $A$ are in the span of the column vectors of $B$, and so

$$\Lambda \subseteq \text{span}\{b_1, \ldots, b_n\}.$$ 

On the other hand, $b_i \in \Lambda$ for each $1 \leq i \leq n$. As a result, $B$ is a basis matrix for $\Lambda$. \hfill \square

By this theorem, we show that the determinant of the lattice is a invariant, which means it does not depend on any particular basis of the lattice. And we can easily acquire,

**COROLLARY 3.1.7.** If $A$ and $B$ are two basis matrices for the same lattice $\Lambda$, then

$$|\det(A)| = |\det(B)|.$$

Before we enter into more fun parts, I want to introduce another important definition about the lattice.

**DEFINITION 3.1.8.** The Voronoi cell of a lattice $\Lambda$ is the set

$$V(\Lambda) = \{x \in \mathbb{R}^n : \|x\| \leq \|x - y\| \quad \forall y \in \Lambda\}.$$

### 2.2 Successive minima

Given a lattice $\Lambda$ in $\mathbb{R}^n$, consider a unit ball centered at the origin denoted by $B_n$. An interesting operation involves scaling $B_n$ such that it encompasses one nonzero point of the lattice $\Lambda$. This scaling factor is denoted by $\lambda_1$. Naturally, one may inquire further: how much should $B_n$ be expanded to contain two linearly independent points of $\Lambda$? Or three linearly independent points? And so on. This section aims to address such inquiries.

First, let us have a formal definition of this observation. Let $B$ be a compact convex set in $\mathbb{R}^n$, which states that a set is a closed and bounded subset of a Euclidean space or more generally, a topological vector space, that contains all line segments connecting any pair of its points.
DEFINITION 3.2.1. For any $1 \leq i \leq n$, the successive minima

$$\lambda_i(L) = \min \{ \lambda \in \mathbb{R}_{>0} : \dim \text{span}_R(\lambda B \cap L) \geq i \}.$$ 

By definition of the above, we define the $i$-th successive minimum of $M$ with respect to $\Lambda$, with the definition of an open ball here, let $B_n(0, r) = \{ x \in \mathbb{R}^n : \| x \| < r \}$ be the $n$-dimensional open ball of radius $r$ centered at $0$. We usually refer to as the $i$-th successive minimum of $\Lambda$.

The $i$th successive minimum $\lambda_i(L)$ is the radius of the smallest sphere centered in the origin containing $i$ linearly independent vectors of the lattice. And it is straightforward to observe that its successive minima follows that $0 < \lambda_1(L) \leq \lambda_2(L) \leq \ldots \leq \lambda_n(L)$.

Given the established lower bound $\lambda_1(L) > 0$, a natural inquiry pertains to the existence of an upper bound for $\lambda_1$. Prior to addressing this, however, a fundamental question arises: How can one ascertain the presence of lattice points $L$ within $B$? I will introduce the theorem of Blichfeldt here.

THEOREM 3.2.2. [Blichfeldt, 1914] [Fuk22] Let $\Lambda$ be a lattice of full rank $\mathbb{Z}^n$ and a compact convex set $B$ in $\mathbb{R}^n$. Suppose that $\text{Vol}(B) \geq 1$. Then there exist $x, y \in B$ such that $0 \neq x - y \in \mathbb{Z}^n$.

Proof. Assume that $\text{Vol}(B) > 1$. Define

$$P = \{ x \in \mathbb{R}^n : 0 < x_i < 1, \forall 1 \leq i \leq n \}$$

and consider

$$S = \{ u \in \mathbb{Z}^n : B \cap (P + u) \neq \emptyset \}.$$
As defined above, \( B \) is bounded, \( S \) is a finite set, namely \( S = \{u_1, \ldots, u_{r_0}\} \). For each \( 1 \leq r \leq r_0 \), define \( B_r = M \cap (P + u_r) \) and \( B'_r = B_r - u_r \). Hence, \( B'_1, \ldots, B'_r \subseteq P \). Furthermore, \( \bigcup_{r=1}^{r_0} B_r = B \) and \( B_r \cap B_s = \emptyset \) for all \( 1 \leq r < s \leq r_0 \). We know that \( B \subseteq P + u_r, B \subseteq P + u_s, \) and \( (P + u_r) \cap (P + u_s) = \emptyset \). This implies

\[
1 < \text{Vol}(B) = \sum_{r=1}^{r_0} \text{Vol}(B_r).
\]

Nonetheless, for each \( 1 \leq r \leq r_0 \), \( \text{Vol}(B'_r) = \text{Vol}(B_r) \), and

\[
\sum_{r=1}^{r_0} \text{Vol}(B'_r) > 1,
\]

yet

\[
\bigcup_{r=1}^{r_0} B'_r \subseteq P,
\]

resulting in

\[
\text{Vol}\left(\bigcup_{r=1}^{r_0} B'_r\right) \leq \text{Vol}(P) = 1.
\]

The sets \( B'_1, \ldots, B'_r \) are not pairwise disjoint, which implies that there exist indices \( 1 \leq r < s \leq r_0 \) such that \( B'_r \cap B'_s \neq \emptyset \). Therefore, there exists an element \( x \in B'_r \cap B'_s \), which leads us to \( x + u_r, x + u_s \in B \), and

\[
(x + u_r) - (x + u_s) = u_r - u_s \in \mathbb{Z}^n.
\]

Now, consider \( B \) to be bounded, and with \( \text{Vol}(B) = 1 \). Let \( \{s_r\}_{r=1}^\infty \) be a sequence of numbers greater than 1 converging to 1, such that

\[
\lim_{r \to \infty} s_r = 1.
\]

From the preceding argument, we deduce that for each \( r \), there exist distinct

\[
x_r \neq y_r \in s_rB
\]

such that \( x_r - y_r \in \mathbb{Z}^n \). Consequently, there are subsequences \( \{x_{r_k}\} \) and \( \{y_{r_k}\} \) converging to points \( x, y \in B \). Since for each \( r_k \), \( x_{r_k} - y_{r_k} \) is a nonzero lattice point, it must be true that \( x \neq y \) and \( x - y \in \mathbb{Z}^n \). \( \square \)

As a corollary to Blichfeldt theorem we can immediately prove the following theorem of Minkowski.

However, it is imperative to establish a lemma concerning volumes as a prerequisite to the proof of Minkowski’s theorem.

**Lemma 3.2.3.** Let \( S \) be a compact convex set in \( \mathbb{R}^n \), \( A \in GL_n(\mathbb{R}) \), and define \( T = AS = \{Ax :
$x \in S$, then $\text{Vol}(T(S)) = \det(A) \cdot \text{Vol}(S)$.

**Proof.** As defined above, a linear transformation $A$ and a set $S$. The volume transformation under $A$ can be expressed as follows:

\[
\text{Vol}(T(S)) = \text{Vol}(AS) = \int_{AS} dx = \int_{S} |\det(A)| \, dy \quad \text{(change of variables $x = Ay$, $dx$ equals to jacobian determinant $|\det(A)| \cdot dy$)}
\]

\[
= |\det(A)| \int_{S} dy = |\det(A)| \text{Vol}(S).
\]

\[ \square \]

**THEOREM 3.2.4.** [Minkowski Convex Body Theorem] [Fuk22]

Let $\Lambda$ be a lattice of full rank in $\mathbb{Z}^n$ and $B$ be a compact convex set in $\mathbb{R}^n$ symmetric about the origin with $\text{Vol}(B) \geq 2^n$, Then there exists $0 \neq x \in B \cap \mathbb{Z}^n$.

**Proof.** Observe that the set

\[
\frac{1}{2}B = \left\{ \frac{1}{2}x : x \in B \right\} = \left\{ \begin{pmatrix} \frac{1}{2} & 0 & \cdots & 0 \\ 0 & \frac{1}{2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{2} \end{pmatrix} B \right\}
\]

is also convex and symmetric about the origin, and by Lemma 3.2.3 its volume is

\[
\text{Vol} \left( \frac{1}{2}B \right) = \det \begin{pmatrix} \frac{1}{2} & 0 & \cdots & 0 \\ 0 & \frac{1}{2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{2} \end{pmatrix} \text{Vol}(B) = 2^{-n} \text{Vol}(B) \geq 1.
\]

Therefore, by Theorem 3.2.2, there exist $\frac{1}{2}x \neq \frac{1}{2}y$ for $\frac{1}{2}x, \frac{1}{2}y \in \frac{1}{2}B$ such that

\[
\frac{1}{2}x - \frac{1}{2}y \in \mathbb{Z}^n.
\]
However, by symmetry, since \( y \in B \), \( -y \in B \), and by convexity, since \( x, -y \in B \),

\[
\frac{1}{2}x - \frac{1}{2}(-y) = \frac{1}{2}x + \frac{1}{2}y \in B.
\]

Based on the Convex Body Theorem, we can deduce the following corollary.

**COROLLARY 3.2.5.** [Fuk22] Let \( B \) be a compact convex set in \( \mathbb{R}^n \) symmetric about the origin, \( \Lambda \) be a lattice of full rank \( \mathbb{Z}^n \) and \( \lambda_1 \) to be the first successive minima of \( B \) with respect to \( \Lambda \). Then

\[
0 < \lambda_1 \leq \left( \frac{\det(\Lambda)}{\text{Vol}(B)} \right)^{1/n}.
\]

*Proof.* As discussed above, \( \lambda_1 \) is positive because of \( \Lambda \) being a discrete set. Therefore, only the upper bound needs to be proven. Combined Lemma 3.2.3 and Theorem 3.2.4, we know that for a lattice \( \Lambda \), if

\[
\text{Vol}(\lambda B) \geq 2^n \det(\Lambda),
\]

then \( \lambda B \) contains a nonzero point of \( \Lambda \). On the other hand, as we shown in Lemma 3.2.3, we can deduce that

\[
\text{Vol}(\lambda B) = \lambda^n \text{Vol}(b).
\]

Hence, if the condition satisfies

\[
\lambda^n \text{Vol}(M) \geq 2^n \det(\Lambda),
\]

the expanded set \( \lambda B \) is guaranteed to contain a non-zero point of \( \Lambda \). The proves the corollary.

By this corollary, the first successive minima \( \lambda_1 \) is bounded on both sides. Furthermore, Minkowski also gave a stronger proof to produce bounds on all the successive minima, known as *Minkowski’s Successive Minima Theorem*. The three proofs presented are constructed by closely adhering to the methodology delineated in the expository notes on discrete optimization authored by Lenny Fukshansky. Please see the bibliography for further details. In this section, the proofs are presented specifically for \( \mathbb{Z}^n \), showcasing fundamental lattice properties. To extend these results to an arbitrary lattice, one simply needs to apply similar methodologies while adjusting for the lattice’s basis and metric considerations.

**THEOREM 3.2.6.** [Minkowski’s Successive Minima Theorem] [Fuk22] Let \( B \subseteq \mathbb{R}^n \) be a compact convex set symmetric about the origin, \( \Lambda \subseteq \mathbb{R}^n \) to be a lattice of full rank and \( \lambda_1 \ldots \lambda_n \) to be the
corresponding successive minima of $B$ with respect to $\Lambda$. Then,

$$\frac{2^n \det(\Lambda)}{n! \text{Vol}(M)} \leq \lambda_1 \ldots \lambda_n \leq \frac{2^n \det(\Lambda)}{\text{Vol}(M)}$$

**Proof.** This proof starts with a weaker version of $\Lambda = \mathbb{Z}^n$. I will talk about how to extend this argument to arbitrary lattices in the end. First, we prove the lower bound part following [GL87]. Let $u_1, \ldots, u_n$ be the $n$ linearly independent vectors that correspond to the successive minima $\lambda_1, \ldots, \lambda_n$. We then define the matrix $U$ as

$$U = (u_1 \ldots u_n) = \begin{pmatrix} u_{11} & \ldots & u_{n1} \\ \vdots & \ddots & \vdots \\ u_{1n} & \ldots & u_{nn} \end{pmatrix}$$

Then $U = U\mathbb{Z}^n$ constitutes a full-rank sublattice of $\mathbb{Z}^n$. It can be noted that there are $2^n$ points

$$\pm \frac{u_1}{\lambda_1}, \ldots, \pm \frac{u_n}{\lambda_n}$$

are contained within $B$. And $B$ encompasses the convex hull $P$ of these points, which is defined to be the smallest convex set containing these points. The convex hull forms a generalized octahedron.

Think about a polyhedron in $\mathbb{R}^n$, which can be decomposed into simplices that intersect pairwise only at their boundaries. A *standard k-simplex* in $\mathbb{R}^n$ is a $k$-dimensional polytope that is the convex hull of $k + 1$ points, satisfying no $k$ of which lie in a $(k - 1)$-dimensional subspace of $\mathbb{R}^n$. Thus the convex hull does not encompass any integer lattice points in its interior.

As referenced, the volume of a standard simplex in $\mathbb{R}^n$ is $\frac{1}{n!}$. Our generalized octahedron $P$ can be decomposed into $2^n$ simplices, which are obtained from the standard simplex by multiplication by the matrix

$$\begin{pmatrix} \frac{u_{11}}{\lambda_1} & \ldots & \frac{u_{n1}}{\lambda_n} \\ \vdots & \ddots & \vdots \\ \frac{u_{1n}}{\lambda_1} & \ldots & \frac{u_{nn}}{\lambda_n} \end{pmatrix},$$

therefore its volume is

$$\text{Vol}(P) = \frac{2^n}{n!} \det \begin{pmatrix} \frac{u_{11}}{\lambda_1} & \ldots & \frac{u_{n1}}{\lambda_n} \\ \vdots & \ddots & \vdots \\ \frac{u_{1n}}{\lambda_1} & \ldots & \frac{u_{nn}}{\lambda_n} \end{pmatrix} = \frac{2^n \det(U)}{n! \lambda_1 \cdots \lambda_n} \quad (1)$$

As we know $\det(U)$ is an integer, since $P \subseteq B$, $\text{Vol}(P) \leq \text{Vol}(B)$. Combining this with (1) yields the lower bound of the theorem.

We then advance to establishing the upper bound. The approach employed is largely based on
M. Henk’s framework [Hen02], drawing upon foundational geometric principles originally proposed by Minkowski.

For each \(1 \leq i \leq n\), let \(E_i\) be the space spanned by the first \(i\) standard basis vectors of \(\mathbb{R}^n\), i.e.,

\[ E_i = \text{span}_\mathbb{R}\{e_1, \ldots, e_i\}, \]

and define

\[ B_i = \frac{\lambda_i}{2} B. \]

Similarly, we consider \(u_1, \ldots, u_n\) to be the \(n\) linearly independent vectors corresponding to the successive minima \(\lambda_1, \ldots, \lambda_n\). Observe that there exists a matrix \(A \in GL_n(\mathbb{Z})\) such that

\[ A\text{span}_\mathbb{R}\{u_1, \ldots, u_i\} \subseteq E_i, \]

for each \(1 \leq i \leq n\), implying that we can rotate each \(\text{span}_\mathbb{R}\{u_1, \ldots, u_i\}\) to make it within \(E_i\). Furthermore, the volume of \(AB\) is invariant to that of \(B\), since \(\det(A) = 1\) (rotations preserve volumes), and

\[ Au_i \in \lambda_i' AB \cap E_i, \quad \forall 1 \leq i \leq n, \]

where \(\lambda_1', \ldots, \lambda_n'\) denote the successive minima of \(AB\) with respect to \(\mathbb{Z}^n\). Consequently, it is permissible to assume, without loss of generality, that

\[ \text{span}_\mathbb{R}\{u_1, \ldots, u_i\} \subseteq E_i, \]

for each \(1 \leq i \leq n\).

For an integer \(q \in \mathbb{Z}_{\geq 0}\), define the integral cube of sidelength \(2q\) centered at 0 in \(\mathbb{R}^n\) as

\[ C_q^n = \{z \in \mathbb{Z}^n : |z| \leq q\}, \]

and for each \(1 \leq i \leq n\) define the section of \(C_q^n\) by \(E_i\) as

\[ C_q^i = C_q^n \cap E_i. \]

Notice that \(C_q^n\) is contained in a real cube of volume \((2q)^n\), and so the volume of all translates of \(b\) by the points of \(C_q^n\) can be bounded as

\[ \text{Vol}(C_q^n + B_n) \leq (2q + \gamma)^n, \quad (2) \]
where $\gamma$ is a constant that depends on $B$ only. Also note that if $x \neq y \in \mathbb{Z}^n$, then

$$\text{int}(x + B_1) \cap \text{int}(y + B_1) = \emptyset,$$

where int stands for the interior of a set: suppose not, then there exists

$$z \in \text{int}(x + B_1) \cap \text{int}(y + B_1),$$

and so

$$(z - x) - (z - y) = y - x \in \text{int}(B_1) - \text{int}(B_1)$$

$$= \{z_1 - z_2 : z_1, z_2 \in B_1\} = \text{int}(\lambda_1 B),$$

which would contradict the minimality of $\lambda_1$. Therefore

$$\text{Vol}(C_q^n + B_1) \leq (2q + 1)^n \text{Vol}(B_1) = (2q + 1)^n \left(\frac{\lambda_1}{2}\right)^n \text{Vol}(B).$$

(4)

Here we need another lemma.

**Lemma 3.2.7.** For each $1 \leq i \leq n - 1$,

$$\text{Vol}(C_q^n + B_{i+1}) \geq \left(\frac{\lambda_{i+1}}{\lambda_i}\right)^{n-i} \text{Vol}(C_q^n + B_i).$$

(5)

**Proof.** If $\lambda_{i+1} = \lambda_i$, the statement is obvious, so we assume $\lambda_{i+1} > \lambda_i$. Let $x, y \in \mathbb{Z}^n$ be such that

$$(x_{i+1}, \ldots, x_n) \neq (y_{i+1}, \ldots, y_n).$$

Then

$$(x + \text{int}(B_{i+1})) \cap (y + \text{int}(B_{i+1})) = \emptyset.$$ 

(6)

Indeed, suppose the above statement is not true, i.e., there exists $z \in (x + \text{int}(B_{i+1})) \cap (y + \text{int}(M_{i+1}))$. Then, as in (3) stated above, $x - y \in \text{int}(\lambda_{i+1} B)$. But we also have

$$u_1, \ldots, u_i \in \text{int}(\lambda_{i+1} B),$$

since $\lambda_{i+1} > \lambda_i$ and so $\lambda_i B \subseteq \text{int}(\lambda_{i+1} B)$. Moreover, $u_1, \ldots, u_i \in E_i$, meaning that

$$u_{jk} = 0 \forall 1 \leq j \leq i, \ i + 1 \leq k \leq n.$$
On the other hand, at least one of

$$x_k - y_k, \ i + 1 \leq k \leq n,$$

is not equal to 0. Hence $x - y, u_1, \ldots, u_i$ are linearly independent, but this means that $\text{int}(\lambda_{i+1}B)_{q}$ contains $i + 1$ linearly independent points, which contradicts the minimality of $\lambda_{i+1}$. This proves (6).

Notice that (6) implies

$$\text{Vol}(C_q^n + B_i + 1) = (2q + 1)^{n-i} \text{Vol}(C_q^n + B_i + 1),$$

and

$$\text{Vol}(C_q^n + B_i) = (2q + 1)^{n-i} \text{Vol}(C_q^n + B_i),$$

since $B_i \subseteq B_{i+1}$. Hence, in order to prove the lemma we need to prove that

$$\text{Vol}(C_q^n + M_{i+1}) \geq \left( \frac{\lambda_{i+1}}{\lambda_i} \right)^{n-i} \text{Vol}(C_q^n + M_i).$$

Define two linear maps $f_1, f_2 : \mathbb{R}^n \to \mathbb{R}^n$, given by

$$f_1(x) = \left( \frac{\lambda_{i+1}}{\lambda_i} x_1, \ldots, \frac{\lambda_{i+1}}{\lambda_i} x_i, x_{i+1}, \ldots, x_n \right),$$

and

$$f_2(x) = \left( x_1, \ldots, x_i, \frac{\lambda_{i+1}}{\lambda_i} x_{i+1}, \ldots, \frac{\lambda_{i+1}}{\lambda_i} x_n \right),$$

and notice that $f_2(f_1(B_i)) = B_{i+1}, f_2(C_q^n) = C_q^n$. Therefore

$$f_2(C_q^n + f_1(B_i)) = C_q^n + B_{i+1}.$$  

This implies that

$$\text{Vol}(C_q^n + B_i + 1) = \left( \frac{\lambda_{i+1}}{\lambda_i} \right)^{n-i} \text{Vol}(C_q^n + f_1(B_i)),$$

and so to establish (9) it is sufficient to show that

$$\text{Vol}(C_q^n + f_1(B_i)) \geq \text{Vol}(C_q^n + B_i).$$

Let $E_i^\perp = \text{span}_\mathbb{R}\{e_{i+1}, \ldots, e_n\}$, i.e., $E_i^\perp$ is the orthogonal complement of $E_i$ with dimension $n - i$.

Notice that for every $x \in E_i^\perp$ there exists $t(x) \in E_i$ such that

$$B_i \cap (x + E_i) \subseteq (f_1(B_i) \cap (x + E_i)) + t(x),$$

and so to establish (9) it is sufficient to show that

$$\text{Vol}(C_q^n + f_1(B_i)) \geq \text{Vol}(C_q^n + B_i).$$

Let $E_i^\perp = \text{span}_\mathbb{R}\{e_{i+1}, \ldots, e_n\}$, i.e., $E_i^\perp$ is the orthogonal complement of $E_i$ with dimension $n - i$. Notice that for every $x \in E_i^\perp$ there exists $t(x) \in E_i$ such that

$$B_i \cap (x + E_i) \subseteq (f_1(B_i) \cap (x + E_i)) + t(x),$$

and so to establish (9) it is sufficient to show that

$$\text{Vol}(C_q^n + f_1(B_i)) \geq \text{Vol}(C_q^n + B_i).$$

Let $E_i^\perp = \text{span}_\mathbb{R}\{e_{i+1}, \ldots, e_n\}$, i.e., $E_i^\perp$ is the orthogonal complement of $E_i$ with dimension $n - i$.
in other words, although it is not necessarily true that $B_i \subseteq f_1(B_i)$, each section of $B_i$ by a translation of $E_i$ is contained in a translate of some such section of $f_1(B_i)$. Therefore

$$(C_q^i + B_i) \cap (x + E_i) \subseteq (C_q^i + f_1(B_i)) \cap (x + E_i) + t(x),$$

and hence

$$\text{Vol}(C_q^i + B_i) = \int_{x \in E_i^+} \text{Vol}_i((C_q^i + B_i) \cap (x + E_i)) \, dx$$

$$\leq \int_{x \in E_i^+} \text{Vol}_i((C_q^i + f_1(B_i)) \cap (x + E_i)) \, dx$$

$$= \text{Vol}(C_q^i + f_1(B_i)),$$

where $\text{Vol}_i$ stands for the $i$-dimensional volume. This completes the proof of the statement $\text{Vol}(C_q^i + f_1(B_i)) \geq \text{Vol}(C_q^i + B_i)$, and hence of the lemma.

Now, combining (2), (4), and (5), we obtain:

$$(2q + \gamma)^n \geq \text{Vol}(C_q^n + B_n) \geq \left(\frac{\lambda_n}{\lambda_{n-1}}\right) \text{Vol}(C_q^n + B_{n-1}) \geq \cdots \geq \left(\frac{\lambda_n}{\lambda_{n-1}}\right)^2 \cdots \left(\frac{\lambda_2}{\lambda_1}\right)^{n-1} \text{Vol}(C_q^n + B_1)$$

$$= \lambda_n \cdots \lambda_1 \frac{\text{Vol}B}{2^n} (2q + 1)^n,$$

hence

$$\lambda_1 \cdots \lambda_n \leq \frac{2^n}{\text{Vol}(B)} \left(\frac{2q + \gamma}{2q + 1}\right)^n \to \frac{2^n}{\text{Vol}(B)},$$

as $q \to \infty$, since $q \in \mathbb{Z}_{\geq 0}$ is arbitrary. We finish the proof in the case $\Lambda = \mathbb{Z}^n$.

Finally, I will extend the argument to arbitrary lattices. Let $L$ be an arbitrary lattice such that $L = AZ^n$, where the full-rank transformation matrix $A$ is given by $A = (a_1, a_2, \ldots, a_n)$. Suppose $\lambda_1, \ldots, \lambda_n$ are the corresponding successive minima of $\mathbb{Z}^n$ with respect to a compact convex set $K_1$, and let $x_1, \ldots, x_n$ be the corresponding shortest vectors. Then $Ax_1, \ldots, Ax_n$ will be the vectors corresponding to the successive minima of $L$ with respect to the convex body $K = AK_1$.

Assume the successive minima of $L$ with respect to $K$ to be $\mu_1, \ldots, \mu_n$. By definition,

$$\mu_i(L) = \min\{\mu \in \mathbb{R}_{>0} : \dim \text{span}_\mathbb{R}(\mu K \cap L) \geq i\}.$$
that is,

\[ \mu_i = \min\{\mu : \dim(\mu K \cap L) \geq i\} \]
\[ = \min\{\mu : \dim(\mu AK_1 \cap AZ^n) \geq i\} \]
\[ = \min\{\mu : \dim(A(\mu K_1 \cap Z^n)) \geq i\}. \]

The transformation matrix, being of full rank, maintains the dimensional properties; a lattice transformed by such a matrix retains its number of linearly independent vectors. Specifically, if we originally have \( n \) linearly independent vectors constituting the dimensionality of the space, applying the full-rank transformation does not alter this dimensionality. Hence, we have that \( \dim(\mu K_1 \cap Z^n) \geq i \iff \dim(A(\mu K_1 \cap Z^n)) \geq i \). Therefore, \( \mu_i = \lambda_i \). This allows us to extend the concept of successive minima from the standard lattice \( \mathbb{Z}^n \) to arbitrary lattices \( L \). The bound for the corresponding successive minima also holds true for arbitrary lattices. This completes the proof.

\[ \square \]

2.3 Computational lattice problems

In the concluding part of this chapter, after having established a foundational understanding of lattice structures, it is pertinent to revisit three venerable problems that share a profound connection with the techniques employed in the geometry of numbers, which are sphere packing, sphere covering, and kissing number. These historical problems are not only central to the field of discrete geometry but also provide critical insights into the optimization and structural properties of lattices. The techniques developed in the study of these problems have far-reaching applications, extending beyond pure mathematics to inform solutions in physics, materials science, and information theory. In addition, these three problems have deep connection with the NP-hard problems I will cover in Chapter 3. The following theorems refer to the discussion on this subject in the well-known book by Conway and Sloane [CS99].

In this section, we set up two definitions first. We will define a sphere in \( \mathbb{R}^n \) \( (n \geq 2) \) to be a closed ball whose boundary is this sphere. Moreover, we will define a collection of spheres \( \{B_i\} \) of radius \( r \) to be packed in \( \mathbb{R}^n \) if

\[ \text{int}(B_i) \cap \text{int}(B_j) = \emptyset, \quad \forall i \neq j, \]

and there exist indices \( i \neq j \) such that

\[ \text{int}(B_i') \cap \text{int}(B_j') \neq \emptyset, \]

whenever \( B_i' \) and \( B_j' \) are spheres of radius larger than \( r \) such that \( B_i \subset B_i' \), \( B_j \subset B_j' \).

The \( n \)-dimensional sphere packing problem aims to find how densely identical spheres can be
packed in $\mathbb{R}^n$, inherently relates to the arrangement of sphere centers. The essence of the problem narrows to devising a strategy that positions the centers of spheres so as to maximize the density of the packing. An intriguing approach to this is the utilization of lattice structures. Specifically, one can consider positioning the sphere centers at the points of a lattice $\Lambda$ of full rank in $\mathbb{R}^n$. Packings formed in this manner are referred to as lattice packings.

**DEFINITION 3.3.1.** Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice of full rank. The *density* of the corresponding sphere packing is defined to be

$$\Delta(\Lambda) := \frac{\text{the proportion of space occupied by spheres}}{\text{volume of a sphere}} = \frac{\text{volume of a fundamental domain of } \Lambda}{\omega_n} = \frac{r^n \omega_n}{\det(\Lambda)},$$

where $r$ is the *packing radius*, i.e., the radius of each sphere in this lattice packing, and $\omega_n$ is the volume of a unit ball in $\mathbb{R}^n$, given by

$$\omega_n = \begin{cases} \frac{\pi^k}{k!} & \text{if } n = 2k \text{ for some } k \in \mathbb{Z}, \\ \frac{2^{2k+1}k!2^k}{(2k+1)!} & \text{if } n = 2k + 1 \text{ for some } k \in \mathbb{Z}. \end{cases}$$

Hence the volume of a ball of radius $r$ in $\mathbb{R}^n$ is $\omega_n r^n$. It is apparent that the packing radius $r$ coincides with the radius of the largest sphere that can be inscribed within the Voronoi cell $V$ of $\Lambda$, commonly referred to as the *inradius* of $V$. Evidently, the inequality $\Delta \leq 1$ holds true.

An immediate inference can be drawn regarding the packing radius $r$; it is intrinsically linked to the structure of the lattice. More precisely, $r$ equates to exactly one-half the length of the shortest non-zero lattice vector, thus $r = \frac{\lambda_1}{2}$, with $\lambda_1$ being the first successive minimum of $\Lambda$. This leads us to the formula:

$$\Delta = \frac{\lambda_1 r^n \omega_n}{2^n \det(\Lambda)}.$$

The question of whether the densest packings in each dimension are exclusively lattices remains open. Nonetheless, we have the classical theorem originally posited by Minkowski in 1905 and further refined by Hlawka in 1944, which is widely recognized as the Minkowski-Hlawka theorem.

**THEOREM 3.3.2.** In every dimension $n$, there exist lattice packings whose density

$$\Delta \geq \frac{\zeta(n)}{2^{n-1}},$$

where $\zeta(s)$ represents the Riemann zeta-function, defined as $\zeta(s) = \sum_{k=1}^{\infty} \frac{1}{k^s}$.
The proofs of Theorem 3.3.2 in literature are all nonconstructive, leaving the specific construction techniques for achieving such high-density lattice packings as something of a mystery. Further details on this prominent theorem can be found in [GL87] and [Cas59].

Then we move to the second problem: sphere covering. The essence of this problem lies in blanketing $\mathbb{R}^n$ with spheres to achieve the minimum possible overlap, striving for the slimmest feasible covering. Our attention is specifically drawn to lattice coverings, which are coverings where spheres are centered at lattice points.

**DEFINITION 3.3.3.** Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice of full rank. The

$$
\Theta(\Lambda) = \frac{\text{average number of spheres containing a point of the space}}{	ext{volume of one sphere}}
= \frac{\text{volume of a fundamental domain of } \Lambda}{R^n \omega_n}
= \frac{\text{R}^n \omega_n}{\det(\Lambda)}.
$$

where $\omega_n$ is the volume of a unit ball in $\mathbb{R}^n$, and $R$ is the covering radius, i.e., the radius of each sphere in this lattice covering. It is apparent that $R$ is the radius of the tiniest ball that can be wrapped around the Voronoi cell $V$ of $\Lambda$, known as the circumradius of $V$. It follows that $\Theta \geq 1$.

The optimal sphere covering is known exclusively in dimension $n = 2$. The finest lattice coverings are currently recognized only in dimensions $n \leq 5$, and it is yet to be determined whether optimal coverings in every dimension are necessarily lattice-based.

We can observe that the packing and covering characteristics of a lattice $\Lambda$ hinge critically on its Voronoi cell $V$. For the optimal packing and covering of $\Lambda$, it is crucial to maximize the inradius $r$ of $V$ while simultaneously minimizing the circumradius $R$. This optimization criterion can be expressed as:

$$
\lambda_1 = \ldots = \lambda_n.
$$

Finally, the third problem, kissing number problem describes: in $\mathbb{R}^n$, for a given sphere, how many other non-overlapping spheres of the same radius can touch it? Specifically, consider a sphere centered at the origin, and the challenge is to find out the maximum number of identical spheres that can all touch the original without intersecting. The solution in two dimensions is straightforward – six spheres can touch the central one. However, the kissing number, like the other two problems, is only known for certain dimensions.

In general, this section illustrates that there seems to be a lack of efficient way to find solutions for lattice problems and they are computationally difficult to explore. In next chapter, I will introduce
the notions of complexity theory and we will find lattice problems are, no doubt, among one of the most difficult problems to deal with.

3 Computational Complexity

3.1 Basic Notion

This section will provide an introduction to basic notions of computational complexity for the audience to understand P, NP, and NP-hard. However, the details of Turing machine and related information will not be covered[GL87]. An alphabet is a finite set comprising symbols denoted as Σ. A string over Σ is a finite sequence of symbols from Σ. The number of symbols in a string y is its length, denoted by |y|. The collection of all sequences of symbols over Σ is expressed by Σ*. A Turing machine M operates within a time-frame t(n) if, given any input string w of length n from an established input alphabet Σ, M(n) ceases functioning after no more than t(n) steps. Efficient computation via Turing machines is characterized by those halting within a time frame polynomial in the input size; that is, the problem is computable by the Turing machine in $t(n) = n^a + b$ time, where $a, b$ are constants.

An important special case of functions in the context of computational complexity is the Boolean functions, which maps strings to strings and outputs a single bit. We identify such a function f with the set

$$L_f = \{ x : f(x) = 1 \}$$

and call such sets languages or decision problems. The ensemble of decision problems solvable by a deterministic Turing machine in polynomial time is termed as class P. Analogously, a decision problem / language is defined to be in NP(nondeterministic polynomial time) if given an input $x$, it is easily verified that $x$ is a YES instance of the problem in polynomial time by a deterministic Turing machine or it is addressable by a nondeterministic Turing machine within polynomial time.

It is readily apparent that $P \subseteq NP$, which is to say that any problem solvable in polynomial time ($P$) also permits its solutions to be verified within polynomial time ($NP$). This insight comes from a straightforward observation: if a problem can be tackled and resolved in a relatively swift manner, then it stands to reason that verifying a proposed solution for that problem should also be a quick process. One could simply reapply the problem-solving steps using the provided solution and check if the outcomes align. Thus, it is established that every problem in $P$ inherently belongs to $NP$, rendering $P \subseteq NP$.

The inquiry into whether $P$ is equivalent to $NP$—that is, whether every problem whose solution can be verified in polynomial time is also susceptible to being resolved within the same time
complexity—stands as one of the paramount and yet unresolved conundrums in the domain of computer science.

But how do we ascertain that one problem is computationally more challenging than another? This is where the concept of NP-hardness comes into play. A problem is considered NP-hard when solving it is at least as difficult as the hardest problems in NP, meaning that an efficient solution to an NP-hard problem would result in efficient solutions to all problems in NP.

Let $A$ and $B$ be two decision problems. A Karp reduction from $A$ to $B$, denoted as $A \leq_P B$, is a polynomial time computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that $x \in A$ if and only if $f(x) \in B$. A decision problem $A$ is NP-hard if any other NP problem $B$ can be Karp-reduced to it. Furthermore, we say $B$ is NP-complete if it is NP-hard and, additionally, $B \in$ NP.

### 3.2 Cook-Levin Theorem

After presenting the foundational concepts, this section delves into the Cook-Levin theorem. Cook and Levin, through their independent research, unveiled the concept of NP-completeness. An exposition of one of the most elementary NP-complete problems will be provided. To facilitate this, it is crucial to understand the definition of a Boolean formula. A Boolean formula is constructed using the variables $v_1, \ldots, v_n$, which are interconnected by the logical operators AND ($\land$), NOT ($\lnot$), and OR ($\lor$).

Given a Boolean formula $\varphi$ with variables $v_1, \ldots, v_n$ and a vector $z \in \{0, 1\}^n$, the value of $\varphi$ corresponding to $z$ is denoted by $\varphi(z)$, where the assignment of 1 is equated to True and 0 to False. A formula $\varphi$ is said to be satisfiable if there exists at least one assignment $z$ such that $\varphi(z)$ is True; otherwise, it is deemed unsatisfiable.

A Boolean formula structured in CNF form (Conjunctive Normal Form) is one that is composed of a conjunction (AND) of one or more clauses, where each clause is a disjunction (OR) of literals, and a literal is either a variable or its negation. An example of a CNF formula, specifically a 3CNF formula, is given by:

$$(v_1 \lor \neg v_2 \lor v_3) \land (v_2 \lor v_3 \lor v_4) \land (\neg v_1 \lor \neg v_3 \lor \neg v_4),$$

where $\neg v$ signifies the negation of the variable $v$.

Thus I can introduce the Cook-Levin Theorem,

**THEOREM 4.2.1** [Cook-Levin Theorem][Coo71][AB09] Let $SAT$ be the language of all satisfiable CNF formulae and $3SAT$ be the language of all satisfiable 3CNF formulae. Then,

1. SAT is NP-complete.
2. 3SAT is NP-complete.

To establish that SAT is NP-complete, we must demonstrate two parts of the proof: firstly, that SAT is in NP, and secondly, that SAT is NP-hard. It is straightforward to verify that SAT is in NP, as given a satisfying assignment, we can quickly check the satisfiability of a formula. The proof that SAT is NP-hard is more involved. Although this theorem is not the central focus of the thesis, a brief overview of the proof for SAT’s NP-hardness will be provided. For a comprehensive exposition, readers should consult the detailed proofs referenced in the bibliography.

Let $L$ be an NP language and let $M$ be the polynomial time Turing Machine (TM) such that for every $x \in \{0, 1\}^*$, $x \in L$ if and only if $M(x, u) = 1$ for some $u \in \{0, 1\}^{p(|x|)}$, where $p : \mathbb{N} \rightarrow \mathbb{N}$ is some polynomial. We need to demonstrate that $L$ is polynomial-time Karp reducible to SAT by outlining a procedure to transform, in polynomial time, every string $x \in \{0, 1\}^*$ into a CNF formula $\varphi_x$ such that $x \in L$ if and only if $\varphi_x$ is satisfiable.

Consider $L$ as an NP language, and let $M$ represent a Turing Machine (TM) that operates in polynomial time. For any binary string $x$, $x$ belongs to $L$ if and only if there is some string $u$, for some $u \in \{0, 1\}^{p(|x|)}$, such that $M(x, u)$ evaluates to 1. Here, $p$ is a polynomial function mapping natural numbers to natural numbers.

Our objective is to show that $L$ is Karp reducible to SAT in polynomial time. To accomplish this, we need to describe a method to convert each binary string $x$ into a CNF formula $\varphi_x$. This formula $\varphi_x$ is crafted such that $x$ is an element of $L$ if and only if $\varphi_x$ turns out to be satisfiable.

To construct $\varphi_x$, we encode the entire computation history of $M$ as a sequence of configurations. Each configuration is represented by a string of variables indicating the state of the Turing machine at a given time step. The formula $\varphi_x$ comprises clauses that enforce the correct transition between successive configurations according to the rules of $M$.

The proof leverages the concept of a Turing machine’s "locality" — the idea that the state at each step of the computation depends only on a small, fixed number of bits, which correspond to the machine’s current state, the symbol under the machine’s head, and the content of the tape cells within the head’s immediate vicinity. The logical operators within $\varphi_x$ ensure that each step of $M$’s computation is simulated accurately.

The resulting CNF formula $\varphi_M$ is thus a conjunction of clauses, each representing a small part of the Turing machine’s computation. If the machine can reach an accepting state for some input, then $\varphi_x$ will be satisfiable, as there exists a sequence of variable assignments corresponding to a valid computation history leading to acceptance.

This construction is executed in polynomial time relative to the length of the input, satisfying the condition for Karp reductions and thereby confirming the NP-hardness of SAT. Detailed proof and the
corresponding Turing machine simulations are available in the origin bibliography.

### 3.3 Knapsack and SSP problems

With the NP-hardness of the SAT problem established, the endeavor of proving the NP-completeness of other problems within the realm of NP becomes substantially streamlined. This facilitation arises from the ability to employ SAT as a starting point for subsequent polynomial-time reductions.

This section introduces a class of combinatorial optimization problems termed the 'knapsack' problems. At first glance, the knapsack problems may appear disjoint from the domain of lattices. However, as we will explore, they serve as a formidable tool in our computational arsenal, offering profound implications for lattice-based approaches and strategies.

The knapsack problem poses a quintessential question in combinatorial optimization: given a set of items, each characterized by a weight and a value, which subset of items should be selected to maximize the total value while ensuring the cumulative weight does not exceed a predetermined capacity limit? Then I introduce the definition of the Subset Sum Problem, which is a special case of the knapsack problems.

**DEFINITION 4.3.1** [Subset Sum Problem, SSP] Consider a sequence of $n+1$ integers $(a_1, \ldots, a_n, T)$. The task is to ascertain whether there exists a subset of the $a_i$’s that sums to $T$. More formally, the objective is to identify coefficients $x_i \in \{0, 1\}$ such that the equation $\sum_{i=1}^{n} a_i x_i = T$ holds true. Within the decision variant of the subset sum problem, one is provided the set $(a_1, \ldots, a_n, T)$ and must determine the feasibility of finding such coefficients $x_i$ that satisfy the aforementioned sum equation.

**Proof.** This proof is based on the approach outlined in the notes by Lenny Fukshansky [Fuk22]. To establish the NP-completeness of the Subset Sum Problem (SSP), we undertake a two-pronged approach. Initially, we must verify that SSP resides in the complexity class NP. Consider a collection of weights $\mathcal{A} = \{A_1, \ldots, A_n\}$. Given a specific subset $\mathcal{A}' \subseteq \mathcal{A}$, the verification process entails summing the elements of $\mathcal{A}'$ to check if the total equals the target sum $T$. This verification can be performed in linear time due to the bounded length of $\mathcal{A}'$, thereby confirming that SSP is in NP.

Subsequently, to demonstrate that the decision version of SSP is NP-hard, we construct a polynomial-time reduction from the well-known NP-complete problem, 3-SAT. This reduction maps an instance of 3-SAT to an instance of SSP such that the former is satisfiable if and only if the corresponding SSP instance has a subset summing to $T$.

Let us define $\Phi$ as a Boolean formula with $n$ variables and $k$ clauses. Each variable $x_i$ has two corresponding representations: $x_{i0}$ indicating the choice of $x_i$ being FALSE, and $x_{i1}$ representing $x_i$ as TRUE. Additionally, slack variables are introduced to ensure that the sum within a clause’s column equates to 3. We can construct a table with $2(n + k) + 1$ rows and $n + k$ columns.
The values within this table are assigned based on the following conditions:

1. For the variable \( x_i \), \( x_{i1} \) signifies the variable taking the value TRUE, while \( x_{i0} \) denotes it as FALSE. Consequently, the table cell at the intersection of \( x_{i1} \) and \( x_j \), or \( x_{i0} \) and \( x_j \), is set to 1 if \( i = j \) and 0 otherwise.

2. For a clause \( c_j \), the cell corresponding to \( x_{i1} \) and \( c_j \) receives a 1 if assigning TRUE to \( x_i \) satisfies \( c_j \), and 0 if it does not. Similarly, the cell at \( x_{i0} \) and \( c_j \) is set to 1 if setting \( x_i \) as FALSE satisfies \( c_j \), and 0 otherwise.

3. Slack row cells corresponding to \( s_{il} \) and \( x_j \) are set to 0 for all \( i \), \( l \), and \( j \).

4. The cell at \( s_{il} \) and \( c_j \) is assigned a value of 1 if \( i = j \) and 0 otherwise, for each \( 1 \leq l \leq n - 1 \).

5. In the row \( T \), cells corresponding to a variable \( x_i \) are filled with 1, while those corresponding to a clause \( c_i \) are set to \( n \), the total number of variables.

We present an illustrative example for the boolean formula \( \Phi \) that is defined as \( \Phi = (x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3) \land (x_1 \lor \neg x_2 \lor \neg x_3) \land (x_1 \lor \neg x_2 \lor x_3) \). Assuming \( n = 3 \) variables and \( k = 4 \) clauses, the reduction table is constructed as follows:

<table>
<thead>
<tr>
<th></th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( c_1 )</th>
<th>( c_2 )</th>
<th>( c_3 )</th>
<th>( c_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_{11} )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( v_{12} )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( v_{21} )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( v_{22} )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( v_{31} )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( v_{32} )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( s_{11} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s_{12} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s_{21} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s_{22} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s_{31} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( s_{32} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( s_{41} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s_{42} )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( T )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Let us define $\mathcal{A}$ as the set comprising of all binary strings represented in the aforementioned table, excluding the target sum $T$. Therefore,

$$\mathcal{A} = \{1001011, 1000100, 0101001, 0100110, 0011110, 0010001, 0001000, 0001000, 0000100, 0000100, 0000010, 0000001, 0000001\}$$

and $T = 1113333$. The satisfiability of the Boolean formula $\Phi$ encoded by this table is contingent upon the existence of a subset $\mathcal{A}' \subseteq \mathcal{A}$ whose summation equals $T$.

The initial segment of $T$ comprises solely ones, reflecting the necessity to select either $v_{i0}$ or $v_{i1}$ but not both, corresponding to the truth assignment of variable $x_i$. Slack rows are then employed to adjust the sum of rest segment in each clause column to the value of $n$. From the existence of a satisfying assignment, it is inferred that at least one literal within each clause must evaluate to true; should all literals within any single clause evaluate to false, the clause—along with the entire Boolean formula $\Phi$—would consequently be unsatisfiable. Hence, it is assured that each clause column contains at least one true literal. The inclusion of slack rows serves to guarantee that the sum of the entries in every clause column reaches $n$, thereby upholding the satisfiability condition.

In the provided example, the selection of rows $v_{11}, v_{21}, v_{31}$ and slack rows $s_{21}, s_{22}, s_{31}, s_{41}$ yields a sum that equates to $T$, thereby satisfying $\Phi$: $1001011 + 0101001 + 0011110 + 0001000 + 0000100 + 0000010 + 0000001 = 1113333$. This demonstrates the existence of the desired subset within $\mathcal{A}$, fulfilling the target sum $T$, and thus, establishing a reduction from $\Phi$ to an instance of the Subset Sum Problem (SSP).

Lastly, we affirm that the reduction is accomplished in polynomial time. Each element in $\mathcal{A}$ is encoded with at most $n + k$ digits and there are at most $2^n + (n - 1)k$ elements in $\mathcal{A}$. Given the finite nature of the table, the reduction procedure’s computational complexity is polynomial with respect to the size of the input parameters $n$ and $k$.

\[\square\]

4 Lattice Problems in Computational Complexity

4.1 SVP and CVP

Having established the fundamental NP-complete problems and their reductions, we now arrive at the crux of lattice-based computational challenges. This section is devoted to introducing two of the most renowned computational problems associated with lattices.

Recall that Minkowski’s Successive Minima Theorem stands as a foundational result in the geometric theory of numbers, offering essential bounds on the successive minima of a lattice. Among
all successive minima, $\lambda_1$ is considered to be the length of the shortest nonzero vector in a lattice. Notwithstanding its significance, the proof of Minkowski’s theorem is not constructive, offering existence without furnishing a direct computational pathway to ascertain these vectors. The quest for efficient algorithms to pinpoint these minimal vectors gives rise to several computational lattice problems of profound complexity and practical relevance.

Here is the shared setup for the two problems. Given a basis matrix $A \in GL_n(\mathbb{R})$ satisfying that $\Lambda = AZ^n$, we have its successive minima to be $0 < \lambda_1 \leq \ldots \leq \lambda_n$.

**DEFINITION 5.1.1.** [Shortest Vector Problem, SVP] Given such basis matrix $A$, find a nonzero lattice vector $x_1 \in \Lambda = AZ^n$ such that $\|x_1\| = \lambda_1$.

**DEFINITION 5.1.2.** [Closest Vector Problem, CVP] Given such basis matrix $A$ and a vector $y \in \mathbb{R}^n$, find a nonzero lattice vector $x \in \Lambda = AZ^n$ such that

$$\|x - y\| \leq \|z - y\| \quad \forall z \in \Lambda.$$

## 4.2 CVP is NP-complete

**THEOREM 5.2.1** The decision version of CVP is NP-complete.

We reiterate the decision version of CVP here.

Given an $n \times m$ integer basis matrix $A$, with $m \leq n$, a target vector $t \in \mathbb{Z}^n$, and a number $r > 0$, does there exist a vector $x \in AZ^n$ such that $\|x - t\| \leq r$?

**Proof.** [Fuk22][MG02] We aim to prove that the Closest Vector Problem (CVP) is NP-complete. To begin, we establish that CVP belongs to the complexity class NP. Given a lattice $\Lambda$ generated by a basis matrix $A$ and a target vector $t$, we can verify in polynomial time whether a vector $x \in \Lambda$ satisfies $\|x - t\| \leq r$. The procedure involves computing the difference vectors, calculating their Euclidean norms, and performing a comparison. Each step is executable within polynomial time bounds.

To demonstrate that CVP is NP-complete, we devise a polynomial-time reduction from the SSP to CVP. As established in Section 4.3.1, SSP is NP-complete, and thus a reduction from SSP to CVP substantiates the NP-completeness of CVP.

Let $a = (a_1, \ldots, a_n), T$ be an instance of SSP, i.e., $a$ is the $n$-tuple of weights and $T$ is the target sum. Define the $(n + 1) \times n$ basis matrix $A$ for a lattice $AZ^n \subseteq \mathbb{Z}^{n+1}$ by
\[
A = \begin{pmatrix}
a_1 & a_2 & \cdots & a_n \\
2 & 0 & \cdots & 0 \\
0 & 2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 2 \\
\end{pmatrix} = \begin{pmatrix}
a \\
2 \mathbf{I}_n \\
\end{pmatrix},
\]

and let \( t = (T, 1, \ldots, 1)^T \in \mathbb{Z}^{n+1} \) be the target vector. Then we can consider the instance of decision CVP with \( A, t \) and \( r = \sqrt{n} \), i.e.,

Is there an existing vector \( x \in A \mathbb{Z}^n \) such that \( \|x - t\| \leq \sqrt{n} \)?

Assume \( a = (a_1, \ldots, a_n), T \) is a YES instance of SSP, that is, a set of coefficients \( x_1, \ldots, x_n \in \{0, 1\} \) exists satisfying \( \sum_{i=1}^n a_i x_i = T \), then

\[
Ax - t = \begin{pmatrix}
\sum_{i=1}^n a_i x_i - T \\
2x_1 - 1 \\
\vdots \\
2x_n - 1 \\
\end{pmatrix},
\]

and so \( \|Ax - t\|^2 = \sum_{i=1}^n |2x_i - 1|^2 = n \), since \( 2x_i - 1 = \pm 1 \) for every \( i \), and it also leads to a YES instance for decision CVP. Conversely, a YES instance of decision CVP with \( A \) and \( t \) as above gives a vector \( y = Ax \in A \mathbb{Z}^n \) such that

\[
\|Ay - t\|^2 = \left( \sum_{i=1}^n a_i x_i - T \right)^2 + \sum_{i=1}^n |2x_i - 1|^2 \leq n,
\]

This is only possible if \( \sum_{i=1}^n a_i x_i - T = 0 \), since again \( 2x_i - 1 = \pm 1 \) for every \( i \). Hence we obtain a YES instance of SSP. Therefore, we establish a polynomial-time reduction from SSP to CVP, indicating that CVP is indeed NP-complete.

The proof of the NP-completeness of the Shortest Vector Problem (SVP) is more intricate and thus is not included within the scope of this discourse.

### 4.3 Gauss-Lagrange algorithm in dimension 2

Though computational lattice problems have been shown to be NP-complete, indicating the absence of efficient algorithms under standard complexity assumptions, there has been substantial progress in the discovery of efficient approximation algorithms for SVP and CVP. These algorithms
strive to provide a solution within a factor of an approximation ratio $\gamma$, defined as follows: for an input instance, the goal is to procure an output that does not exceed the optimal solution by more than a multiplicative factor of $\gamma$. For instance, a $\gamma$-SVP seeks a vector $x \in \Lambda$ such that $\|x\| \leq \gamma \cdot \lambda_1$, where $\lambda_1$ denotes the length of the shortest non-zero vector in the lattice.

In this segment, I will introduce the Lagrange-Gauss Algorithm, a classical approach for solving SVP in two dimensions. This algorithm is renowned for its efficiency in computing a shortest basis for lattices within $\mathbb{R}^2$, thereby providing valuable insights and strategies for approximate solutions in higher dimensions.

**Input:** An ordered basis $B = [u, v]$ for the lattice $\Lambda$.

Assume that $\|v\| \leq \|u\|$, otherwise swap them.

**Compute:** the Gram-Schmidt coefficient $\mu = \frac{u^Tv}{\|u\|^2}$

**Check $\mu$:** If $|\mu| \leq \frac{1}{2}$, output $u, v$;

else set $v \leftarrow v - \lceil \mu \rceil u$ and repeat the algorithm

(swapping $u, v$, if necessary, to ensure $\|u\| \leq \|v\|$)

**Output:** A basis $[u, v]$ with $\|u\| = \lambda_1(\Lambda), \|v\| = \lambda_2(\Lambda)$.

**Proof.** In this proof, I try to show $[u, v]$ is a minimal basis. Let $m = \alpha u + \beta v$ be an arbitrary nonzero element of $\Lambda$.

First, we will show that $\|u\| \leq \|m\|$, implying that $u$ is the smallest nonzero vector.

$$\|m\|^2 = \|\alpha u + \beta v\|^2$$

$$\quad = \alpha^2 \|u\|^2 + 2\alpha\beta (u \cdot v) + \beta^2 \|v\|^2$$

$$\quad \geq \alpha^2 \|u\|^2 - \alpha\beta \|u\|^2 + \beta^2 \|v\|^2 \quad \text{(since } \mu \geq -\frac{1}{2}\text{)}$$

$$\quad \geq (\alpha^2 - \alpha\beta + \beta^2) \|u\|^2 \quad \text{(since } \|u\| \leq \|v\|\text{)}$$

$$\quad = (\alpha - \frac{1}{2} \beta)^2 + \frac{3}{4} \beta^2 \|u\|^2$$

$$\quad \geq \|u\|^2 \quad \text{(since } \alpha \neq 0 \text{ or } \beta \neq 0\text{)}$$

Similarly, we will show $\|v\| \leq \|m\|$ when $\beta \neq 0$, implying that $v$ is the smallest nonzero vector
which is not a multiple of $u$.

$$\|m\|^2 = \|\alpha u + \beta v\|^2$$

$$= \alpha^2 \|u\|^2 + 2\alpha \beta (u \cdot v) + \beta^2 \|v\|^2$$

$$= \beta^2 (\|v\|^2 - \|u\|^2) + (\alpha^2 + \beta^2) \|u\|^2$$

$$\geq \beta^2 (\|v\|^2 - \|u\|^2) + (\alpha^2 - \alpha \beta + \beta^2) \|u\|^2$$

$$\geq \beta^2 (\|v\|^2 - \|u\|^2) + \|v\|^2$$

$$\geq \|v\|^2$$

Hence, we obtain the ordered minimal basis $[u, v]$.

We can conclude that the output of the algorithm is the ordered minimal basis $[u, v]$. The algorithm performs elementary column operations at each step, ensuring that the basis $[u, v]$ is always a basis of the original lattice. Therefore, we only need to demonstrate that the algorithm always terminates.

It is easy to observe that $\min\{\|u\|, \|v\|\}$ will be replaced by a new basis strictly less in every iteration. The lengths of $u$ and $v$ can only be equal in the final step. Moreover, since there are only finitely many lattice vectors shorter than any given constant, the algorithm must terminate after a finite number of iterations. Hence, we finish the proof.

4.4 The LLL approximation algorithm

In the preceding section, we delineated a polynomial-time algorithm capable of identifying the shortest vector within 2-dimensional lattices. The objective of this section is to extend these results to encompass $n$-dimensional lattices. However, the extension introduces a certain complexity; we demonstrate that while we can ascertain a vector in a reduced basis, this vector’s length is within an exponential factor $\gamma(n) = \left(\frac{2}{\sqrt{3}}\right)^n$ of the shortest vector in the lattice.

Let $B = \{b_1, \ldots, b_n\}$ be a basis for an $n$-dimensional lattice $\Lambda$ in $\mathbb{R}^n$ and let $\{b^*_1, \ldots, b^*_n\}$ be the corresponding orthogonal basis obtained through the Gram-Schmidt orthogonalization process. Similar strategy was covered in the last proof, but I will give a formal definition here.

For a given basis $B = \{b_1, \ldots, b_n\}$ of a lattice, the orthogonalization process constructs a new set of vectors $\{b^*_1, \ldots, b^*_n\}$, where each $b^*_i$ is orthogonal to all vectors $b^*_j$ with $j < i$. The process is
defined recursively by:

\[ b_1^* = b_1, \]
\[ b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^*, \quad \text{where } \mu_{i,j} = \frac{b_i^T b_j^*}{\|b_j^*\|^2}. \]

**DEFINITION 5.4.1 [LLL reduced basis]** A basis \( B = [b_1 \ldots b_n] \in \mathbb{R}^n \) is **LLL-reduced** with parameter \( \delta \) (\( \delta \)-LLL-reduced) if

1. \( |\mu_{i,j}| \leq \frac{1}{2} \) for all \( i > j \), where \( \mu_{i,j} \) are the Gram-Schmidt coefficients,

2. For some parameter \( \delta \in \left[ \frac{1}{4}, 1 \right] \), for all \( 1 \leq i \leq n \),

\[ \delta \|b_i^*\|^2 \leq \|b_i\|^2 + \mu_{i,(i-1)}^2 \|b_{i-1}\|^2. \]

If \( \delta = 1 \), the above definition holds for the 2-dimensional basis is reduced. In the subsequent sections, we prove that the first vector in a \( \delta \)-LLL-reduced basis is not substantially longer than \( \lambda_1 \).

We define projection operations \( \pi_i \) onto \( \text{span}(b_i^*, b_{i+1}^*, \ldots, b_n^*) \) by

\[ \pi_i(x) = \sum_{j=1}^{n} x^T b_j^* b_j^*. \]

The Gram-Schmidt orthogonalized vectors can be expressed as \( b_i^* = \pi_i(b_i) \).

**LEMMA 5.4.2 [MG02]** If \( B = [b_1 \ldots b_n] \in \mathbb{R}^n \) is an \( \delta \)-LLL-reduced basis with \( \delta \in [1/4, 1] \), then \( \|b_1\| \leq (2/\sqrt{4\delta - 1})^{n-1} \lambda_1 \). In particular, if \( \delta = (1/4) + (3/4)^{n/(n-1)} \) then \( \|b_1\| \leq (2/\sqrt{3})^n \lambda_1 \).

**Proof.** Notice that if the basis is LLL reduced, then for all \( i \),

\[ \delta \|b_i^*\|^2 = \delta \|\pi(b_i)\|^2 \]
\[ \leq \|\pi(b_{i+1})\|^2 \]
\[ = \|b_{i+1}^* + \mu_{i+1,i} b_i^*\|^2 \]
\[ = \|b_{i+1}^*\|^2 + \mu_{i+1,i}^2 \|b_i^*\|^2 \]
\[ \leq \|b_{i+1}^*\|^2 + \frac{1}{4} \|b_i^*\|^2 \]
\[ \left( \delta - \frac{1}{4} \right) \|b_i^*\|^2 \leq \|b_{i+1}^*\|^2. \]

By induction on \( i - j \), this implies that for all \( i \geq j \)

\[ \left( \delta - \frac{1}{4} \right)^{i-j} \|b_j^*\|^2 \leq \|b_i^*\|^2, \]
and we can extend the result to $n$,

$$
\|b_i^*\| \geq \left( \delta - \frac{1}{4} \right)^{\frac{i-1}{2}} \|b_i^*\| \geq \left( \delta - \frac{1}{4} \right)^{\frac{n-1}{2}} \|b_1\|.
$$

We need to use a lemma to continue our proof.

**Lemma 5.4.3** The first minimum of the lattice satisfies

$$
\lambda_1 \geq \min_i \|b_i^*\| > 0.
$$

**Proof.** Given a non-zero lattice vector $u$,

$$
u = \sum_{i=1}^k a_i b_i
= \sum_{i=1}^k a_i \left( b_i^* + \sum_{j=1}^{i-1} \mu_{i,j} b_j^* \right)
= \sum_{i=1}^k a_i \left( \sum_{j=1}^i u_{i,j} b_j^* \right)
= a_k b_k^* + \sum_{j=1}^{k-1} \sum_{i=j+1}^k a_i u_{i,j} b_j^*.
$$

Thus,

$$
\|v\| \geq \|a_k b_k^*\| \geq \|b_k^*\|.
$$

As $u$ can be any non-zero lattice vector, then the conclusion is reached that $\lambda_1 \geq \min_i \|b_i^*\| > 0$. Applying this lemma and using the lower bound,

$$
\lambda_1 \geq \min_i \|b_i^*\|^2 \geq \left( \delta - \frac{1}{4} \right)^{\frac{n-1}{2}} \|b_1\|,
$$

Setting $\delta = (1/4) + (3/4)^{n/(n-1)}$, we get $\|b_1\| \leq (2/\sqrt{3})^n \lambda_1$.

Next, we describe the algorithm for computing LLL reduced bases as presented by Lenstra, Lenstra, and Lovász in 1982. This algorithm follows the structure of Gauss-Lagrange algorithm and is
commonly referred to as the LLL algorithm. The procedure is as follows:

**Input:** A basis $B = [b_1, \ldots, b_n]$ for the lattice $\Lambda$.

*(loop) Compute:* for $i = 1, \ldots, n$, do

for $j = i - 1, \ldots, 1$, do

$$b_i := b_i - \mu_{i,j} b_j,$$

where $\mu_{i,j} = \frac{b_i^T b_j^*}{\|b_j^*\|^2}$.

**Check:** if $\delta \|\pi_i(b_i)\|^2 \geq \|\pi_i(b_{i+1})\|^2$ for some $i$

then swap $b_i$ and $b_{i+1}$ and repeat the algorithm

**Output:** An LLL reduced basis $B^*$ for $\Lambda$.

**Proof.** In this section, I establish the validity of the output of the LLL algorithm, asserting that it indeed produces an LLL reduced basis for the input lattice. I denote by $B$ a basis of the input lattice with vectors $[b_1, \ldots, b_n]$. The orthogonalized basis $B^*$ is derived from $B$ through a series of elementary integer column operations and the conditional swapping of columns given the LLL algorithm’s procedures, thereby affirming that $B$ and $B^*$ are equivalent bases.

As per the definition presented in Definition 5.4.1, a basis is LLL reduced if it satisfies two specific properties. It is straightforward to verify that $B^*$ satisfies the first property. We will now focus our attention on proving the second property.

In particular, for any consecutive vector pair $b_i, b_{i+1}$, the second LLL condition requires that:

$$\delta \|\pi_i(b_i)\|^2 \leq \|\pi_i(b_{i+1})\|^2.$$

Upon detecting a violation of this condition, the algorithm swaps $b_i$ and $b_{i+1}$ and repeats the reduction process. This ensures that every swap reduces the length of the basis vectors, progressing towards an LLL reduced basis. It becomes evident that once the reduction step concludes without the necessity of swapping any consecutive vector pair, we can affirm that the basis $B$ is indeed LLL-reduced.

Assuming that both properties for an LLL-reduced basis are met and acknowledging that the final matrix $B^*$ remains equivalent to the original matrix $B$, we can deduce that the algorithm’s termination guarantees an LLL-reduced basis as an output. To prove that, I will establish both the number of iterations is polynomial in the input size, and each iteration takes polynomial time. The finer details are delineated in the reference [MG02]. It has been demonstrated that the LLL algorithm, through a polynomial number of steps, allows all intermediate values to be represented within a
polynomial bit size. It first asserts that the LLL algorithm completes within a polynomial number of iterations. Furthermore, throughout the execution of the LLL algorithm, all intermediate quantities can be expressed using a number of bits that is polynomial in the size of the input. Considering the arithmetic operations per iteration are polynomial, thus the entire algorithm, operates within polynomial time bounds.

5 Lattice Cryptography Applications

5.1 Introduction to asymmetric cryptography

Cryptography is devoted to the creation of protocols that safeguard communication against adversarial breaches, aspiring to maintain the confidentiality of messages exchanged between two legitimate parties. This domain deals with the analytical and developmental aspects of protocols to prevent unauthorized access to information. The quintessence of cryptography lies in facilitating private conversations; where only the intended recipients, equipped with a shared secret key, can decode the sent message. Encryption transforms the original message (cleartext) into a coded format (ciphertext), and decryption reverses this process. The goal is to render the extraction of any information from the ciphertext as computationally prohibitive without the secret key. Consequently, an adversary, upon intercepting the ciphertext, is unable to glean any insights about the message except for its existence and length. The permissible disclosure of message length is often a trade-off for practical efficiency and is considered a tolerable deviation from absolute secrecy in secure communication.

More specifically, asymmetric-key cryptography, often termed public-key cryptography, operates on a key pair mechanism that employs a public and a private key for the encryption and decryption of data. The public key is made openly accessible, while the private key is retained confidentially by the owner.

One of the most illustrative examples of asymmetric cryptography involves two parties, typically referred to as Alice and Bob. In this scenario, Alice wishes to send a secure message $m$ to Bob. To achieve this, $m$ is first converted into a ciphertext $C$ using Bob’s public key $k_b$. This encryption process is denoted by $E$, where $C = E(m, k_b)$. Upon receiving $C$, Bob can then convert it back into the plaintext $m$ using his private key $k_v$, through the decryption algorithm $D$, such that $m = D(C, k_v)$. This process ensures that the communication between Alice and Bob remains confidential and secure against eavesdroppers.

In cryptography, the critical measure of a system’s robustness is average-case hardness, which stipulates that if a key is randomly selected, then no probabilistic polynomial-time algorithm should be able to compromise the cryptographic scheme with any significant probability. This contrasts sharply
with the worst-case scenario typically used in computational complexity, such as NP-completeness, where demonstrating NP-hardness implies the absence of a polynomial-time solution for all instances of a problem. As the merge of the quantum computer, it is insufficient to have some keys that are hard to break; security must extend to virtually all keys generated through standard procedures to ensure their practical unbreakability.

However, constructing cryptosystems that are provably hard to break remains a theoretical challenge akin to resolving the P vs. NP problem. Establishing such security unequivocally would essentially require iterating proofs pertaining to fundamental questions of computational complexity. Consequently, the second most sought-after goal in cryptography is to develop cryptosystems that are provably hard to break on average, based on well-established computational complexity assumptions. Currently, no such constructions are definitively known, but the seminal work by Ajtai [Ajt96], which bridges the gap between worst-case and average-case complexities using lattice-based approaches, offers promising avenues.

5.2 The Rivest-Shamir-Adleman (RSA) Cryptosystem

In this section, we introduce the RSA cryptosystem, which remains one of the most prevalently used encryption tools in the digital world today. The choice of RSA serves as a foundational comparison for the lattice-based cryptographic systems discussed later in this chapter.

The security of RSA is primarily based on the assumption that factoring large integers is computationally infeasible. More specifically, RSA’s security relies on the assumption that factoring is hard not only in the worst case but also on average. This assumption is critical because the strength of RSA encryption is compromised if an adversary is able to efficiently factor the product of the two large prime numbers used to generate the public and private keys.

Consider the scenario where one of the prime factors of an RSA modulus is significantly smaller than the other. If the smaller factor is sufficiently small, an adversary could feasibly employ brute-force techniques to discover this factor. Once the smaller factor is known, the security of the RSA cryptosystem is effectively broken, as the remaining factor can be easily computed. This vulnerability underscores the importance of selecting prime numbers of similar and substantial size when implementing RSA to ensure the robustness of the encryption.

The RSA algorithm comprises three pivotal steps: key generation, encryption, and decryption. Here, we detail each step:

1. **Key Generation**: Begin by selecting two large prime numbers \( p \) and \( q \), and compute \( m = pq \), where \( m \) is part of the public key. The function \( \phi(m) \), Euler’s totient function, is calculated as
\( \phi(m) = (p - 1)(q - 1) \). Choose \( k \), a number relatively prime to \( \phi(m) \), which will also be part of the public key. The private key consists of \( p \) and \( q \).

2. **Encryption**: Assume the plaintext consists of letters. Each letter is encoded into numbers; for instance, \( a \to 11 \), \( b \to 12 \), and so forth. After encoding, the message is divided into blocks \( a_1, \ldots, a_n \) such that each block’s length \( \leq m \). Each block \( a_i \) is encrypted using the formula \( b_i = a_i^k \mod m \). Successive squaring can be employed for efficient computation. For example, to compute \( 7^{19} \mod 11 \):

\[
\begin{align*}
7^2 &\equiv 5 \mod 11, \\
7^4 &\equiv 3 \mod 11, \\
7^8 &\equiv 9 \mod 11, \\
7^{16} &\equiv 4 \mod 11, \\
7^{19} &\equiv 7^{16} \cdot 7^2 \cdot 7 = 4 \cdot 5 \cdot 7 \equiv 8 \mod 11.
\end{align*}
\]

The sequence \( b_1, \ldots, b_n \) forms the encrypted message.

3. **Decryption**: Having received the encrypted blocks \( b_1, \ldots, b_n \), the goal is to retrieve the original plaintext. To decrypt, we first compute \( \phi(m) \). Next, we need to find integers \( x \) and \( y \) such that the Diophantine equation \( kx - \phi(m)y = 1 \) holds, which is always solvable since \( \gcd(k, \phi(m)) = 1 \).

Each ciphertext block \( b_i \) is then raised to the power \( x \) modulo \( m \), using successive squaring, to compute \( a_i \) as follows:

\[
a_i^k = (b_i^x)^k = b_i^{xk} = b_i^{1+\phi(m)y} = b_i(b_i^{\phi(m)})^y \equiv b_i^x \mod m
\]

which also congruence to

\[
a_i^k \equiv 1 \mod m
\]

by Euler’s theorem if \( \gcd(b_i, m) = 1 \), which is highly likely for large \( m \). This process reconstructs \( a_i \) from \( b_i \), effectively decrypting the message.

Understanding \( \phi(m) \) is crucial for the decryption process. While \( \phi(m) \) is straightforward to determine with knowledge of the private keys \( p \) and \( q \), it remains secure against adversaries who only have access to the public key.

In conclusion, the correctness of the RSA algorithm is fundamentally based on Fermat’s Little Theorem. For a comprehensive understanding of this theorem and its application in the proof of RSA’s
correctness, readers are encouraged to refer to the original proof. The security and hardness of the RSA algorithm hinge on the problem of integer factorization. While this problem is acknowledged to reside within the complexity class NP, it has not yet been proven to be NP-complete.

5.3 The Goldreich-Goldwasser-Halevi (GGH) cryptosystem

We now explore lattice-based cryptographic schemes, where security is predicated on the assumption that decoding without the private key is computationally challenging, specifically based on the Closest Vector Problem (CVP). The protocol is structured into three phases: key generation, encryption, and decryption [MG02].

1. **Key Generation**: The scheme utilizes a pair of keys; the private key comprises a short basis $B$ for a lattice $L$ in $\mathbb{R}^n$ and a unimodular matrix $U$ in $\text{GL}_n(\mathbb{Z})$. The public key is another basis $B' = BU$ for the same lattice $L$. The message $m$ is represented as a vector $m = (m_1, ..., m_n)$ in $\mathbb{Z}^n$, with a small error vector $\epsilon$ added to maintain security.

2. **Encryption**: The plaintext vector $m$ is encrypted as $m' = B'm + \epsilon$, where $m$ consists of integer values, and $m'$ also constitutes a point in the lattice.

3. **Decryption**: Given the private key $B$ and $U$, one can compute $B^{-1}$ and $U^{-1}$. The decryption is performed by calculating:

$$B^{-1}m' = B^{-1}BUm + B^{-1}\epsilon = Um + B^{-1}\epsilon \quad (11)$$

Babai’s nearest plane algorithm is then employed to approximate $Um$ by addressing the associated CVP, effectively nullifying the effect of $B^{-1}\epsilon$ provided it is sufficiently small. Finally, $m$ is recovered by computing $m = m \cdot U \cdot U^{-1}$.

On the other hand, an adversary armed solely with the public key faces significant obstacles. Attempting to decrypt, the adversary would compute $(B')^{-1}m' = m + (B')^{-1}\epsilon$. However, the magnitude of $(B')^{-1}\epsilon$ can be sufficiently large to yield an incorrect solution to the Closest Vector Problem (CVP). Specifically, the resulting lattice vector derived from this decryption attempt would likely differ from the original message vector $m$, thereby failing to reveal the true plaintext.

However, as noted by Nguyen in 1999 [Ngu99], this cryptographic scheme potentially reveals some information about the plaintext because the decryption problem can be converted into a special case of CVP that is easier to solve than the general CVP. While Nguyen’s attacks are not asymptotic—they show that the system can be efficiently compromised with certain security parameters—increasing the security parameter can mitigate these vulnerabilities.
This discussion outlines the structure of the GGH lattice-based cryptographic scheme. For more detailed explanations on processes such as how to generate the public key from the private key through random generation, please refer to the original proofs detailed in the foundational literature on this cryptographic method.

5.4 Conclusion

This thesis began with an introduction to lattices, an algebraic structure that offers a new perspective on mathematical problems. Lattices boast intriguing properties, such as determinant and successive minima. A significant focus was placed on Minkowski’s theorem on successive minima, which elegantly bounds these properties. However, while the theorem provides bounds, it does not offer constructions for achieving them. To address this gap, I explored several lattice-related problems, including lattice packing, lattice covering, and kissing number. These problems, while seemingly intractable, underscore the necessity of understanding computational complexity.

The discussion on complexity commenced with the Cook-Levin theorem, extending to the broader realm of NP-completeness with applications to the subset sum problem. In Chapter 3, this concept was leveraged to establish that problems such as SVP and CVP are NP-complete. Despite the formidable challenge these problems present, the pursuit of efficient algorithms has not waned. I detailed algorithms for solving the CVP, including the Gauss-Lagrange method in two dimensions and the LLL approximation algorithm for higher dimensions.

In the final chapter, the application of lattices in cryptography was introduced, demonstrating the practical implications of mathematical theories. The discourse included an exposition of asymmetric cryptography, detailing both RSA and GGH cryptographic schemes and contrasting factorization-based cryptography with lattice-based approaches.

It is my hope that this thesis not only illuminates the mathematical beauty of lattices but also underscores their profound impact on real-world applications, bridging the abstract with the applicable and delivering a comprehensive view of lattice theory and its cryptographic relevance.
References


