ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Λογικής και Επιστήμης Υπολογισμών

Lattices and Cryptography

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Γεωργίου Α. Ζηρδέλη

Επιβλέπων: Ευστάθιος Ζάχος
Καθηγητής Ε.Μ.Π.

Αθήνα, Δεκέμβριος 2012

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Λογικής και Επιστήμης Υπολογισμών

Lattices and Cryptography

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Γεωργίου Α. Ζηρδέλη

Επιβλέπων: Ευστάθιος Ζάχος
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 9η Νοεμβρίου 2012.

....	....	....
Ευστάθιος Ζάχος	Αριστείδης Παγουρτζής	Δημήτρης Φωτάκης
Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.	Λέκτορας Ε.Μ.Π.

Αθήνα, Δεκέμβριος 2012.

....
Γεώργιος Α. Ζηρδέλης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Τα πλέγματα μελετήθηκα για πρώτη φορά απο τους J. L. Lagrange και C. F. Gauss στην θεωρία των τετραγωνικών μορφών στα τέλη του 18ου αιώνα αλλά και αργότερα από άλλους μαθηματικούς. Με την επινόηση της αλγοριθμικής θεωρίας αριθμών τα πλέγματα ήρθαν και πάλι στο προσκήνιο περίπου το 1980 και ειδικότερα μετά την εφεύρεση του περίφημου LLL αλγορίθμου το 1982. Έκτότε τα πλέγματα αποτελούν μια ενεργή περιοχή έρευνας στην θεωρητική πληροφορική και έχουν εφαρμογές, μεταξύ άλλων, στην υπολογιστική θεωρία αριθμών, στην κρυπτογραφία, στην κρυπτανάλυση και στον ακέραιο προγραμματισμό και επιπλέον έχουν μερικές μοναδικές ιδιότητες από πλευράς υπολογιστικής πολυπλοκότητας.

Κρυπτογραφικά σχήματα βασισμένα σε πλέγματα εμφανίστηκαν για πρώτη φορά στην πρωτότυπη δουλειά του M. Ajtai το 1996 και έχουν αναπτυχθεί σημαντικά από τότε. Ο Ajtai παρουσίασε μια οικογένεια συναρτήσεων μονής κατεύθυνσης των οποίων η ασφάλεια βασίζεται δυσκολία προσέγγισης, εντός ενός πολυωνυμικού παράγοντα στην διάσταση του πλέγματος, της χειρότερης περίπτωσης για το πρόβλημα του μικρότερου διανύσματος στα πλέγματα. Με άλλα λόγια έδειξε ότι αν κάποιος αντιστρέψει με σημαντική πιθανότητα μια συνάρτηση από αυτή την οικογένεια τότε μπορεί να λύσει ένα οποιοδήποτε στιγμιότυπο του προσεγγιστικού προβλήματος, εντός ενός πολυωνυμικού παράγοντα στην διάσταση του πλέγματος, του μικρότερου διανύσματος σε ένα πλέγμα. Αυτή η μοναδική σύνδεση μεταξύ δυσκολότερης και μέσης περίπτωσης έχει ιδιαίτερο ενδιαφέρον για την κρυπτογραφία αλλά και γενικότερα για την θεωρία πολυπλοκότητας.

Ο κύριος σκοπός αυτής της διπλωματικής εργασίας είναι η ανασκόπηση της θεωρίας των πλεγμάτων και της εφαρμογής τους στην κρυπτογραφία. Στο πρωτο κεφάλαιο δίνουμε βασικούς ορισμούς και ιδιότητες των πλεγμάτων ενώ στο δεύτερο κεφάλαιο περιγράφουμε κάποια βασικά υπολογιστικά προβλήματα των πλεγμάτων, περιγράφουμε την έννοια της μειωμένης βάσης πλέγματος με έμφαση στον αλγόριθμο LLL. Στο τρίτο κεφάλαιο παρουσιάζουμε αποτελέσματα από την θεωρία πολυπλοκότητας που αφορούν στα πλέγματα ενώ στο τέταρτο και τελευταίο κεφάλαιο περιγράφουμε κάποια κρυπτοσυστήματα δημοσίου κλειδιού που βασίζονται σε προβλήματα των πλεγμάτων αλλά και κάποια που βασίζονται στο συναφές πρόβλημα, "Εκμάθηση με σφάλματα".

Λέξεις Κλειδιά

πλέγμα, πρόβλημα πλέγματος, μείωση βάσης, αλγόριθμος LLL, πολυπλοκότητα, κρυπτογραφία δημοσίου κλειδιού, εκμάθηση με σφάλματα

Abstract

Lattices were first studied by J. L. Lagrange and C. F. Gauss in the theory of quadratic forms in the late 18th century and later on by other mathematicians. With the advent of algorithmic number theory, the subject had a revival around 1980 especially after the invention of the celebrated LLL algorithm in 1982. Since then lattices have become a topic of active research in computer science and have many applications in computational number theory, cryptography, cryptanalysis and integer programming among others and also have some unique properties from a computational complexity point of view.

Cryptographic schemes based on lattices first emerged in the seminal work of M. Ajtai in 1996 and have developed rapidly in the past few years. Ajtai presented a family of one-way functions whose security is based on the worst-case approximation hardness of the Shortest Vector Problem (SVP) in lattices, within a polynomial factor in the lattice dimension. In other words, he showed that being able to invert a function chosen from this family with non-negligible probability implies the ability to solve *any* instance of approximate SVP within a polynomial factor in the lattice dimension. This remarkable connection between worst-case and average-case complexity in certain lattice problems is of particular interest in cryptography and more general in complexity theory.

The main purpose of this diploma thesis is to overview lattices and their application to cryptography. In the first chapter we give some basic mathematical background on lattices and, while in the second chapter we describe some basic computational lattice problems and introduce the notion of a reduced lattice basis with emphasis on the LLL algorithm. In the third chapter we present complexity results regarding lattice problems and in the fourth and last chapter we describe public-key encryption schemes that are based on lattice problems and some that are based on the related problem, *"Learning with errors"*.

Keywords

lattice, lattice problem, basis reduction, LLL algorithm, complexity, public-key cryptography, learning with errors

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας και των προπτυχιακών μου σπουδών στο Εθνικό Μετσόβιο Πολυτεχνείο, θα ήθελα ευχαριστήσω όλους του ανθρώπους που με βοήθησαν και μοιράστηκαν μαζί μου ένα μέρος από αυτό το ταξίδι.

Ιδιαίτερα θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Ε. Ζάχο που με ώθησε στο να αγαπήσω την Θεωρητική Πληροφορική αλλά και για την ευχάριστη παρουσία του στις σπουδές μου. Στον καθηγητή μου κ. Α. Παγουρτζή που μου έδωσε το έναυσμα για το θέμα της διπλωματικής αυτής εργασίας και στη συνέχεια με συμβούλεψε σε κάθε βήμα της. Στον καθηγητή μου κ. Δ. Φωτάκη για το ενδιαφέρον που έδειξε και για τις χρήσιμες παρατηρήσεις του.

Επίσης, θα ήθελα να ευχαριστήσω τους φίλους μου για την συμπαράσταση και την κατανόηση που έδειξαν σε όλα τα χρόνια των σπουδών μου. Φυσικά, θα πρέπει να ευχαριστήσω όλα τα μέλη του Εργαστηρίου Λογικής και Επιστήμης Υπολογισμών για το δημιουργικό κλίμα και τις ευχάριστες στιγμές που μοιραστήκαμε.

Πάνω απ' όλα όμως, θα ήθελα να ευχαριστήσω την οικογένεια μου για την αγάπη που μου δίνει και την στήριξη που μου παρέχει.

Contents

$\boxed{1}$

Introduction to Lattices

1.1 Vector Spaces

Before we define what a lattice is, we start with some important definitions and ideas from linear algebra.

We regard $n-tuples$ of elements from a field $\mathbb{F}$ as either row vectors or column vectors, and denote them by boldface roman letters:

$$\mathbf{v} = (v_1, v_2, \ldots, v_n) \in \mathbb{F}^n, \qquad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \in \mathbb{F}^n$$

For any field $\mathbb{F}$, and for any positive integer n, the ***vector space*** $\mathbb{F}^n$ consists of all $n-tuples$ of elements from $\mathbb{F}$, with the familiar operations of vector addition and scalar multiplication defined by

$$\mathbf{v}+\mathbf{w} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} v_1 + w_1 \\ v_2 + w_2 \\ \vdots \\ v_n + w_n \end{pmatrix}, \quad a\mathbf{v} = a \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} av_1 \\ av_2 \\ \vdots \\ av_n \end{pmatrix}$$

for any $\mathbf{v}, \mathbf{w} \in \mathbb{F}^n$ and any $a \in \mathbb{F}$.

Definition 1 *Let $V \subset \mathbb{F}^n$ be a vector space. An **inner (scalar) product** on V is a function*

$$\langle \cdot, \cdot \rangle : V \times V \to \mathbb{F}$$

that satisfies the following three conditions for all $\mathbf{v}, \mathbf{u}, \mathbf{w} \in V$ and for all $a, b \in \mathbb{F}$

(a) $\langle \mathbf{v}, \mathbf{v} \rangle \geq 0$ for all $\mathbf{v} \in V$ and $\langle \mathbf{v}, \mathbf{v} \rangle = 0$ if and only if $\mathbf{v} = \mathbf{0}$

(b) $\langle \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{w}, \mathbf{v} \rangle$

(c) $\langle a\mathbf{v} + b\mathbf{w}, \mathbf{u} \rangle = a\langle \mathbf{v}, \mathbf{u} \rangle + b\langle \mathbf{w}, \mathbf{u} \rangle$

For our purposes it is enough to consider vector spaces V that are contained in $\mathbb{R}^n$ for some positive integer n.

Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k \in V$. A ***linear combination*** of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k \in V$ is any vector of the form

$$\mathbf{w} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_k\mathbf{v}_k \quad \text{with } a_1, a_2, \ldots, a_k \in \mathbb{R}$$

The collection of all such linear combinations,

$$\{a_1\mathbf{v}_1 + \cdots + a_k\mathbf{v}_k \, : \, a_1, \ldots, a_k \in \mathbb{R}\}$$

is called the ***span*** of $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$.

A set of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k \in V$ is ***linear independent*** if the only way to get

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_k\mathbf{v}_k = \mathbf{0}$$

is to have $a_1 = a_2 = \cdots = a_k = 0$.

The set is ***linear dependent*** if for $a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_k\mathbf{v}_k = \mathbf{0}$ we have at least one a_i nonzero.

A ***basis*** for V is a set of linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ that span V. This is equivalent to saying that every vector in $\mathbf{w} \in V$ can be written in the form

$$\mathbf{w} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_n\mathbf{v}_n$$

for a *unique* choice of $a_1, a_2, \ldots, a_n$.

We next describe the relationship between different bases and the important concept of dimension.

Proposition 1.1 *Let $V \subset \mathbb{R}^n$ be a vector space.*

(a) *There exists a basis for V.*

(b) *Any two bases for V have the same number of elements. The number of elements in a basis for V is called the **dimension** of V.*

(c) *Let* $\mathbf{v}_1, \ldots, \mathbf{v}_n$ *be a basis for* V *and let* $\mathbf{w}_1, \ldots, \mathbf{w}_n$ *be another set of* n *vectors in* V. *Write each* $\mathbf{w}_j$ *as a linear combination of the* $\mathbf{v}_i$,

$$\begin{aligned}
\mathbf{w}_1 &= a_{11}\mathbf{v}_1 + a_{12}\mathbf{v}_2 + \cdots + a_{1n}\mathbf{v}_n, \\
\mathbf{w}_2 &= a_{21}\mathbf{v}_1 + a_{22}\mathbf{v}_2 + \cdots + a_{2n}\mathbf{v}_n, \\
&\vdots \\
\mathbf{w}_n &= a_{n1}\mathbf{v}_1 + a_{n2}\mathbf{v}_2 + \cdots + a_{nn}\mathbf{v}_n.
\end{aligned}$$

Then $\mathbf{w}_1, \ldots, \mathbf{w}_n$ *is also a basis for* V *if and only if the determinant of the matrix*

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

is not equal to 0.

We next explain how to measure lengths of vectors in $\mathbb{R}^n$ and the angles between pairs of vectors. These important concepts are tied up with the notion of dot product and the Euclidean norm.

Definition 2 *Let* $\mathbf{v}, \mathbf{w} \in V \subset \mathbb{R}^n$ *and write* $\mathbf{v}$ *and* $\mathbf{w}$ *using coordinates as*

$$\mathbf{v} = (v_1, v_2, \ldots, v_n) \quad \textit{and} \quad \mathbf{w} = (w_1, w_2, \ldots, w_n)$$

The ***dot product*** *of* $\mathbf{v}$ *and* $\mathbf{w}$ *is the quantity*

$$\mathbf{v} \cdot \mathbf{w} = v_1 w_1 + v_2 w_2 + \cdots + v_n w_n$$

Definition 3 *Given a vector space* $V \subseteq \mathbb{R}^n$, *a (vector)* ***norm*** *on* V *is a function,* $\|\cdot\| : V \to \mathbb{R}$ *that satisfies the following properties:*

For all $\mathbf{v}, \mathbf{w} \in V$ *and all* $c \in \mathbb{R}$,

(a) $\|\mathbf{v}\| \geq 0$ for all $\mathbf{v} \in V$ and $\|\mathbf{v}\| = 0$ if and only if $\mathbf{v} = \mathbf{0}$

(b) $\|\mathbf{v} + \mathbf{w}\| \leq \|\mathbf{v}\| + \|\mathbf{w}\|$

(c) $\|c\mathbf{v}\| = |c|\|\mathbf{v}\|$

Next, we give definitions for some of the most common norms.

Definition 4 *For any $p \geq 1$, the ℓ_p **norm** of a vector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ in $\mathbb{R}^n$ is defined by*

$$\|\mathbf{v}\|_p = \sqrt[p]{\sum_{i=1}^{n} |v_i|^p}$$

For $p = 1$ we have the ℓ_1 norm,

$$\|\mathbf{v}\|_1 = \sum_{i=1}^{n} |v_i|$$

and as $p \to \infty$ we have the ℓ_∞ norm,

$$\|\mathbf{v}\|_\infty = \max_{1 \leq i \leq n} |v_i|$$

Definition 5 *The length, or **Euclidean norm**, of $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is the quantity*

$$\|\mathbf{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

*and the **distance** between two vectors is denoted by*

$$dist(\mathbf{v}, \mathbf{w}) = \|\mathbf{v} - \mathbf{w}\|_2$$

*A vector of norm 1 is called a **unit vector**.*

The Euclidean norm is also frequently referred to as the ℓ_2 norm. Unless stated otherwise the norm $\|\cdot\|$ will be the euclidean norm. Notice that the dot products and norms are related by the formula

$$\mathbf{v} \cdot \mathbf{v} = \|\mathbf{v}\|^2$$

Since we will be working with the **Euclidean space** $\mathbb{R}^n$, the inner product will be the same as the dot product.

Definition 6 *The distance function is extended to sets as*

$$dist(\mathbf{v}, \mathcal{S}) = dist(\mathcal{S}, \mathbf{v}) = \min_{\mathbf{s} \in \mathcal{S}} \{dist(\mathbf{v}, \mathbf{s})\}$$

Definition 7 *The **standard basis** (natural basis) for the n-dimensional Euclidean space $\mathbb{R}^n$ consists of n distinct vectors*

$$\{\mathbf{e}_i \, : \, 1 \leq i \leq n\}$$

where $\mathbf{e}_i$ denotes the vector with a 1 in the $i-th$ coordinate and 0's everywhere else, e.g., for $n = 4$, $\mathbf{e}_2 = (0, 1, 0, 0)^\intercal$.

Definition 8 *The* ***angle*** θ *between nonzero vectors* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ *is given by*

$$\mathbf{v} \cdot \mathbf{w} = \|\mathbf{v}\|\|\mathbf{w}\| cos\theta, \qquad cos\theta = \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\|\|\mathbf{w}\|}, \qquad \theta = arccos\left(\frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\|\|\mathbf{w}\|}\right)$$

Lemma 1.2 *Two vectors* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ *are* ***orthogonal*** *to one another if and only if* $\mathbf{v} \cdot \mathbf{w} = 0$

Proof. The cosine is 0 if and only if the angle θ is and odd multiple of $\frac{\pi}{2}$. □

Lemma 1.3 ***(Cauchy-Schwarz inequality)***
For any two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$*,*

$$|\mathbf{v} \cdot \mathbf{w}| \leq \|\mathbf{v}\|\|\mathbf{w}\|$$

Proof. We observe that the Cauchy-Schwarz inequality follows immediately from Lemma 1.2, but we will present a direct proof. If $\mathbf{w} = \mathbf{0}$, there is nothing to prove, so we may assume that $\mathbf{w} \neq \mathbf{0}$. We consider the function

$$\begin{aligned} f(t) = \|\mathbf{v} - t\mathbf{w}\|^2 &= (\mathbf{v} - t\mathbf{w}) \cdot (\mathbf{v} - t\mathbf{w}) \\ &= \mathbf{v} \cdot \mathbf{v} - 2t\mathbf{v} \cdot \mathbf{w} + t^2 \mathbf{w} \cdot \mathbf{w} \\ &= \|\mathbf{v}\|^2 - 2t\mathbf{v} \cdot \mathbf{w} + t^2\|\mathbf{w}\|^2 \end{aligned}$$

We know that $f(t) \geq 0$ for all $t \in \mathbb{R}$, so we choose the value of t that minimizes $f(t)$ and see what it gives. This minimizing value is $t = \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{w}\|^2}$. Hence

$$0 \leq f\left(\frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{w}\|^2}\right) = \|\mathbf{v}\|^2 - \frac{(\mathbf{v} \cdot \mathbf{w})^2}{\|\mathbf{w}\|^2}$$

Simplifying this expressions and taking square roots gives the desired result. □

Definition 9 *Given vectors* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ *with* $\mathbf{w} \neq \mathbf{0}$*, we write* $\mathbf{u}, \mathbf{u}_\perp$ *for the* ***projections*** *of* $\mathbf{v}$ *parallel and orthogonal to* $\mathbf{w}$*, respectively*

$$\mathbf{u} = \left(\frac{\mathbf{v} \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}}\right), \qquad \mathbf{u}_\perp = \mathbf{v} - \left(\frac{\mathbf{v} \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}}\right)$$

Definition 10 *An* ***orthogonal basis*** *for a vector space* V *is a basis* $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ *with the property that*

$$\mathbf{v}_i \cdot \mathbf{v}_j = 0 \quad \textit{for all} \quad i \neq j$$

A basis is ***orthonormal*** if in addition, $\|\mathbf{v}_i\| = 1$ for all i.

There are many formulas that become much simpler using an orthogonal or orthonormal basis. In particular, if $\mathbf{v}_1, \dots, \mathbf{v}_n$ is an orthogonal basis and if $\mathbf{v} = a_1\mathbf{v}_1, \dots, a_n\mathbf{v}_n$ is a linear combination of the basis vectors, then

$$\begin{aligned}
\|\mathbf{v}\|^2 &= \|a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n\|^2 \\
&= (a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n) \cdot (a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j (\mathbf{v}_i \cdot \mathbf{v}_j) \\
&= \sum_{i=1}^{n} a_i^2 \|\mathbf{v}_i\|^2 \quad \text{since} \quad \mathbf{v}_i \cdot \mathbf{v}_j \quad \text{for} \quad i \neq j
\end{aligned}$$

If the basis is orthonormal, then this further simplifies to $\|\mathbf{v}\| = \sum a_i^2$.

Definition 11 *For any $\mathbf{c} \in \mathbb{R}^n$ and any $r > 0$, the **open ball** of radius r centered at $\mathbf{c}$ is the set*

$$\mathbb{B}(\mathbf{c}, r) = \{\mathbf{v} \in \mathbb{R}^n : \|\mathbf{v} - \mathbf{c}\| < r\}$$

Definition 12 *For any $\mathbf{c} \in \mathbb{R}^n$ and any $r > 0$, the **closed ball** of radius r centered at $\mathbf{c}$ is the set*

$$\overline{\mathbb{B}}(\mathbf{c}, r) = \{\mathbf{v} \in \mathbb{R}^n : \|\mathbf{v} - \mathbf{c}\| \leq r\}$$

1.2 Lattices in $\mathbb{R}^m$

Definition 13 *Let $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ be a set of linearly independent vectors. The **lattice** $\mathcal{L}$ generated by $\mathbf{b}_1, \ldots, \mathbf{b}_n$ ($n \leq m$) is the set of linear combinations of $\mathbf{b}_1, \ldots, \mathbf{b}_n$ with coefficients in $\mathbb{Z}$.*

$$\mathcal{L} = \{a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \cdots + a_n\mathbf{b}_n \,:\, a_1, a_2, \ldots, a_n \in \mathbb{Z}\}$$

Equivalently, if we define $\mathbf{B}$ as the $m \times n$ matrix whose columns are the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ then the lattice generated by $\mathbf{B}$ is

$$\mathcal{L}(\mathbf{B}) = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \{\mathbf{Bz} \,:\, \mathbf{z} \in \mathbb{Z}^n\}$$

We say that the ***dimension*** of the lattice is m. A ***basis*** for $\mathcal{L}$ is any set of linearly independent vectors that generates $\mathcal{L}$. Any two such sets have the same number of elements.

The ***rank*** of $\mathcal{L}$ is the number of vectors in a basis for $\mathcal{L}$ which is n in Definition 13. If the *lattice rank* is equal to the *lattice dimension* then the lattice is called a ***full-rank lattice*** . The basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are said to ***generate*** the lattice. Next we define the span and the sublattice of a lattice.

Definition 14 *The **span** of a lattice $span(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n))$ is the linear space generated by its vectors.*

$$span(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)) = span(\mathcal{L}(\mathbf{B})) = span(\mathbf{B}) = \{\mathbf{Bx} \,:\, \mathbf{x} \in \mathbb{R}^n\}$$

Definition 15 *Let $\mathcal{L}(\mathbf{B}) \subset \mathbb{R}^m$ be a lattice with basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$. Suppose that $\mathbf{b}'_1, \ldots, \mathbf{b}'_n \in \mathcal{L}$ are linearly independent, and let $\mathcal{L}(\mathbf{B}')$ be the lattice generated by $\mathbf{b}'_1, \ldots, \mathbf{b}'_n$. We call $\mathcal{L}(\mathbf{B}')$ a **sublattice** of $\mathcal{L}(\mathbf{B})$ and write $\mathcal{L}(\mathbf{B}') \subseteq \mathcal{L}(\mathbf{B})$. If $\mathcal{L}(\mathbf{B}') = \mathcal{L}(\mathbf{B})$ we say that the basis $\mathbf{B}$ and $\mathbf{B}'$ are **equivalent**. If $\mathcal{L}(\mathbf{B}') \subseteq \mathcal{L}(\mathbf{B})$, but $\mathcal{L}(\mathbf{B}') \neq \mathcal{L}(\mathbf{B})$ then basis $\mathbf{B}$ and $\mathbf{B}'$ are not equivalent, $\mathcal{L}(\mathbf{B}')$ is a **proper sublattice** of $\mathcal{L}(\mathbf{B})$ and write $\mathcal{L}(\mathbf{B}') \subset \mathcal{L}(\mathbf{B})$.*

There is an alternative, more abstract, way to define lattices. A subset $\mathcal{S}$ of $\mathbb{R}^m$ is an ***additive subgroup*** if it is closed under addition and subtraction. It is called a ***discrete additive subgroup*** if there is a positive constant $\epsilon > 0$ such that for every $\mathbf{v} \in \mathcal{S}$,

$$\mathcal{S} \cap \{\mathbf{w} \in \mathbb{R}^m : \|\mathbf{v} - \mathbf{w}\| < \epsilon\} = \{\mathbf{v}\}$$

or equivalently,

$$\exists \epsilon > 0 \text{ such that}, \forall \mathbf{x} \neq \mathbf{y} \in \mathcal{S}, \|\mathbf{x} - \mathbf{y}\| \geq \epsilon$$

Definition 16 *A lattice $\mathcal{L}$ is a discrete additive subgroup of $\mathbb{R}^n$.*

In other words if we take any vector $\mathbf{v} \in \mathcal{L}$ and draw a closed ball of radius ϵ around $\mathbf{v}$, then there is no other points of $\mathcal{L}$ inside the ball.

Proposition 1.4 *Any two bases for a lattice $\mathcal{L}$ are related by a matrix having integer coefficients and a determinant equal to ± 1.*

Proof. Suppose that the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are a basis for a lattice $\mathcal{L}$ and that $\mathbf{w}_1, \ldots, \mathbf{w}_n$ is another collection of vectors in $\mathcal{L}$. We can write each $\mathbf{w}_j$ as a linear combination of the basis vectors,

$$\begin{aligned} \mathbf{w}_1 &= a_{11}\mathbf{b}_1 + a_{12}\mathbf{b}_2 + \cdots + a_{1n}\mathbf{b}_n, \\ \mathbf{w}_2 &= a_{21}\mathbf{b}_1 + a_{22}\mathbf{b}_2 + \cdots + a_{2n}\mathbf{b}_n, \\ &\vdots \\ \mathbf{w}_n &= a_{n1}\mathbf{b}_1 + a_{n2}\mathbf{b}_2 + \cdots + a_{nn}\mathbf{b}_n, \end{aligned}$$

but since we are dealing with lattices, we know that all of the a_{ij} coefficients are integers.

Suppose that we try to express the $\mathbf{v}_i$ in terms of the $\mathbf{w}_j$. This involves inverting the matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

We need the $\mathbf{b}_i$ to be linear combinations of $\mathbf{w}_j$ using *integer* coefficients, so we need the entries of $\mathbf{A}^{-1}$ to have integer entries. Hence,

$$1 = det(\mathbf{I}) = det(\mathbf{A}\mathbf{A}^{-1}) = det(\mathbf{A})det(\mathbf{A}^{-1})$$

where $det(\mathbf{A})$ and $det(\mathbf{A}^{-1})$ are integers, so we must have $det(\mathbf{A}) = \pm 1$. Conversely, if $det(\mathbf{A}) = \pm 1$, the property $\mathbf{A}^{-1} = \dfrac{Adj(\mathbf{A})}{det(\mathbf{A})}$ tells us that since the adjugate matrix $Adj(\mathbf{A})$ of $\mathbf{A}$ is an integer matrix because $\mathbb{Z}$ is closed under multiplication and addition, then $\mathbf{A}^{-1}$ does indeed have integer entries. □

Definition 17 *An $n \times n$ matrix* **U** *with integer coefficients and determinant ± 1 will be called* ***unimodular***.

It follows from Proposition 1.4 and Definition 17 that $\mathbf{U}^{-1}$ is defined and is also unimodular.

Definition 18 *A* ***unimodular column operation*** *on a matrix is one of the following elementary operations:*

- multiply any column by -1
- interchange any two columns
- add an integral multiply of any column to any other column

To generate examples of $n \times n$ unimodular matrices, we start with the identity matrix $\mathbf{I}_n$, and the apply any finite sequence of unimodular column operations. The result will be a $n \times n$ unimodular matrix and in fact any such matrix can be obtained in this way.

If we apply unimodular column operations to a matrix whose columns contain a basis for a lattice $\mathcal{L}$, then we obtain another basis for the same lattice.

For computational purposes, it is often convenient to work with lattices whose vectors have integer coordinates. For example,

$$\mathbb{Z}^n = \{(b_1, b_2, \dots, b_n) \,:\, b_1, b_2, \dots, b_n \in \mathbb{Z}\}$$

is the lattice consisting of all vectors with integer coordinates.

Definition 19 *The* ***determinant*** *of a lattice $\mathcal{L}(\mathbf{B})$ is*

$$det(\mathcal{L}(\mathbf{B})) = \sqrt{det(\mathbf{B}^\intercal \mathbf{B})}$$

In the special case that $\mathcal{L}(\mathbf{B})$ is a full-rank lattice we have

$$det(\mathcal{L}(\mathbf{B})) = |det(\mathbf{B})|$$

Lemma 1.5 *The determinant of a lattice does not depend on the basis.*

Proof. Suppose the lattice $\mathcal{L} \subset \mathbb{R}^n$ has two bases $\mathbf{B}_1, \mathbf{B}_2$. Then by Proposition 1.4, $\mathbf{B}_2 = \mathbf{B}_1\mathbf{U}$ and the properties $(\mathbf{AB})^\intercal = \mathbf{B}^\intercal \mathbf{A}^\intercal$, $det(\mathbf{AB}) = det(\mathbf{A})det(\mathbf{B})$ and $det(\mathbf{A}^\intercal) = det(\mathbf{A})$ we have that

$$\sqrt{det(\mathbf{B}_2^\intercal \mathbf{B}_2)} = \sqrt{det(\mathbf{U}^\intercal \mathbf{B}_1^\intercal \mathbf{B}_1 \mathbf{U})} = \sqrt{det^2(\mathbf{U}) det(\mathbf{B}_1^\intercal \mathbf{B}_1)} = \sqrt{det(\mathbf{B}_1^\intercal \mathbf{B}_1)}$$

More simple, in the case of a full-rank lattice

$$|det(\mathbf{B}_2)| = |det(\mathbf{B}_1\mathbf{U})| = |det(\mathbf{B}_1)||det(\mathbf{U})| = |det(\mathbf{B}_1)||\pm 1| = |det(\mathbf{B}_1)|$$

Since the two bases are arbitrary, this completes the proof. $\square$

Definition 20 *Let $\mathcal{L} \subset \mathbb{R}^n$ be a lattice and let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ be a basis for $\mathcal{L}$. The **fundamental parallelepiped** for $\mathcal{L}$ is the set*

$$\mathcal{P}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{t_1\mathbf{b}_1 + t_2\mathbf{b}_2 + \cdots + t_n\mathbf{b}_n \,:\, 0 \leq t_i < 1\}$$

Thus, pictorially, a fundamental parallelepiped is the half-open region enclosed by the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$. Clearly, different bases of the same lattice generate different fundamental parallelepipeds.

Proposition 1.6 *Let $\mathcal{L} \subset \mathbb{R}^n$ be a full-rank lattice and $\mathcal{P}$ be a fundamental parallelepiped for $\mathcal{L}$. Then every vector $\mathbf{v} \in \mathbb{R}^n$ can be written in the form*

$$\mathbf{v} = \mathbf{x} + \mathbf{y}$$

for a unique $\mathbf{x} \in \mathcal{P}$ and a unique $\mathbf{y} \in \mathcal{L}$.

Equivalently, the union of the translated fundamental parallelepipeds

$$\mathcal{P} + \mathbf{y} = \{\mathbf{x} + \mathbf{y} : \mathbf{x} \in \mathcal{P}\}$$

as $\mathbf{y}$ ranges over the vectors in lattice $\mathcal{L}$ exactly covers $\mathbb{R}^n$.

Proof. Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be basis of $\mathcal{L}$ that gives the fundamental parallelepiped $\mathcal{P}$. Then $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are linearly independent in $\mathbb{R}^n$, so they form a basis of $\mathbb{R}^n$ and then any vector $\mathbf{v}$ of $\mathbb{R}^n$ can be written as

$$\mathbf{v} = a_1\mathbf{b}_1 + \cdots + a_n\mathbf{b}_n$$

for some unique choice of $a_1, \ldots, a_n \in \mathbb{R}$

We write each a_i as its integer and fractional part

$$a_i = r_i + t_i, \quad \text{with } r_i \in \mathbb{Z} \text{ and } 0 \leq t_i < 1$$

therefore $\mathbf{v}$ can be written in the desired form as

$$\mathbf{v} = (r_1 + t_1)\mathbf{b}_1 + \cdots + (r_n + t_n)\mathbf{b}_n = \overbrace{(r_1\mathbf{b}_1 + \cdots + r_n\mathbf{b}_n)}^{\mathbf{x} \in \mathcal{P}} + \overbrace{(t_1\mathbf{b}_1 + \cdots + t_n\mathbf{b}_n)}^{\mathbf{y} \in \mathcal{L}}$$

Now suppose that $\mathbf{v}$ can be written as a sum of two different representations,

$$\begin{aligned}
\mathbf{v} &= \mathbf{x} + \mathbf{y} = \mathbf{x}' + \mathbf{y}' \\
&= (r_1\mathbf{b}_1 + \cdots + r_n\mathbf{b}_n) + (t_1\mathbf{b}_1 + \cdots + t_n\mathbf{b}_n) \\
&= (r'_1\mathbf{b}_1 + \cdots + r'_n\mathbf{b}_n) + (t'_1\mathbf{b}_1 + \cdots + t'_n\mathbf{b}_n)
\end{aligned}$$

where $r_i, r_i' \in \mathbb{Z}$ and $0 \le t_i, t_i' < 1$. Because $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are linearly independent we get that for all $i = 1, \ldots, n$

$$r_i + t_i = r_i' + t_i'$$
$$t_i' - t_i = r_i' - r_i \in \mathbb{Z}$$

Since $0 \le t_i, t_i' < 1$ and $t_i' - t_i \in \mathbb{Z}$ it must be the case that $t_i' - t_i = 0 \Rightarrow t_i' = t_i$ and thus $r_i' - r_i = 0 \Rightarrow r_i' = r_i$. From that we conclude that $\mathbf{x}' = \mathbf{x}$ and $\mathbf{y}' = \mathbf{y}$, and this completes the proof. □

Proposition 1.7 *Let $\mathcal{L} \subset \mathbb{R}^n$ be a full-rank lattice, let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a basis for $\mathcal{L}$, and let $\mathcal{P} = \mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be the associated fundamental parallelepiped. We write the basis of $\mathcal{L}$ in square matrix form as*

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{pmatrix}$$

*where each $\mathbf{b}_i$ is the $i-th$ row of the matrix $\mathbf{B}$. Then the **volume** of $\mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is given by the formula*

$$Vol(\mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)) = |det(\mathbf{B})|$$

Proof. We can compute the volume of $\mathcal{P}$ as the integral of the constant function 1 over the region $\mathcal{P}$,

$$Vol(\mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)) = \int_{\mathcal{P}} 1\, dx_1 dx_2 \cdots dx_n$$

We make a change of variables from $\mathbf{x} = (x_1, \ldots, x_n)$ to $\mathbf{t} = (t_1, \ldots, t_n)$ according to the formula

$$(x_1, \ldots, x_n) = t_1\mathbf{b}_1 + \cdots + t_n\mathbf{b}_n \Longleftrightarrow \mathbf{x} = \mathbf{tB} \quad \text{(matrix form)}$$

The Jacobian matrix of this change of variables is $\mathbf{B}$ and the fundamental parallelepiped $\mathcal{P}$ is the image under $\mathbf{B}$ of the unit cube $C_n = [0,1]^n$, so the change of variables formula for integrals yields

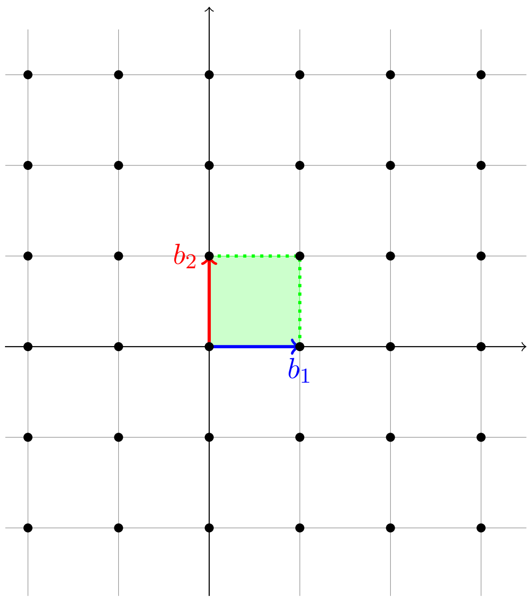
$$\begin{aligned} Vol(\mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)) &= \int_{\mathcal{P}} 1\, dx_1 \cdots dx_n = \int_{\mathbf{B}C_n} 1\, dx_1 \cdots dx_n = \int_{C_n} |det(\mathbf{B})|\, dt_1 \cdots dt_n \\ &= |det(\mathbf{B})|\, Vol(C_n) \\ &= |det(\mathbf{B})| \end{aligned}$$

□

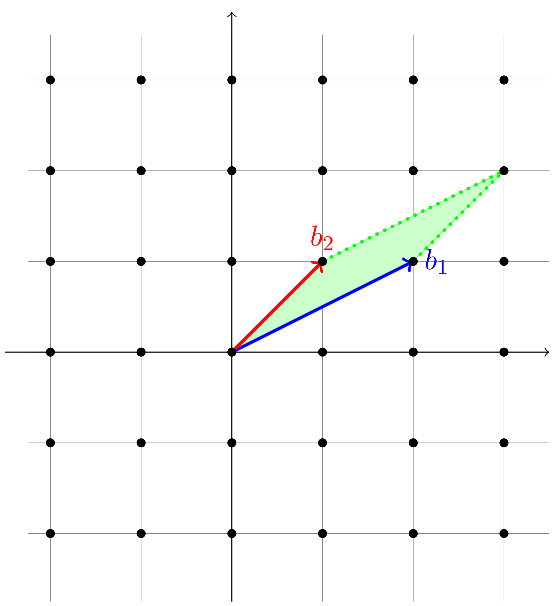
It is easy to see that if we take each $\mathbf{b}_i$ to be the $i - th$ column of a matrix $\mathbf{B}'$ we get

$$|det(\mathbf{B}')| = |det(\mathbf{B}^{\mathsf{T}})| = |det(\mathbf{B})| = Vol(\mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n))$$

and from Lemma 1.5 we get that the volume of the fundamental parallelepiped does not depend on the basis.



(a) The lattice $\mathbb{Z}^2$ with basis vectors $(0, 1)$ and $(1, 0)$ and the associated fundamental parallelepiped.



(b) The lattice $\mathbb{Z}^2$ with a different basis consisting of vectors $(1, 1)$ and $(2, 1)$, and the associated fundamental parallelepiped.

Figure 1.1. Parallelepipeds for various bases of the lattice $\mathbb{Z}^2$. Note that the parallelepipeds in either case do not contain any nonzero lattice point.

Theorem 1.8 *Let $\mathcal{L} \subset \mathbb{R}^n$ be a full-rank lattice, and let $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$ denote linearly independent vectors in $\mathcal{L}$. Then, $\mathbf{b}_1, \ldots, \mathbf{b}_n$ form a basis $\mathcal{L}$ if and only if $\mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \cap \mathcal{L} = \{\mathbf{0}\}$*

Proof. Assume first that $\mathbf{b}_1, \ldots, \mathbf{b}_n$ form a basis of $\mathcal{L}$. Let

$$\mathbf{v} = \sum_{i=1}^{n} t_i \mathbf{b}_i \in \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \cap \mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$$

Since $\mathbf{v} \in \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ then $t_i \in \mathbb{Z}, \forall i$. Since $\mathbf{v} \in \mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ then $t_i \in [0,1), \forall i$. But only 0 is an integer in $[0,1)$ and that means that $t_i = 0\,, \forall i$ so we get that $\mathbf{v} = \mathbf{0}$.

For the other direction assume that $\mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \cap \mathcal{L} = \{\mathbf{0}\}$. The vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are linearly independent and since they belong to $\mathcal{L}$ we have that $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \subseteq \mathcal{L}$. It suffices to show that $\mathcal{L} \subseteq \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$. Since $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathcal{L}$ are linearly independent we can write any lattice vector $\mathbf{v}$ as

$$\mathbf{v} = \sum_{i=1}^{n} t_i \mathbf{b}_i \quad \text{where } t_i \in \mathbb{R}$$

Consider now the vector

$$\mathbf{v}' = \sum_{i=1}^{n} \lfloor t_i \rfloor \mathbf{b}_i$$

where $\lfloor t_i \rfloor$ denotes the integer part of t_i. The vector $\mathbf{v}'$ is in the lattice $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ since the coefficients $\lfloor t_i \rfloor$ are integers. Therefore, the vector $\mathbf{v} - \mathbf{v}'$ is in $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ as well. Now the vector,

$$\mathbf{v} - \mathbf{v}' = \sum_{i=1}^{n} (t_i - \lfloor t_i \rfloor) \mathbf{b}_i$$

is in $\mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ since $0 \le t_i - \lfloor t_i \rfloor < 1$, $\forall i$.
Since $\mathbf{v} - \mathbf{v}' \in \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \cap \mathcal{P}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$, it must the case that $\mathbf{v} - \mathbf{v}' = \mathbf{0}$ by assumption. But since the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are linearly independent, this means that $t_i - \lfloor t_i \rfloor = 0$, $\forall i$ from which we get that $t_i \in \mathbb{Z}$, $\forall i$.

Thus, $\mathbf{v} \in \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ and therefore $\mathcal{L} \subseteq \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n)$. $\square$

1.3 Gram-Schmidt Orthogonalization

The "best" basis we can have for a vector space is an orthogonal basis. That is because we can most easily find the coefficients that are needed to express a vector as a linear combination of the basis vectors.

But usually we are not given an orthogonal basis. We will show how to find an orthogonal basis starting from an arbitrary basis.

Definition 21 *Let* $\mathbf{b}_1, \ldots, \mathbf{b}_n$ *be a basis of* $\mathbb{R}^n$. *The* ***Gram-Schmidt orthogonalization*** *of* $\mathbf{b}_1, \ldots, \mathbf{b}_n$ *is the following basis* $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$

$$\tilde{\mathbf{b}}_1 = \mathbf{b}_1$$

$$\tilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{ij} \tilde{\mathbf{b}}_j \quad (2 \leq i \leq n), \qquad \mu_{ij} = \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle} \quad (1 \leq j < i \leq n)$$

We do not normalize the vectors. It is important to note that usually the Gram-Schmidt basis vectors $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ are not in the lattice generated by $\mathbf{b}_1, \ldots, \mathbf{b}_n$ because in general the vectors $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ are not integral linear combinations of $\mathbf{b}_1, \ldots, \mathbf{b}_n$.

If $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ are linearly independent in $\mathbb{R}^m$ the running time complexity of Gram-Schmidt orthogonalization is $O(mn^2)$ and thus polynomial in the input size (see ch. 2.1, p. 30, for asymptotic notation).

If we set $\mu_{ii} = 1$ for $1 \leq i \leq n$ then we have

$$\tilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{ij} \tilde{\mathbf{b}}_j \Rightarrow \mathbf{b}_i = \sum_{j=1}^{i} \mu_{ij} \tilde{\mathbf{b}}_j$$

Let $\tilde{\mathbf{b}}_1 / \|\tilde{\mathbf{b}}_1\|, \ldots, \tilde{\mathbf{b}}_n / \|\tilde{\mathbf{b}}_n\|$ denote the unit vectors in the direction of the Gram-Schmidt vectors.

Then the Gram-Schmidt orthogonalization process can be written in matrix form as

$$\overbrace{\begin{pmatrix} | & & | \\ \mathbf{b}_1 & \dots & \mathbf{b}_n \\ | & & | \end{pmatrix}}^{\mathbf{B}} = \overbrace{\begin{pmatrix} | & & | \\ \tilde{\mathbf{b}}_1 & \dots & \tilde{\mathbf{b}}_n \\ | & & | \end{pmatrix}}^{\tilde{\mathbf{B}}} \cdot \overbrace{\begin{pmatrix} 1 & \mu_{2,1} & \mu_{3,1} & \dots & \mu_{n,1} \\ 0 & 1 & \mu_{3,2} & \dots & \mu_{n,2} \\ 0 & 0 & 1 & \dots & \mu_{n,3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}}^{\mathbf{G}}$$

$$= \underbrace{\begin{pmatrix} | & & | \\ \frac{\tilde{\mathbf{b}}_1}{\|\tilde{\mathbf{b}}_1\|} & \dots & \frac{\tilde{\mathbf{b}}_n}{\|\tilde{\mathbf{b}}_n\|} \\ | & & | \end{pmatrix}}_{\tilde{\mathbf{B}}'} \cdot \underbrace{\begin{pmatrix} \|\tilde{\mathbf{b}}_1\| & \mu_{2,1}\|\tilde{\mathbf{b}}_1\| & \mu_{3,1}\|\tilde{\mathbf{b}}_1\| & \dots & \mu_{n,1}\|\tilde{\mathbf{b}}_1\| \\ 0 & \|\tilde{\mathbf{b}}_2\| & \mu_{3,2}\|\tilde{\mathbf{b}}_2\| & \dots & \mu_{n,2}\|\tilde{\mathbf{b}}_2\| \\ 0 & 0 & \|\tilde{\mathbf{b}}_3\| & \dots & \mu_{n,3}\|\tilde{\mathbf{b}}_3\| \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \|\tilde{\mathbf{b}}_n\| \end{pmatrix}}_{\mathbf{G}'}$$

where each $\mathbf{b}_i$ (resp. $\tilde{\mathbf{b}}_i$) is the $i-th$ column of the matrix $\mathbf{B}$ (resp. $\tilde{\mathbf{B}}$).

Remark 1.9 *We can write matrix $\mathbf{B}$ as $\mathbf{B} = \tilde{\mathbf{B}} \cdot \mathbf{G}$ where $\mathbf{G}$ is an upper triangular matrix with diagonal entries $g_{ii} = 1$ for $1 \leq i \leq n$ (therefore its determinant equals to 1) then*

$$det(\mathbf{B}) = det(\tilde{\mathbf{B}} \cdot \mathbf{G}) = det(\tilde{\mathbf{B}})det(\mathbf{G}) = det(\tilde{\mathbf{B}}) \cdot 1 = det(\tilde{\mathbf{B}})$$

Remark 1.10 *We can also write matrix $\mathbf{B}$ as $\mathbf{B} = \tilde{\mathbf{B}}' \cdot \mathbf{G}'$ where $\mathbf{G}'$ is a lower triangular matrix with diagonal entries $g'_{ii} = \|\tilde{\mathbf{b}}_i\|$ for $1 \leq i \leq n$ therefore its determinant equals to $\prod_{i=1}^n \|\tilde{\mathbf{b}}_i\|$. Since the vectors $\frac{\mathbf{b}_i}{\|\mathbf{b}_i\|}$ are orthonormal, the determinant of the matrix $\tilde{\mathbf{B}}'$ with columns $\frac{\mathbf{b}_i}{\|\mathbf{b}_i\|}$ is ± 1. Thus, we have*

$$det(\mathcal{L}(\mathbf{B})) = |det(\tilde{\mathbf{B}}' \cdot \mathbf{G}')| = |det(\tilde{\mathbf{B}}')||det(\mathbf{G}')| = |\pm 1| \prod_{i=1}^{n} \|\tilde{\mathbf{b}}_i\| = \prod_{i=1}^{n} \|\tilde{\mathbf{b}}_i\|$$

Theorem 1.11 *Let $\mathbf{b}_1, \dots, \mathbf{b}_n$ be a basis for $\mathbb{R}^n$ and let $\tilde{\mathbf{b}}_1, \dots, \tilde{\mathbf{b}}_n$ be its Gram-Schmidt orthogonalization. We have:*

(a) $\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_j \rangle = 0$ for $1 \leq i < j \leq n$

(b) $span(\mathbf{b}_1, \dots, \mathbf{b}_k) = span(\tilde{\mathbf{b}}_1, \dots, \tilde{\mathbf{b}}_k)$ for $1 \leq k \leq n$

(c) For $1 \leq k \leq n$, the vector $\tilde{\mathbf{b}}_k$ is the projection of $\mathbf{b}_k$ onto the orthogonal complement of $span(\mathbf{b}_1, \dots, \mathbf{b}_{k-1})$

(d) $\|\tilde{\mathbf{b}}_k\| \leq \|\mathbf{b}_k\|$ for $1 \leq k \leq n$

Proof.

(a) Induction on j. For $j = 1$ there is nothing to prove. Assume that the claim holds for some $j \geq 1$. For $1 \leq i < j + 1$ we have

$$\begin{aligned}
\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_j \rangle &= \left\langle \tilde{\mathbf{b}}_i, \left(\mathbf{b}_{j+1} - \sum_{k=1}^{j} \mu_{j+1,k} \tilde{\mathbf{b}}_k \right) \right\rangle \\
&= \langle \tilde{\mathbf{b}}_i, \mathbf{b}_{j+1} \rangle - \sum_{k=1}^{j} \mu_{j+1,k} \langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_k \rangle \\
&= \langle \tilde{\mathbf{b}}_i, \mathbf{b}_{j+1} \rangle - \mu_{j+1,i} \langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_i \rangle \\
&= \langle \tilde{\mathbf{b}}_i, \mathbf{b}_{j+1} \rangle - \frac{\langle \mathbf{b}_{j+1}, \tilde{\mathbf{b}}_i \rangle}{\langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_i \rangle} \langle \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_i \rangle \\
&= 0
\end{aligned}$$

(b) By Remark 1.9 we have $\mathbf{b}_i \in span(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_k)$ for $1 \leq i \leq k$ hence

$$span(\mathbf{b}_1, \ldots, \mathbf{b}_k) \subseteq span(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_k)$$

For the reverse inclusion we use induction on k. For $k = 1$ we have $\mathbf{b}_1 = \tilde{\mathbf{b}}_1$ and so the claim is obvious. Assume that the claim holds for some $k \geq 1$. We have

$$\tilde{\mathbf{b}}_{k+1} = \mathbf{b}_{k+1} - \sum_{j=1}^{k} \mu_{k+1,j} \tilde{\mathbf{b}}_j = \mathbf{b}_{k+1} + \mathbf{v}, \quad \mathbf{v} \in span(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_k)$$

The induction hypothesis gives $span(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_k) \subseteq span(\mathbf{b}_1, \ldots, \mathbf{b}_k)$, and so the last equation implies $\tilde{\mathbf{b}}_{k+1} \in span(\mathbf{b}_1, \ldots, \mathbf{b}_k)$. Therefore

$$span(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_k) \subseteq span(\mathbf{b}_1, \ldots, \mathbf{b}_k)$$

(c) We write $S = span(\mathbf{b}_1, \ldots, \mathbf{b}_{k-1})$ and $S^{\perp}$ for the subspace of $\mathbb{R}^n$ consisting of all vectors $\mathbf{b}'$ such that $\langle \mathbf{b}', \mathbf{b} \rangle = 0, \ \forall \, \mathbf{b} \in S$. There is a unique decomposition $\mathbf{b}_k = \mathbf{b}'_k + \mathbf{s}$ where $\mathbf{b}'_k \in S^{\perp}$ and $\mathbf{s} \in S$. Here $\mathbf{b}'_k$ is the projection of $\mathbf{b}_k$ on the orthogonal complement of S. By Remark 1.9 we have

$$\mathbf{b}_k = \tilde{\mathbf{b}}_k + \sum_{j=1}^{k-1} \mu_{kj} \tilde{\mathbf{b}}_j$$

From part (b), we have $S = span(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_{k-1})$, and so $\tilde{\mathbf{b}}_k = \mathbf{b}'_k$.

(d) Again by Remark 1.9 we have

$$\mathbf{b}_k = \tilde{\mathbf{b}}_k + \sum_{j=1}^{k-1} \mu_{kj} \tilde{\mathbf{b}}_j \Rightarrow \|\mathbf{b}_k\|^2 = \|\tilde{\mathbf{b}}_k + \sum_{j=1}^{k-1} \mu_{kj} \tilde{\mathbf{b}}_j\|^2 = \left\langle \tilde{\mathbf{b}}_k + \sum_{j=1}^{k-1} \mu_{kj} \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_k + \sum_{j=1}^{k-1} \mu_{kj} \tilde{\mathbf{b}}_j \right\rangle$$

Part (a) ($\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ are orthogonal) implies that

$$\left\langle \tilde{\mathbf{b}}_k + \sum_{j=1}^{k-1} \mu_{kj} \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_k + \sum_{j=1}^{k-1} \mu_{kj} \tilde{\mathbf{b}}_j \right\rangle = \|\tilde{\mathbf{b}}_k\|^2 + \sum_{j=1}^{k-1} \mu_{kj}^2 \|\tilde{\mathbf{b}}_j\|^2$$

Therefore

$$\|\mathbf{b}_k\|^2 = \|\tilde{\mathbf{b}}_k\|^2 + \sum_{j=1}^{k-1} \mu_{kj}^2 \|\tilde{\mathbf{b}}_j\|^2$$

Since every term in the sum is nonnegative, this proves the claim.

□

Corollary 1.12 (*Hadamard's inequality*) *Let* $\mathbf{B} \in \mathbb{R}^{n \times n}$ *with vectors* $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ *as columns (or rows). Then,*

$$det(\mathbf{B}) \leq \|\mathbf{b}_1\| \cdot \|\mathbf{b}_2\| \cdots \|\mathbf{b}_n\|$$

Proof. From Remarks 1.9 and 1.10 we have that

$$\det(\mathbf{B}) = \prod_{i=1}^{n} \|\tilde{\mathbf{b}}_i\|$$

and from Theorem 1.11(d) the inequality follows. □

1.4 Successive minima

A basic parameter of the lattices is the length of the shortest nonzero vector in the lattice (since any lattice contains the zero vector which has a zero norm). When we speak of length, we mean the Euclidean norm. Finding the shortest nonzero lattice vector is also a fundamental computational problem associated with lattices.

This parameter is also called the ***first successive minimum*** of the lattice, and is denoted $\lambda_1(\mathcal{L})$. The ***second successive minimum*** of the lattice is the smallest real number r such that there exist two linearly independent vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{L}$ with $\|\mathbf{v}_1\|, \|\mathbf{v}_2\| \leq r$. This leads to the following generalization of the first successive minimum known as successive minima.

Definition 22 *Let $\mathcal{L}$ be a lattice of rank n. Then for every $i \in \{1, \dots, n\}$ we define the $i-th$ successive minimum as*

$$\lambda_i(\mathcal{L}) = inf\{r : dim(span(\mathcal{L} \cap \mathbb{B}(\mathbf{0}, r))) \geq i\}$$

A more descriptive definition is the following one:

Definition *Let $\mathcal{L}$ be a lattice of rank n. Then for every $i \in \{1, \dots, n\}$ we define the $i-th$ successive minimum as*

$$\lambda_i(\mathcal{L}) = inf\{r : \mathbb{B}(\mathbf{0}, r) \textit{ contains } \geq i \textit{ linearly independent lattice vectors}\}$$

It follows from the characterization of lattices as discrete subgroups of $\mathbb{R}^n$ that there always exist vectors achieving the successive minima. So, the infimum is actually a minimum if $\mathbb{B}(\mathbf{0}, r)$ is replaced with the closed ball $\overline{\mathbb{B}}(\mathbf{0}, r)$.

Theorem 1.13 *Let $\mathcal{L}$ be a lattice of rank n with successive minima $\lambda_1(\mathcal{L}), \dots, \lambda_n(\mathcal{L})$. Then there exist linearly independent lattice vectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathcal{L}$ such that $\|\mathbf{v}_i\| = \lambda_i(\mathcal{L})$ for all $i = 1, \dots, n$.*

Interestingly, the vectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ achieving the minima are not necessarily a basis for $\mathcal{L}$. It is easy to see that the successive minima are weakly increasing:

$$\lambda_1(\mathcal{L}) \leq \lambda_2(\mathcal{L}) \leq \cdots \leq \lambda_n(\mathcal{L})$$

The best possible basis for a lattice $\mathcal{L}$ of dimension n consists of vectors

$$\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n \quad \text{such that} \quad \|\mathbf{b}_i\| = \lambda_i(\mathcal{L}) \quad \text{for every } i \in \{1, 2, \dots, n\}$$

Such a basis is in general very hard compute and so we would like to know some upper and lower bounds for the successive minima. The following theorem gives a lower bound on the length of the shortest nonzero vector in a lattice.

Theorem 1.14 *Let* $\mathbf{B}$ *be basis of a lattice of rank* n, *and let* $\tilde{\mathbf{B}}$ *be its Gram-Schmidt orthogonalization . Then, the first minimum of the lattice (and therefore every nonzero lattice vector) satisfies*

$$\lambda_1(\mathcal{L}(\mathbf{B})) \geq \underset{1\leq i\leq n}{min} \|\tilde{\mathbf{b}}_i\| > 0$$

Proof. Let $\mathbf{x} \in \mathbb{Z}^n$ be any nonzero integer vector. Let $j \in \{1, \dots, n\}$ be the largest index such that $x_j \neq 0$, i.e. $x_{j+1} = \cdots = x_n = 0$. Then,

$$\begin{aligned}
|\langle \mathbf{Bx}, \tilde{\mathbf{b}}_j\rangle| &= |\langle \sum_{i=1}^{n} x_i \mathbf{b}_i, \tilde{\mathbf{b}}_j\rangle| \\
&= |\sum_{i=1}^{n} x_i \langle \mathbf{b}_i, \tilde{\mathbf{b}}_j\rangle| && \text{inner product linearity} \\
&= |x_j| \langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j\rangle && \langle \mathbf{b}_i, \tilde{\mathbf{b}}_j\rangle \overset{j\leq i}{=} 0,\ x_j \overset{j\geq i}{=} 0 \\
&= |x_j| \|\tilde{\mathbf{b}}_j\|^2 && \langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j\rangle = \|\tilde{\mathbf{b}}_j\|^2
\end{aligned}$$

On the other hand by the Cauchy-Schwarz inequality we have that

$$|\langle \mathbf{Bx}, \tilde{\mathbf{b}}_j\rangle| \leq \|\mathbf{Bx}\| \|\tilde{\mathbf{b}}_j\| \Rightarrow \|\mathbf{Bx}\| \geq \frac{|\langle \mathbf{Bx}, \tilde{\mathbf{b}}_j\rangle|}{\|\tilde{\mathbf{b}}_j\|}$$

From the equation above we get

$$\begin{aligned}
\|\mathbf{Bx}\| &\geq \frac{|\langle \mathbf{Bx}, \tilde{\mathbf{b}}_j\rangle|}{\|\tilde{\mathbf{b}}_j\|} \\
&= |x_j| \|\tilde{\mathbf{b}}_j\| \\
&\geq \|\tilde{\mathbf{b}}_j\| && \text{because } x_j \in \mathbb{Z}^* \\
&\geq \underset{1\leq i\leq n}{min} \|\tilde{\mathbf{b}}_i\|
\end{aligned}$$

Since the length of any lattice vector is at least $\underset{1\leq i\leq n}{min} \|\tilde{\mathbf{b}}_i\|$ then $\lambda_1(\mathcal{L}(\mathbf{B})) \geq \underset{1\leq i\leq n}{min} \|\tilde{\mathbf{b}}_i\|$ and because $\mathbf{b}_1, \dots, \mathbf{b}_n$ are linearly independent, this quantity is strictly positive, i.e. $\underset{1\leq i\leq n}{min} \|\tilde{\mathbf{b}}_i\| > 0$. □

Before we give upper bounds on the successive minima let us give some useful definitions for sets.

Definition 23 *Let* $\mathcal{S}$ *be a subset of* $\mathbb{R}^n$

(a) $\mathcal{S}$ is ***bounded*** if the lengths of the vectors in $\mathcal{S}$ are bounded. Equivalently, $\mathcal{S}$ is bounded if there is a radius r such that $\mathcal{S}$ is contained within the ball $\overline{\mathbb{B}}(\mathbf{0}, r)$.

(b) $\mathcal{S}$ is ***centrally symmetric (symmetric about the origin)*** if for every point $\mathbf{x}$ in $\mathcal{S}$, the negation $-\mathbf{x}$ is also in $\mathcal{S}$.

(c) $\mathcal{S}$ is ***convex*** if whenever two points $\mathbf{x}$ and $\mathbf{y}$ are in $\mathcal{S}$, then the entire line segment connecting $\mathbf{x}$ and $\mathbf{y}$ lies completely in $\mathcal{S}$, i.e.,

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{S}, \mathbf{x} \neq \mathbf{y}, \forall a \in [0,1],\ a\mathbf{x} + (1-a)\mathbf{y}\ \in \mathcal{S}$$

(d) $\mathcal{S}$ is ***closed*** if it has the following property: If $\mathbf{x} \in \mathbb{R}^n$ is a point such that every ball $\overline{\mathbb{B}}(\mathbf{x}, r)$ contains a point of $\mathcal{S}$, then $\mathbf{x}$ is in $\mathcal{S}$.

(e) For $\mathbf{x} \in \mathbb{R}^n$ we let $\mathcal{S} + \mathbf{x} = \{\mathbf{y} + \mathbf{x} : \mathbf{y} \in \mathcal{S}\}$ denote the ***translate*** of $\mathcal{S}$ by $\mathbf{x}$.

(f) For $a \in \mathbb{R}$ we let $a\mathcal{S} = \{a\mathbf{y} : \mathbf{y} \in \mathcal{S}\}$ denote the ***scaling*** of $\mathcal{S}$ by a.

Theorem 1.15 (*Blichfeldt theorem*) *For any lattice $\mathcal{L}(\mathbf{B})$ and for any measurable set $\mathcal{S} \subseteq span(\mathcal{L}(\mathbf{B}))$, if S has volume $vol(\mathcal{S}) > det(\mathcal{L})$, then there exist two distinct points $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{S}$ such that $\mathbf{z}_1 - \mathbf{z}_2 \in \mathcal{L}(\mathbf{B})$*

Proof. Let $\mathcal{L}(B)$ be a lattice with basis $\mathbf{B}$, and $\mathcal{S}$ be any subset of $span(\mathcal{L}(\mathbf{B}))$ such that $vol(\mathcal{S}) > det(\mathcal{L})$. Partition $\mathcal{S}$ into a collection of disjoint regions. For any lattice point $\mathbf{x}$ define the region

$$\mathcal{S}_{\mathbf{x}} = \mathcal{S} \cap (\mathcal{P}(\mathbf{B}) + \mathbf{x})$$

The sets $(\mathcal{P}(\mathbf{B}) + \mathbf{x})$ with $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ partition $span(\mathcal{L}(\mathbf{B}))$. Therefore the sets $\mathcal{S}_{\mathbf{x}}$, $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ form a partition of $\mathcal{S}$, i.e., they are pairwise disjoint and

$$\mathcal{S} = \bigcup_{\mathbf{x} \in \mathcal{L}(\mathbf{B})} \mathcal{S}_{\mathbf{x}}$$

and since $\mathcal{L}(\mathbf{B})$ is countable and set $\mathcal{S}$ is measurable from countable additivity we get,

$$vol(\mathcal{S}) = vol\left(\bigcup_{\mathbf{x} \in \mathcal{L}(\mathbf{B})} \mathcal{S}_{\mathbf{x}}\right) = \sum_{\mathbf{x} \in \mathcal{L}(\mathbf{B})} vol(\mathcal{S}_{\mathbf{x}})$$

Now define translated sets

$$\mathcal{S}'_{\mathbf{x}} = \mathcal{S}_{\mathbf{x}} - \mathbf{x} = (\mathcal{S} - \mathbf{x}) \cap (\mathcal{P}(\mathbf{B}) + \mathbf{x} - \mathbf{x}) = (\mathcal{S} - \mathbf{x}) \cap \mathcal{P}(\mathbf{B})$$

We claim that sets $\mathcal{S}'_{\mathbf{x}}$ are not pairwise disjoint. Assume, for contradiction, they are.

From the definition of set $\mathcal{S}'_{\mathbf{x}}$ it follows that for all $\mathbf{x} \in \mathcal{L}(B)$ $\mathcal{S}'_{\mathbf{x}}$ is contained in $\mathcal{P}(B)$,

$$\sum_{\mathbf{x} \in \mathcal{L}(B)} vol(\mathcal{S}'_{\mathbf{x}}) = vol\left(\bigcup_{\mathbf{x} \in \mathcal{L}(\mathbf{B})} \mathcal{S}'_{\mathbf{x}}\right) \leq vol(\mathcal{P}(B)) \tag{1.1}$$

Since $\mathcal{S}'_{\mathbf{x}}$ is a translation of $\mathcal{S}_{\mathbf{x}}$, they have the same volume, and from the assumption of the theorem we get,

$$\sum_{\mathbf{x} \in \mathcal{L}(B)} vol(S'_{\mathbf{x}}) = \sum_{\mathbf{x} \in \mathcal{L}(B)} vol(\mathcal{S}_{\mathbf{x}}) = vol(\mathcal{S}) > det(\mathcal{L}) \tag{1.2}$$

Combining (1.1) and (1.2) we get $det(\mathcal{L}(\mathbf{B})) < vol(\mathcal{P}(\mathbf{B}))$, which is a contradiction because $det(\mathcal{L}(\mathbf{B})) = vol(\mathcal{P}(\mathbf{B}))$ by definition. This proves that sets $S'_{\mathbf{x}}$ are not pairwise disjoint, i.e., for $\mathbf{x}, \mathbf{y} \in \mathcal{L}(\mathbf{B})$ there exist two sets $\mathcal{S}'_{\mathbf{x}}, \mathcal{S}'_{\mathbf{y}}$ such that $\mathcal{S}'_{\mathbf{x}} \cap \mathcal{S}'_{\mathbf{y}} \neq \emptyset$. Let $\mathbf{z}$ be any vector in $\mathcal{S}'_{\mathbf{x}} \cap \mathcal{S}'_{\mathbf{y}}$ and define

$$\mathbf{z}_1 = \mathbf{z} + \mathbf{x}$$
$$\mathbf{z}_2 = \mathbf{z} + \mathbf{y}$$

Since $\mathbf{x} \neq \mathbf{y}$ we have that $\mathbf{z}_1 \neq \mathbf{z}_2$. From $z \in \mathcal{S}'_{\mathbf{x}}$ and $z \in \mathcal{S}'_{\mathbf{y}}$ we get $\mathbf{z}_1 \in \mathcal{S}_{\mathbf{x}} \subseteq \mathcal{S}$ and $\mathbf{z}_2 \in \mathcal{S}_{\mathbf{y}} \subseteq \mathcal{S}$. Finally, the difference between $\mathbf{z}_1, \mathbf{z}_2$ is a nonzero vector that satisfies

$$\mathbf{z}_1 - \mathbf{z}_2 = \mathbf{x} - \mathbf{y} \in \mathcal{L}(\mathbf{B})$$

completing the proof of the theorem. □

As a corollary to Blichfeldt theorem we get the following theorem of Minkowski.

Theorem 1.16 (*Convex Body theorem*) *For any full-rank lattice $\mathcal{L}$ of rank n, and any centrally symmetric convex set $\mathcal{S} \subset span(\mathcal{L})$, if $vol(\mathcal{S}) > 2^n det(\mathcal{L})$, then $\mathcal{S}$ contains a nonzero lattice point. If $\mathcal{S}$ is also closed, then it suffices to take $vol(\mathcal{S}) \geq 2^n det(\mathcal{L})$.*

Proof. Let $\mathcal{S}' = \{\mathbf{x} : 2\mathbf{x} \in \mathcal{S}\}$. Then $vol(\mathcal{S}') = 2^{-n} vol(\mathcal{S}) > det(\mathcal{L})$. By Blichfeldt theorem there exist two distinct points $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{S}'$ such that $\mathbf{z}_1 - \mathbf{z}_2 \in \mathcal{L}$. From the definition of $\mathcal{S}'$, we get $2\mathbf{z}_1, 2\mathbf{z}_2 \in \mathcal{S}$ and since $\mathcal{S}$ is centrally symmetric we also have $-2\mathbf{z}_2 \in \mathcal{S}$. Finally, by convexity, the midpoint of segment $[2\mathbf{z}_1, -2\mathbf{z}_2]$ also belongs in $\mathcal{S}$, i.e.,

$$\frac{2\mathbf{z}_1 + (-2\mathbf{z}_2)}{2} = \mathbf{z}_1 - \mathbf{z}_2 \in \mathcal{S}$$

Therefore $\mathbf{z}_1 - \mathbf{z}_2 \in \mathcal{S} \cap \mathcal{L}$ and this completes the proof. □

Definition 24 ***Hermite's constant****, denoted γ_n, is the supremum of the following quantities as $\mathcal{L}$ ranges over all lattices of dimension n:*

$$\frac{\lambda_1(\mathcal{L})^2}{det(\mathcal{L})^{2/n}}$$

The quantities γ_n give an upper bound for $\lambda_1(\mathcal{L})$ but are very difficult to compute. They are known (2012) only for $1 \le n \le 8$ and $n = 24$ (see [71, ch. 2]).

We now give an upper bound for $\lambda_1(\mathcal{L})$.

Theorem 1.17 (*Minkowski's first theorem*) *Let $\mathcal{L}$ be a lattice of dimension n. Then there is a vector $\mathbf{v} \in \mathcal{L}$ satisfying*

$$\|\mathbf{v}\| \le \sqrt{n}\, det(\mathcal{L})^{1/n}$$

Proof. Let $\mathcal{L}$ be a lattice and let $\mathcal{S}$ be the hypercube in $\mathbb{R}^n$, centered at $\mathbf{0}$, whose sides have length $2\, det(\mathcal{L})^{1/n}$,

$$\mathcal{S} = \{(x_1, \ldots, x_n) \in \mathbb{R}^n : det(\mathcal{L})^{1/n} \le x_i \le det(\mathcal{L})^{1/n} \quad \text{for all } 1 \le i \le n\}$$

The set $\mathcal{S}$ is closed, centrally symmetric and convex, and its volume is

$$vol(\mathcal{S}) = \left(2\, det(\mathcal{L})^{1/n}\right)^n = 2^n\, det(\mathcal{L})$$

therefore we can apply Theorem 1.16 to deduce that there is a nonzero vector $\mathbf{v} \in S \cap \mathcal{L}$. From definition of $\mathcal{S}$, writing the coordinates of $\mathbf{v} = (v_1, \ldots, v_n)$, for all $1 \le i \le n$, we have

$$\begin{aligned}
&|v_i| \le det(\mathcal{L})^{1/n} && \Longrightarrow \\
&v_i^2 \le det(\mathcal{L})^{2/n} && \Longrightarrow \\
&\sum_{i=1}^{n} v_i^2 \le n\, det(\mathcal{L})^{2/n} && \Longrightarrow \\
&\sqrt{v_1^1 + \cdots + v_n^2} \le \sqrt{n}\, det(\mathcal{L})^{1/n} && \Longrightarrow \\
&\|\mathbf{v}\| \le \sqrt{n}\, det(\mathcal{L})^{1/n}
\end{aligned}$$

□

Since the hypercube of Theorem 1.17 has the smallest possible side length, therefore the smallest volume to satisfy the requirements of Theorem 1.16, we obtain an upper bound for $\lambda_1(\mathcal{L})$, namely $\lambda_1(\mathcal{L}) \le \sqrt{n}\, det(\mathcal{L})^{1/n}$.

Minkowski also proved a stronger result involving the geometric mean of all the successive minima.

Theorem 1.18 (*Minkowski's second theorem*) *For any lattice $\mathcal{L}$ of rank n, the successive minima (in the ℓ_2 norm) satisfy*

$$\left(\prod_{i=1}^{n} \lambda_i(\mathcal{L})\right)^{1/n} < \sqrt{n}\, det(\mathcal{L})^{1/n}$$

Now that we have given a lower bound on the shortest lattice vector we give a proof of equivalence of the two lattice definitions.

Theorem 1.19 *Let $\mathcal{L} \subset \mathbb{R}^n$, $\mathcal{L} \neq \emptyset$. Then $\mathcal{L}$ is a lattice if and only if it is a discrete additive subgroup of $\mathbb{R}^n$.*

Proof.
Assume $\mathcal{L}$ is a lattice define as the set of all integer combinations of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ which are linearly independent (Definition 13, p. 7). Then, clearly $\mathcal{L}$ is an additive subgroup of $\mathbb{R}^n$. In addition, $\forall \mathbf{x}, \mathbf{y} \in \mathcal{L}$, $\mathbf{x} - \mathbf{y} \in \mathcal{L}$. Therefore, from Theorem 1.14 we can let $\epsilon = \lambda_1(\mathcal{L})$,

$$\|\mathbf{x} - \mathbf{y}\| \geq \epsilon = \lambda_1(\mathcal{L}) > 0$$

Conversely, assume that $\mathcal{L}$ is a discrete additive subgroup of $\mathbb{R}^n$. We use induction on n. For $n = 1$, let $\{b\}$ be a basis for $\mathbb{R}$, namely

$$\mathbb{R}^1 = \{ab : a \in \mathbb{R}\}$$

Since for every $v \in \mathcal{L}$ there exists $\epsilon > 0$ such that $\mathcal{L} \cap \{r \in \mathbb{R} : \|v - r\| < \epsilon\} = \{v\}$ is finite also for all $r \in \mathbb{R}^+$, there exists a smallest positive value r_1 such that $r_1 b \in \mathcal{L}$. Therefore

$$\{ar_1 b : a \in \mathbb{Z}\} \subseteq \mathcal{L}$$

Since any $s \in \mathbb{R}$ can be written as

$$s = \left\lfloor \frac{s}{r_1} \right\rfloor r_1 + s_1 r_1,$$

for some real number s_1 with $0 \leq s_1 < 1$, then any $sb \in \mathcal{L}$ can be written in the form

$$sb = kr_1 b + s_1(r_1 b) \text{ where } k = \left\lfloor \frac{s}{r_1} \right\rfloor \in \mathbb{Z} \text{ and } 0 \leq s_1 < 1$$

Because sb, $kr_1 b \in \mathcal{L}$ then $s_1(r_1 b)$ must be in $\mathcal{L}$ and from the minimality of r_1, we must have $s_1 = 0$, so $\mathcal{L} = \{ar_1 b : a \in \mathbb{Z}\}$. This establishes the induction step.

Assume the induction hypothesis, namely that any discrete additive subgroup of $\mathbb{R}^c$ for $c < n$ is a lattice. Hence, we may assume that

$$\mathcal{L} \subset \mathbb{R}^n \text{ is discrete and } \mathcal{L} \not\subset \mathbb{R}^c \text{ for any } k < n$$

So we can choose a basis of $\mathbb{R}^n$, namely $\mathbf{b}_1, \ldots, \mathbf{b}_n$, with $\mathbf{b}_i \in \mathcal{L}$ for all $i \in \{1, \ldots, n\}$. Let

$$V = \left\{ \sum_{i=1}^{n-1} a_i \mathbf{b}_i : \forall i, a_i \in \mathbb{R} \right\}$$

. By the induction hypothesis the set $\mathcal{L}_V = \mathcal{L} \cap V$ is a lattice of dimension $n - 1$. Let $\mathbf{b}'_1, \ldots, \mathbf{b}'_{n-1}$ be a basis for $\mathcal{L}_V$. Therefore, any element $\mathbf{z} \in \mathcal{L}$ can be written as

$$\mathbf{z} = \left(\sum_{i=1}^{n-1} r_i \mathbf{b}'_i \right) + r_n \mathbf{b}_n \text{ where } r_i \in \mathbb{R}$$

By the discreteness of $\mathcal{L}$, there exist only finitely many such $\mathbf{z}$ with all r_i bounded. Thus, we may choose one $\mathbf{z}$ with $r_n > 0$, and minimal with respect to $|r_i| < 1$ for all $i \neq n$. Let $\mathbf{b}'_n$ denote this choice. Certainly the set $\{\mathbf{b}'_1, \ldots, \mathbf{b}'_{n-1}\} \cup \{\mathbf{b}'_n\}$ is linearly independent, because of the term $r_n \mathbf{b}_n$ in $\mathbf{b}'_n$. Thus,

$$\mathbb{R}^n = \left\{ \sum_{i=1}^{n} a_i \mathbf{b}'_i : \forall i, a_i \in \mathbb{R} \right\}$$

Because $\mathcal{L} \subset \mathbb{R}^n$ for any $\mathbf{v} \in \mathcal{L}$,

$$\mathbf{v} = \sum_{i=1}^{n} t_j \mathbf{b}'_i \text{ where } t_i \in \mathbb{R}$$

Let

$$\begin{aligned}
\mathbf{w} = \mathbf{v} - \sum_{i=1}^{n} \lfloor t_j \rfloor \mathbf{b}'_i &= \sum_{i=1}^{n} s_i \mathbf{b}'_i \\
&= \left(\sum_{i=1}^{n-1} s_i \mathbf{b}'_i \right) + s_n \mathbf{b}'_n \\
&= \left(\sum_{i=1}^{n-1} s_i \mathbf{b}'_i \right) + \left(\sum_{i=1}^{n-1} s_n r_i \mathbf{b}_i + s_n r_n \mathbf{b}_n \right) \text{ where } r_i \in \mathbb{R}
\end{aligned}$$

Therefore, $0 \leq s_i < 1$ for all $i \in \{1, \ldots, n\}$. By the minimality of r_n, we must have that $s_n = 0$ therefore $t_n \in \mathbb{Z}$. Also we get,

$$\mathbf{w} = \underbrace{\mathbf{v} - \sum_{i=1}^{n} \lfloor t_j \rfloor \mathbf{b}'_i}_{\text{is in } \mathcal{L}} = \underbrace{\sum_{i=1}^{n-1} s_i \mathbf{b}'_i}_{\text{is in } V}$$

so $\mathbf{w} \in \mathcal{L}$ and $\mathbf{w} \in V$, then $\mathbf{w} \in \mathcal{L}_V = \mathcal{L} \cap V$ which is a lattice of dimension $n-1$. Since any $\mathbf{v} \in \mathcal{L}$ can be written as

$$\mathbf{v} = \mathbf{w} + t_j \mathbf{b}'_n = \sum_{i=1}^{n-1} t'_i \mathbf{b}'_i + t_j \mathbf{b}'_n \text{ where } t'_i \in \mathbb{Z}$$

with $\mathbf{w} \in \mathcal{L}_V$ and $t_j \in \mathbb{Z}$ we have that $\mathcal{L}$ is a lattice of dimension n with basis vectors $\mathbf{b}'_1, \ldots, \mathbf{b}'_n$.

$\square$

1.5 Dual lattices

In this section we define the notion of the *dual lattice* and see some of its properties.

Definition 25 *For any lattice $\mathcal{L}$, the* **dual lattice** *of $\mathcal{L}$ is defined as*

$$\mathcal{L}^* = \{\mathbf{y} \in span(\mathcal{L}) : \forall \mathbf{x} \in \mathcal{L},\ \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\}$$

The dual lattice $\mathcal{L}^*$ has the same span with $\mathcal{L}$. We now prove that the dual lattice is indeed a lattice itself.

Proof. Let $\mathcal{L} \subset \mathbb{R}^m$ be a lattice of rank n, and let $\mathbf{B} \in \mathbb{R}^{m \times n}$ be its basis. Define the dual lattice:

$$\mathcal{L}^* = \{\mathbf{y} \in span(\mathcal{L}) : \forall \mathbf{x} \in \mathcal{L},\ \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\}$$

Because $\mathbf{y} \in span(\mathcal{L})$ and $\mathbf{x} \in \mathcal{L}$ we can write them in matrix form as $\mathbf{y} = \mathbf{Bu}$ where $\mathbf{u} \in \mathbb{R}^n$ and $\mathbf{x} = \mathbf{Bw}$ where $\mathbf{w} \in \mathbb{Z}^n$. Now solve $\mathbf{y} = \mathbf{Bu}$ for $\mathbf{u}$ and multiply it again with $\mathbf{B}$:

$$\begin{aligned}
\mathbf{y} &= \mathbf{Bu} && \Rightarrow \\
\mathbf{B}^\intercal \mathbf{y} &= \mathbf{B}^\intercal \mathbf{Bu} && \Rightarrow \\
\mathbf{u} &= \left(\mathbf{B}^\intercal \mathbf{B}\right)^{-1} \mathbf{B}^\intercal \mathbf{y} && \Rightarrow \\
\mathbf{Bu} &= \mathbf{B} \left(\mathbf{B}^\intercal \mathbf{B}\right)^{-1} \mathbf{B}^\intercal \mathbf{y} = \mathbf{y}
\end{aligned}$$

The matrix $\mathbf{B}^\intercal \mathbf{B}$ is invertible because $\mathbf{B}$ is a basis for $\mathcal{L}$, hence its columns are linear independent vectors. Because $\langle \mathbf{x}, \mathbf{y} \rangle = det(\mathbf{x}^\intercal \mathbf{y})$ we have that

$$\begin{aligned}
\langle \mathbf{x}, \mathbf{y} \rangle &= det((\mathbf{Bw})^\intercal \mathbf{B} \left(\mathbf{B}^\intercal \mathbf{B}\right)^{-1} \mathbf{B}^\intercal \mathbf{y}) \\
&= det(\mathbf{w}^\intercal \mathbf{B}^\intercal \mathbf{B} \left(\mathbf{B}^\intercal \mathbf{B}\right)^{-1} \mathbf{B}^\intercal \mathbf{y}) \\
&= det(\mathbf{w}^\intercal \underbrace{\mathbf{B}^\intercal \mathbf{y}}_{\mathbf{z}}) \\
&= det(\mathbf{w}^\intercal \mathbf{z}) \in \mathbb{Z} \\
&= det(w_1 z_1 + \cdots + w_n z_n) \in \mathbb{Z} \quad \text{for all } \mathbf{w} \in \mathbb{Z}^n
\end{aligned}$$

Since we want this to hold for all $\mathbf{w} \in \mathbb{Z}^n$ we can choose $\mathbf{w} = \mathbf{e}_i$ for $1 \leq i \leq n$ where $\mathbf{e}_i$ is the standard basis vector from Definition 7 (p. 4) and we get that $w_i z_i \in \mathbb{Z}$. We already know that $w_i \in \mathbb{Z}$, so we get that $z_i \in \mathbb{Z}$. Because $\mathbf{B}$ has n linearly independent vectors as rows then $det(\mathbf{B}^\intercal \mathbf{B}) \neq 0$ and so the $n \times n$ matrix

$\left(\mathbf{B}^{\intercal}\mathbf{B}\right)^{-1}$ exists, and has also a nonzero determinant. Thus, $rank\left(\mathbf{B}\left(\mathbf{B}^{\intercal}\mathbf{B}\right)^{-1}\right) = rank(\mathbf{B}) = n$ so the $m \times n$ matrix $\mathbf{B}\left(\mathbf{B}^{\intercal}\mathbf{B}\right)^{-1}$ has n linearly independent vectors as columns.

From the equation,

$$\mathbf{y} = \mathbf{B}\left(\mathbf{B}^{\intercal}\mathbf{B}\right)^{-1}\underbrace{\mathbf{B}^{\intercal}\mathbf{y}}_{\mathbf{z}}$$

the properties that $\mathbf{B}^{\intercal}\mathbf{y} = \mathbf{z} \in \mathbb{Z}^n$ and that the matrix $\left(\mathbf{B}\left(\mathbf{B}^{\intercal}\mathbf{B}\right)^{-1}\right)$ consists of column vectors that are linearly independent we conclude that $\mathcal{L}^*$ is a lattice with basis $\mathbf{B}^* = \mathbf{B}\left(\mathbf{B}^{\intercal}\mathbf{B}\right)^{-1}$. $\square$

In the case of a full-rank lattice we have that $\mathbf{B}^* = \left(\mathbf{B}^{\intercal}\right)^{-1}$. The next theorem provides some useful properties for a lattice $\mathcal{L}$ and its dual lattice $\mathcal{L}^*$.

Theorem 1.20 *Let $\mathcal{L} \subset \mathbb{R}^m$ be a lattice of rank n with basis $\mathbf{B} \in \mathbb{R}^{m \times n}$, and let $\mathcal{L}^*$ be its dual lattice with basis $\mathbf{B}^* = \mathbf{B}\left(\mathbf{B}^{\intercal}\mathbf{B}\right)^{-1}$.*

The following properties hold:

(a) $(\mathcal{L}^*)^* = \mathcal{L}$

(b) $det(\mathcal{L}^*) = \dfrac{1}{det(\mathcal{L})}$

(c) $\lambda_1(\mathcal{L}) \cdot \lambda_1(\mathcal{L}^*) \leq n$

(d) $\lambda_1(\mathcal{L}) \cdot \lambda_n(\mathcal{L}^*) \geq 1$

Proof.

(a) The basis for $(\mathcal{L}^*)^*$ is

$$\begin{aligned}
(\mathbf{B}^*)^* &= \mathbf{B}^*\left((\mathbf{B}^*)^{\intercal}\mathbf{B}^*\right)^{-1} \\
&= \left(\mathbf{B}\left(\mathbf{B}^{\intercal}\mathbf{B}\right)^{-1}\right)\left(\left(\mathbf{B}\left(\mathbf{B}^{\intercal}\mathbf{B}\right)^{-1}\right)^{\intercal}\left(\mathbf{B}\left(\mathbf{B}^{\intercal}\mathbf{B}\right)^{-1}\right)\right)^{-1} \\
&= \mathbf{B} \text{ which is a basis for } \mathcal{L}.
\end{aligned}$$

Thus, $(\mathcal{L}^*)^* = \mathcal{L}$.

(b) We have that,

$$
\begin{aligned}
det(\mathcal{L}^*) &= \sqrt{det\left((\mathbf{B}^*)^\intercal \mathbf{B}^*\right)} \\
&= \sqrt{det\left(\left(\mathbf{B}\left(\mathbf{B}^\intercal\mathbf{B}\right)^{-1}\right)^\intercal \left(\mathbf{B}\left(\mathbf{B}^\intercal\mathbf{B}\right)^{-1}\right)\right)} \\
&= \sqrt{det\left(\left(\left(\mathbf{B}^\intercal\mathbf{B}\right)^{-1}\right)^\intercal \mathbf{B}^T\mathbf{B}\left(\mathbf{B}^\intercal\mathbf{B}\right)^{-1}\right)} \\
&= \sqrt{det\left(\left(\left(\mathbf{B}^\intercal\mathbf{B}\right)^{-1}\right)^\intercal\right)} \\
&= \sqrt{det\left(\left(\mathbf{B}^\intercal\mathbf{B}\right)^{-1}\right)} \\
&= \sqrt{\frac{1}{det\left(\mathbf{B}^\intercal\mathbf{B}\right)}} = \frac{1}{det(\mathcal{L})}
\end{aligned}
$$

(c) From Theorem 1.17 (p. 23) and part (b), we have that

$$\lambda_1(\mathcal{L}) \leq \sqrt{n}\, det(\mathcal{L})^{1/n} \quad \text{and} \quad \lambda_1(\mathcal{L}^*) \leq \sqrt{n}\, det(\mathcal{L}^*)^{1/n} = \frac{\sqrt{n}}{det(\mathcal{L})^{1/n}}$$

Thus,

$$\lambda_1(\mathcal{L}) \cdot \lambda_1(\mathcal{L}^*) \leq \frac{\sqrt{n}\, det(\mathcal{L})^{1/n}}{det(\mathcal{L})^{1/n}} = \sqrt{n}$$

(d) Let $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$ (can be more that one) and let $\mathbf{v}_1^*, \ldots, \mathbf{v}_n^*$ be any set on n linearly independent vectors in $\mathcal{L}^*$. Then there exists an $i \in \{1, \ldots, n\}$ such that $\langle \mathbf{v}_i^*, \mathbf{v} \rangle \neq 0$ exactly because the vectors $\mathbf{v}_j^*$ are linearly independent. We have that $\langle \mathbf{v}_i^*, \mathbf{v} \rangle = k$ where $k \in \mathbb{Z}^*$ so,

$$\lambda_n(\mathcal{L}^*) \geq \mathbf{v}_i^* \geq \frac{k}{\|\mathbf{v}\|} \geq \frac{1}{\|\mathbf{v}\|} = \frac{1}{\lambda_1(\mathcal{L})} \Rightarrow \lambda_1(\mathcal{L}) \cdot \lambda_n(\mathcal{L}^*) \geq 1$$

$\square$

2

Lattice basis reduction

2.1 Asymptotic notation

Throughout this thesis we will use standard asymptotic notation symbols O, o, Ω, ω and Θ to measure the running-time complexity of algorithms. We recall their definitions here:

- $f(n) = O(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$
- $f(n) = o(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
- $f(n) = \Omega(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} > 0$
- $f(n) = \omega(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$
- $f(n) = \Theta(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$ where c is some constant.
- $f(n) = \tilde{O}(g(n))$ if $f(n) = O(g(n) \cdot log^c(n))$ for some fixed constant c.

A function $f(n)$ is ***negligible***, written $negl(n)$, if $f(n) = o(n^{-c})$ for every constant c.

2.2 Computational lattice problems

Minkowski's first theorem (Theorem 1.17, p. 23) gives a simple way to bound the length of the shortest vector of a lattice. But this bound is not always tight. For example consider the lattice $\mathcal{L}$ generated by vectors $\mathbf{b}_1 = (\epsilon, 0)^\intercal$ and $\mathbf{b}_2 = (0, 1/\epsilon)^\intercal$ for some $\epsilon > 0$. The determinant of $\mathcal{L}$ is 1 which gives an upper bound $\lambda_1(\mathcal{L}) \leq \sqrt{2}$ but the shortest vector is $\lambda_1(\mathcal{L}) = \|\mathbf{b}_1\| = \epsilon$ which can be arbitrarily small. Furthermore, the proof of Minkowski's first theorem does not provide us with a constructive way to find $\lambda_1(\mathcal{L})$.

The problem of finding a nonzero lattice vector of length λ_1 is the *Shortest Vector Problem* and it was formulated by Dirichlet in 1842.

Definition 26 (Shortest Vector Problem, SVP) *Given a basis* $\mathbf{B} \in \mathbb{Z}^{m\times n}$ *for a lattice* $\mathcal{L}(\mathbf{B})$*, find a nonzero lattice vector* $\mathbf{v}$ *such that* $\|\mathbf{v}\| \leq \|\mathbf{w}\|$ *for any other nonzero vector* $\mathbf{w} \in \mathcal{L}(\mathbf{B})$.

In addition to the search version of the SVP we also define its decision version:

Definition 27 *Given a basis* $\mathbf{B} \in \mathbb{Z}^{m\times n}$ *for a lattice* $\mathcal{L}(\mathbf{B})$ *and a positive number* $r \in \mathbb{Q}$*, determine whether* $\lambda_1(\mathcal{L}(\mathbf{B})) \leq r$ *or not.*

Another basic computational problem is the *Closest Vector Problem*.

Definition 28 (Closest Vector Problem, CVP) *Given a basis* $\mathbf{B} \in \mathbb{Z}^{m\times n}$ *for a lattice* $\mathcal{L}(\mathbf{B})$*, and a target vector* $\mathbf{t} \in \mathbb{Z}^m$*, find a lattice vector* $\mathbf{v}$ *such that* $dist(\mathbf{v}, \mathbf{t}) \leq dist(\mathbf{t}, \mathcal{L}(\mathbf{B}))$*, i.e., the vector* $\mathbf{v}$ *is closest to vector* $\mathbf{t}$.

Again, in addition to the search version of the CVP we also define its decision version:

Definition 29 *Given a basis* $\mathbf{B} \in \mathbb{Z}^{m\times n}$ *for a lattice* $\mathcal{L}(\mathbf{B})$*, a target vector* $\mathbf{t} \in \mathbb{Z}^m$*, and a positive number* $r \in \mathbb{Q}$ *decide whether there is a (nonzero) lattice vector* $\mathbf{v}$ *such that* $dist(\mathbf{v}, \mathbf{t}) \leq r$.

From the previous definitions it is implied that $\mathbf{v} = \mathbf{Bx}$ and $\mathbf{w} = \mathbf{By}$ with $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^n$. Notice that we restrict the lattice basis $\mathbf{B}$ and the vector $\mathbf{t}$ to consist of integers because we want the input to be representable in finite number of bits so that we can consider those two problems as standard computational problems. It can be shown that the decision and search versions are polynomially equivalent.

To date, for both SVP and CVP, no polynomial time algorithm is known. In fact, we do not even know how to find nonzero lattice vectors of length within the Minkowski's bound (Theorem 1.17, p. 23).

The hardness of solving SVP and CVP has led to consideration of approximation versions for these problems. We now define the promise [1] approximation versions of SVP and CVP. A solution to any of the promise problems below implies a solution to the corresponding optimization problem (that is, the problem that asks for an approximation to the corresponding lattice parameter, e.g., λ_1). The following definitions are parameterized by a (monotone) function (the gap function) $\gamma : \mathbb{Z}^+ \rightarrow \mathbb{R}^+$ of the lattice dimension where $\gamma(n) \geq 1$. For computational purposes the range of the gap function may be $\mathbb{Z}^+$ or $\mathbb{Q}^+$.

Definition 30 ($\boldsymbol{GapSVP_\gamma}$) *An input to $GapSVP_\gamma$ is a pair $(\mathbf{B}, d)$ where $\mathbf{B}$ is an n-dimensional basis for a lattice $\mathcal{L}$ and d is a positive number. In YES inputs $\lambda_1(\mathcal{L}(\mathbf{B})) \leq d$ and in NO inputs $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$.*

Definition 31 ($\boldsymbol{GapCVP_\gamma}$) *An input to $GapCVP_\gamma$ is a triple $(\mathbf{B}, \mathbf{t}, d)$ where $\mathbf{B}$ is an n-dimensional basis for a lattice $\mathcal{L}$, $\mathbf{t} \in span(\mathcal{L})$ is a target vector, and d is a positive number. In YES inputs $dist(\mathbf{t}, \mathcal{L}(\mathbf{B})) \leq d$ and in NO inputs $dist(\mathbf{t}, \mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$.*

Notice that for $\gamma(n) = 1$ the promise problems SVP_γ and CVP_γ are equivalent to the decision problems of SVP and CVP respectively. In an analogous way we define the search variants of SVP_γ and CVP_γ.

Definition 32 ($\boldsymbol{SVP_\gamma}$) *An input to SVP_γ is a basis $\mathbf{B}$ for an n-dimensional lattice $\mathcal{L}$ and the task if to find a nonzero vector $\mathbf{v} \in \mathcal{L}$ such that*

$$\|\mathbf{v}\| \leq \gamma(n) \cdot \lambda_1(\mathcal{L}(\mathbf{B}))$$

Definition 33 ($\boldsymbol{CVP_\gamma}$) *An input to CVP_γ is a pair $(\mathbf{B}, \mathbf{t})$ where $\mathbf{B}$ is an n-dimensional basis for a lattice $\mathcal{L}$, $\mathbf{t} \in span(\mathcal{L})$ is a target vector, and the task is to find a vector $\mathbf{v} \in \mathcal{L}$ such that*

$$dist(\mathbf{v}, \mathbf{t}) \leq \gamma(n) \cdot dist(\mathbf{t}, \mathcal{L}(\mathbf{B}))$$

The shortest vector and the closest vector problems are fundamental lattice problems but there are other lattice problems which are thought to be computationally hard such as the following:

[1] Promise problems are a generalization of decision problems where one is asked whether a given input satisfies one of two mutually exclusive properties. Unlike decision problems, these two properties are not necessarily exhaustive. The problem is, under the promise that the given input satisfies one of the two conditions, tell which of the two properties is satisfied. If the input satisfies neither property, then any answer is acceptable.

Definition 34 (Closest Vector Problem with Preprocessing, CVPP) *Given a basis* $\mathbf{B} \in \mathbb{Z}^{m \times n}$ *for a lattice* $\mathcal{L}(\mathbf{B})$*, and a target vector* $\mathbf{t} \in \mathbb{Z}^m$*, one is allowed to do arbitrary preprocessing on it and store polynomial (in the dimension of the lattice) amount of information. The task is to find a lattice vector* $\mathbf{v}$ *such that* $dist(\mathbf{v}, \mathbf{t}) \leq dist(\mathbf{t}, \mathcal{L}(\mathbf{B}))$*, i.e., the vector* $\mathbf{v}$ *is closest to vector* $\mathbf{t}$*.*

Definition 35 (γ-Shortest Independent Vectors Problem, $SIVP_\gamma$) *Given a lattice basis* $\mathbf{B} \in \mathbb{Z}^{m \times n}$ *for the lattice* $\mathcal{L}$ *the task is to find* n *linearly independent lattice vectors* $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L}(\mathbf{B})$ *so that* $\max\limits_{i=1,\ldots,n} \|\mathbf{v}_i\| \leq \gamma(n) \cdot \lambda_n(\mathcal{L}(\mathbf{B}))$*.*

Definition 36 (γ-unique Shortest Vector Problem, $uSVP_\gamma$) *Given a lattice basis* $\mathbf{B} \in \mathbb{Z}^{m \times n}$ *for the lattice* $\mathcal{L}$ *for which* $\lambda_2(\mathcal{L}(\mathbf{B})) > \gamma(n) \cdot \lambda_1(\mathcal{L}(\mathbf{B}))$ *the task is to find a nonzero vector* $\mathbf{v} \in \mathcal{L}$ *such that*

$$\|\mathbf{v}\| \leq \gamma(n) \cdot \lambda_1(\mathcal{L}(\mathbf{B}))$$

Definition 37 (Shortest Basis Problem, SBP) *Given a lattice basis* $\mathbf{B} \in \mathbb{Z}^{m \times n}$ *the task is to find the minimum length* r *such that each basis vector has length at most* r*.*

In an analogous way we define the promise approximation versions of $CVPP_\gamma$ and $SIVP_\gamma$ and $uSVP_\gamma$ for $\gamma(n) \geq 1$.

Definition 38 ($GapCVPP_\gamma$) *An input to* $GapCVP_\gamma$ *is a triple* $(\mathbf{B}, \mathbf{t}, d)$ *where* $\mathbf{B}$ *is an* n*-dimensional basis for a lattice* $\mathcal{L}$*,* $\mathbf{t} \in span(\mathcal{L})$ *is a target vector, and* d *is a rational number and one is allowed to do arbitrary preprocessing on it and store polynomial (in the dimension of the lattice) amount of information. In* YES *inputs* $dist(\mathbf{t}, \mathcal{L}(\mathbf{B})) \leq d$ *and in* NO *inputs* $dist(\mathbf{t}, \mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$*.*

Definition 39 ($GapSIVP_\gamma$) *An input to* $GapSIVP_\gamma$ *is a pair* $(\mathbf{B}, d)$ *where* $\mathbf{B}$ *is an* n*-dimensional basis for a lattice* $\mathcal{L}$ *and* d *is a rational number. In* YES *inputs* $\lambda_n(\mathcal{L}(\mathbf{B})) \leq d$ *and in* NO *inputs* $\lambda_n(\mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$*.*

Definition 40 ($GapuSVP_\gamma$) *An input to* $GapuSVP_\gamma$ *is a pair* $(\mathbf{B}, d)$ *where* $\mathbf{B}$ *is an* n*-dimensional basis for a lattice* $\mathcal{L}$ *and* d *is a positive number. In* YES *inputs* $\lambda_1(\mathcal{L}(\mathbf{B})) \leq d$ *and* $\lambda_2(\mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$ *and in* NO *inputs* $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$ *(and* $\lambda_2(\mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$*).*

Definition 41 (Covering Radius Problem, $GapCRP_\gamma$) *Let* $\rho(\mathcal{L}(\mathbf{B}))$ *denote the* ***covering radius*** *of the lattice* $\mathcal{L}(\mathbf{B})$*, i.e., the smallest* r *such that (closed) balls of radius* r *centered at lattice points cover* $span(\mathbf{B})$*. Equivalently,*

$$\rho(\mathcal{L}(\mathbf{B})) = \max_{\mathbf{v} \in span(\mathbf{B})} dist(\mathbf{v}, \mathcal{L}(\mathbf{B}))$$

An input to $GapCRP_\gamma$ is a pair $(\mathbf{B}, d)$ where $\mathbf{B}$ is an n-dimensional basis for a lattice $\mathcal{L}$ and d is a rational number. In YES inputs $\rho(\mathcal{L}(\mathbf{B})) \leq d$ and in NO inputs $\rho(\mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$.

For the Covering Radius Problem, there is no known search problem whose solution can be verified in polynomial time and thus it is not solvable even in non-deterministic polynomial time. In fact the Covering Radius Problem is in Π_2 for the ℓ_p norm $(p \geq 1, p = \infty)$, a complexity class presumably strictly bigger than NP.

Another fundamental problem is the one of the reduced basis. *Given a basis for a lattice which in general consists of long vectors, we want to find another "reduced" basis for the same lattice, that is, a basis consisting of short vectors and close to orthogonal.* We will describe algorithms for this problem in the next sections.

There are also many computational problems that can be solved in polynomial time. Bellow we mention some of them (see Micciancio [61, p. 18-19]):

(a) ***Membership:*** Given a basis $\mathbf{B}$ and a vector $\mathbf{v}$, decide whether $\mathbf{v}$ belongs to the lattice $\mathcal{L}(\mathbf{B})$.

(b) ***Basis:*** Given a set of possibly linearly dependent integral vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$, find a basis of the lattice they generate.

(c) ***Union:*** Given two lattices with integral basis $\mathbf{B}_1$ and $\mathbf{B}_2$, compute a basis for the smallest lattice containing $\mathcal{L}(\mathbf{B}_1) \cup \mathcal{L}(\mathbf{B}_2)$.

(d) ***Intersection:*** Given two lattices with integral basis $\mathbf{B}_1$ and $\mathbf{B}_2$, compute a basis for the intersection $\mathcal{L}(\mathbf{B}_1) \cap \mathcal{L}(\mathbf{B}_2)$.

(e) ***Equivalence:*** Given two lattices with integral basis $\mathbf{B}_1$ and $\mathbf{B}_2$, decide if they generate the same lattice $\mathcal{L}(\mathbf{B}_1) = \mathcal{L}(\mathbf{B}_2)$.

(f) ***Dual:*** Given a lattice with basis $\mathbf{B}$ compute a basis $\mathbf{B}^*$ for the dual lattice. From Theorem 1.20 (p. 28) we know that $\mathbf{B}^* = \mathbf{B}\left(\mathbf{B}^\intercal \mathbf{B}\right)^{-1}$

2.3 Gaussian lattice basis reduction

In general, lattice problems become harder as the dimension grows bigger. But for a 2-dimensional lattice the Gauss lattice basis reduction algorithm solves SVP in polynomial time.

Definition 42 *For $r \in \mathbb{R}$ we write $\lfloor r \rceil$ for the* ***nearest integer*** *to r.*

Definition 43 *We say that a basis $\mathbf{b}_1, \mathbf{b}_2$ of a lattice $\mathcal{L} \subset \mathbb{R}^2$ is* ***minimal*** *if $\mathbf{b}_1$ is a shortest nonzero vector in $\mathcal{L}$ and $\mathbf{b}_2$ is a shortest nonzero vector in $\mathcal{L}$ which is not a multiple of $\mathbf{b}_1$, i.e., $\mathbf{b}_1 = \lambda_1(\mathcal{L})$ and $\mathbf{b}_2 = \lambda_2(\mathcal{L})$.*

The underlying idea of the algorithm[2] is to alternately subtract multiples of one basis vector from the other until no further improvement is possible.

Theorem 2.1 *Let $\mathcal{L} \subset \mathbb{R}^2$ be a 2-dimensional lattice with basis vectors $\mathbf{b}_1$ and $\mathbf{b}_2$. The following algorithm (**Gauss algorithm**) terminates and yields a minimal basis for $\mathcal{L}$.*

Algorithm 1: Gaussian lattice basis reduction.

Input : Basis $\mathbf{b}_1, \mathbf{b}_2$ for the lattice $\mathcal{L} \subset \mathbb{R}^2$.
Output: A minimal basis $\mathbf{b}_1, \mathbf{b}_2$ for the lattice $\mathcal{L}$.

$reduced \leftarrow false$;
while $reduced \neq true$ **do**
    if $\|\mathbf{b}_1\| > \|\mathbf{b}_2\|$ **then**
        swap $\mathbf{b}_1$ and $\mathbf{b}_2$;
    end
    $\mu \leftarrow \left\lfloor \dfrac{\langle \mathbf{b}_1, \mathbf{b}_2 \rangle}{\|\mathbf{b}_1\|^2} \right\rceil$;
    if $\mu = 0$ **then**
        $reduced \leftarrow true$;
    else
        $\mathbf{b}_2 \leftarrow \mathbf{b}_2 - \mu \mathbf{b}_1$;
    end
end
return $\mathbf{b}_1$ and $\mathbf{b}_2$;

[2] The algorithm was first written down by Lagrange and later by Gauss, but is usually called the "Gauss algorithm".

Proof. The proof follows Beukers [14, ch. 3]. Regarding $\mathbf{b}_1, \mathbf{b}_2$ as row vectors we have in matrix form:

$$\left.\begin{array}{ll}\mathbf{b}_1 & \leftarrow \mathbf{b}_1 \\ \mathbf{b}_2 & \leftarrow \mathbf{b}_2 - \mu\mathbf{b}_1\end{array}\right\} \Rightarrow \begin{pmatrix}\mathbf{b}_1 \\ \mathbf{b}_2\end{pmatrix} \leftarrow \underbrace{\begin{pmatrix}1 & 0 \\ -\mu & 1\end{pmatrix}}_{\mathbf{G}} \begin{pmatrix}\mathbf{b}_1 \\ \mathbf{b}_2\end{pmatrix}$$

Since $det(\mathbf{G}) = 1$, i.e., matrix $\mathbf{G}$ is unimodular, it is clear that $\mathbf{b}_1, \mathbf{b}_2$ remain basis vectors after each iteration of the algorithm. The algorithm swaps $\mathbf{b}_1$ and $\mathbf{b}_2$ in step 4 if $\|\mathbf{b}_1\| > \|\mathbf{b}_2\|$ and so the length of $\mathbf{b}_1$ strictly decreases. For any real number $r > 0$, there are only finitely many lattice points in the disk $\overline{\mathbb{B}}(\mathbf{z}, r)$. It follows that the algorithm terminates after a finite number of iterations.

Now suppose that the algorithm has terminated and returned vectors $\mathbf{b}_1$ and $\mathbf{b}_2$. This means that $\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\|$ and from step 6 we also get that

$$\frac{\langle \mathbf{b}_1, \mathbf{b}_2\rangle}{\|\mathbf{b}_1\|^2} \leq \frac{1}{2} \Rightarrow \langle \mathbf{b}_1, \mathbf{b}_2\rangle \leq 2\|\mathbf{b}_1\|^2 \tag{2.1}$$

Let $\mathbf{v}$ be any nonzero vector in $\mathcal{L}$, so that $\mathbf{v} = a_1\mathbf{b}_1 + a_2\mathbf{b}_2$ for some $a_1, a_2 \in \mathbb{Z}$, not both zero. We have that,

$$\begin{aligned}
\|\mathbf{v}\|^2 &= \|a_1\mathbf{b}_1 + a_2\mathbf{b}_2\|^2 \\
&= a_1^2\|\mathbf{b}_1\|^2 + 2a_1a_2\langle \mathbf{b}_1, \mathbf{b}_2\rangle + a_2^2\|\mathbf{b}_2\|^2 \\
&\geq a_1^2\|\mathbf{b}_1\|^2 - 2|a_1a_2|\langle \mathbf{b}_1, \mathbf{b}_2\rangle + a_2^2\|\mathbf{b}_2\|^2 \\
&\geq a_1^2\|\mathbf{b}_1\|^2 - |a_1a_2|\|\mathbf{b}_1\|^2 + a_2^2\|\mathbf{b}_2\|^2 && \text{from (2.1)} \\
&\geq a_1^2\|\mathbf{b}_1\|^2 - |a_1a_2|\langle \mathbf{b}_1, \mathbf{b}_2\rangle + a_2^2\|\mathbf{b}_1\|^2 && \text{since } \|\mathbf{b}_1\| \leq \|\mathbf{b}_2\| \\
&= \left(a_1^2 - |a_1a_2| + a_2^2\right)\|\mathbf{b}_1\|^2 \\
&= \left(|a_1|^2 - |a_1a_2| + |a_2|^2\right)\|\mathbf{b}_1\|^2 \\
&= \left[(|a_1| - |a_2|)^2 + |a_1a_2|\right]\|\mathbf{b}_1\|^2 \\
&\geq \|\mathbf{b}_1\|^2 && \text{since } a_1, a_2 \text{ are not both zero}
\end{aligned}$$

Therefore $\|\mathbf{v}\| \geq \|\mathbf{b}_1\|$, and so $\mathbf{b}_1$ is a shortest vector in $\mathcal{L}$.

Now suppose that $\mathbf{v} = a_1\mathbf{b}_1 + a_2\mathbf{b}_2$ is linearly independent of $\mathbf{b}_1$, that is $a_2 \neq 0$. As before we have,

$$
\begin{aligned}
\|\mathbf{v}\|^2 &\geq a_1^2\|\mathbf{b}_1\|^2 - |a_1a_2|\|\mathbf{b}_1\|^2 + a_2^2\|\mathbf{b}_2\|^2 && \text{from (2.1)}\\
&= a_1^2\|\mathbf{b}_1\|^2 - |a_1a_2|\|\mathbf{b}_1\|^2 + \frac{1}{4}a_2^2\|\mathbf{b}_2\|^2 + \frac{3}{4}a_2^2\|\mathbf{b}_2\|^2\\
&\geq a_1^2\|\mathbf{b}_1\|^2 - |a_1a_2|\|\mathbf{b}_1\|^2 + \frac{1}{4}a_2^2\|\mathbf{b}_1\|^2 + \frac{3}{4}a_2^2\|\mathbf{b}_2\|^2 && \text{since } \|\mathbf{b}_1\| \leq \|\mathbf{b}_2\|\\
&= \left(|a_1| - \frac{1}{2}|a_2|\right)^2 \|\mathbf{b}_1\| + \frac{3}{4}a_2^2\|\mathbf{b}_2\|^2
\end{aligned}
$$

Hence $\|\mathbf{v}\| \geq \|\mathbf{b}_2\|$ if $\frac{3}{4}a_2^2 \geq 1$, that is if $|b| \geq 2$. In case that $|a_2| = 1$ we have that,

$$
\begin{aligned}
\|\mathbf{v}\|^2 &\geq a_1^2\|\mathbf{b}_1\|^2 - |a_1|\|\mathbf{b}_1\|^2 + \|\mathbf{b}_2\|^2 && \text{from (2.1)}\\
&= |a_1|\left(|a_1| - 1\right)\|\mathbf{b}_1\|^2 + \|\mathbf{b}_2\|^2
\end{aligned}
$$

Since $a_1 \in \mathbb{Z}$ we get that $|a_1|\left(|a_1| - 1\right) = 0$ if $|a_1| \leq 1$ and $|a_1|\left(|a_1| - 1\right) > 0$ for $|a_1| \geq 2$, so $|a_1|\left(|a_1| - 1\right) \geq 0$ for all $a_1 \in \mathbb{Z}$. It follows that $\|\mathbf{v}\|^2 \geq \|\mathbf{b}_2\|^2$ in that case too, therefore $\mathbf{b}_2$ is a shortest vector in $\mathcal{L}$ linearly independent for $\mathbf{b}_1$ since $\mathbf{b}_1$ and $\mathbf{b}_2$ are basis vectors for $\mathcal{L}$. $\square$

From the above proof we conclude that $\lambda_1(\mathcal{L}) = \mathbf{b}_1$ and $\lambda_2(\mathcal{L}) = \mathbf{b}_2$.

At line 3 of the Algorithm 1 if we change the **if** condition to $\|\mathbf{b}_1\| \geq t\|\mathbf{b}_2\|$ where $t \geq 1$ is an input parameter we get a new algorithm, called the ***t-Gauss algorithm***.

Algorithm 2: t-Gaussian lattice basis reduction.

Input : A parameter $t \geq 1$ and a basis $\mathbf{b}_1, \mathbf{b}_2$ for the lattice $\mathcal{L} \subset \mathbb{R}^2$.
Output: A minimal basis $\mathbf{b}_1, \mathbf{b}_2$ for the lattice $\mathcal{L}$.

$reduced \leftarrow false$;
while $reduced \neq true$ **do**
  if $\|\mathbf{b}_1\| > t\|\mathbf{b}_2\|$ **then**
    swap $\mathbf{b}_1$ and $\mathbf{b}_2$;
  end
  $\mu \leftarrow \left\lfloor \dfrac{\langle \mathbf{b}_1, \mathbf{b}_2 \rangle}{\|\mathbf{b}_1\|^2} \right\rceil$;
  if $\mu = 0$ **then**
    $reduced \leftarrow true$;
  else
    $\mathbf{b}_2 \leftarrow \mathbf{b}_2 - \mu \mathbf{b}_1$;
  end
end
return $\mathbf{b}_1$ and $\mathbf{b}_2$;

For $t = 1$ the t-Gauss algorithm is the same as the Gauss algorithm. For $t > 1$ the t-Gauss algorithm asks for a new vector that is not shorter than the previous vectors, but is at most t times greater or equal to the previous $\mathbf{b}_2$ vector. This algorithm is used in the LLL algorithm which we will consider in the next section.

Vallée in [83] showed that the run-time complexity of Algorithm 1 is

$$O\left(\frac{1}{2}\log_{\sqrt{3}}(\mathcal{I}) + 2\right)$$

and for $t > 1$ the run-time complexity of Algorithm 2 is

$$O\left(\frac{1}{2}\log_{t}(\mathcal{I}) + 2\right)$$

where $\mathcal{I} = \|\mathbf{b}_1\|^2 + \|\mathbf{b}_2\|^2$, thus polynomial in the input size for both algorithms, therefore in a 2-dimensional lattice we can solve SVP in polynomial time using the Gauss algorithm.

2.4 The Lenstra-Lenstra-Lovász algorithm

Gauss's lattice basis reduction algorithm gives an efficient way to find a shortest nonzero lattice vector in a 2-dimensional lattice. But what can we do when as the dimension increases and SVP becomes harder ? A major advance came in 1982 with the publication of the LLL algorithm [51]. The algorithm is called LLL or L^3 after the initials of its authors, namely, A. K. Lenstra, H. W. Lenstra Jr. and L. Lovász. In their publication, Lenstra, Lenstra and Lovász used the LLL algorithm to factor polynomials with rational coefficients.

The LLL algorithm runs in polynomial time and can find an approximation to a shortest lattice vector and has application in areas such as cryptography, computational number theory and integer programming among others.

First, we must define what is a reduced basis.

Definition 44 *The **reduction parameter** is a real number δ such that*

$$\frac{1}{4} < \delta < 1.$$

The standard value for this parameter is $\delta = \frac{3}{4}$.

Definition 45 *Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a basis for a lattice $\mathcal{L} \subset \mathbb{R}^n$ and let $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ be its Gram-Schmidt orthogonalization (in this section we will consider the basis vectors as row vectors). The basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is called δ-**reduced** if it satisfies*

(a) $|\mu_{ij}| = \frac{|\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle|}{\|\tilde{\mathbf{b}}_j\|^2} \leq \frac{1}{2}$ for all $1 \leq j < i \leq n$

(b) $\|\tilde{\mathbf{b}}_i + \mu_{i,i-1}\tilde{\mathbf{b}}_{i-1}\|^2 \geq \delta \|\tilde{\mathbf{b}}_{i-1}\|^2$ for all $2 \leq i \leq n$.

Condition (a) is called the ***size condition***. Condition (b) can be written as

$$\|\tilde{\mathbf{b}}_i\|^2 \geq \left(\delta - \mu_{i,i-1}^2\right) \|\tilde{\mathbf{b}}_{i-1}\|^2 \text{ for } 2 \leq i \leq n$$

and is called ***exchange*** or **Lovász *condition***.

Condition (a) says that each basis vector $\mathbf{b}_i$ is "almost orthogonal" to the span of the previous vectors, since by Theorem 1.11(b), (p. 16), we have that

$$span(\mathbf{b}_1, \ldots, \mathbf{b}_k) = span(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_k) \text{ for } 1 \leq k \leq n$$

so we want the $\mu_{ij} = \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\|\tilde{\mathbf{b}}_j\|^2}$ to be as close to zero as possible, i.e., vector $\mathbf{b}_i$ to be as parallel to vector $\tilde{\mathbf{b}}_j$ as possible because the Gram-Schmidt orthogonalization vectors are orthogonal to each other.

Condition (b) says that exchanging $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$ and then recomputing the Gram-Schmidt orthogonalization can produce a new shorter vector

$$\tilde{\mathbf{b}}'_{i-1} = \tilde{\mathbf{b}}_i + \mu_{i,i-1}\tilde{\mathbf{b}}_{i-1}$$

but not "too much" shorter as it can be proved.

For any $\delta \in \left(\frac{1}{4}, 1\right)$, the LLL algorithm produces an δ-reduced basis in polynomial time. For $\delta = 1$ we cannot prove that the LLL algorithm terminates in polynomial time.

Definition 46 *We define the* ***auxiliary parameter*** β *as follows:*

$$\beta = \frac{4}{4\delta - 1} \quad \textit{so that} \quad \beta > \frac{4}{3} \quad \textit{and} \quad \frac{1}{\beta} = \delta - \frac{1}{4}$$

For $\delta = \frac{3}{4}$ we obtain $\beta = 2$. A δ-reduced basis has desired properties that we now show.

Proposition 2.2 *Let $\mathcal{L} \subset \mathbb{R}^n$ be a lattice an $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a δ-reduced basis of $\mathcal{L}$, and $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ be its Gram-Schmidt orthogonalization , then*

(a) $\|\mathbf{b}_j\|^2 \le \beta^{i-j}\|\tilde{\mathbf{b}}_i\|^2 \quad$ for $1 \le j < i \le n$

(b) $\det(\mathcal{L}) \le \|\mathbf{b}_1\| \cdots \|\mathbf{b}_n\| \le \beta^{n(n-1)/4}\ det(\mathcal{L})$

(c) $\|\mathbf{b}_1\| \le \beta^{n(n-1)/4}\ det(\mathcal{L})^{1/n}$

Proof.

(a) From the two conditions of Definition 45 we have that

$$\begin{aligned}
\|\tilde{\mathbf{b}}_i\|^2 &\ge \left(\delta - \mu_{i,i-1}^2\right)\|\tilde{\mathbf{b}}_{i-1}\|^2 && \text{for } 2 \le i \le n \\
&\ge \left(\delta - \frac{1}{4}\right)\|\tilde{\mathbf{b}}_{i-1}\|^2 \qquad \text{since } |\mu_{i,i-1}|^2 \le \left(\frac{1}{2}\right)^2 = \frac{1}{4} \text{ and } \frac{1}{4} < \delta < 1 \\
&= \frac{1}{\beta}\|\tilde{\mathbf{b}}_{i-1}\|^2 && \text{Definition 46}
\end{aligned}$$

Therefore $\|\tilde{\mathbf{b}}_{i-1}\|^2 \le \beta\|\tilde{\mathbf{b}}_i\|^2$ meaning for example that $\|\tilde{\mathbf{b}}_{i-2}\|^2$ is at most β times smaller than $\|\tilde{\mathbf{b}}_{i-1}\|^2$ which is at most β times smaller than $\|\tilde{\mathbf{b}}_i\|^2$ and so $\|\tilde{\mathbf{b}}_{i-2}\|^2$ is at most β^2 times smaller than $\|\tilde{\mathbf{b}}_i\|^2$, thus, an easy induction gives

$$\|\tilde{\mathbf{b}}_j\|^2 \le \beta^{j-i}\|\tilde{\mathbf{b}}_i\|^2 \quad \text{for} \quad 1 \le j \le i \le n \tag{2.2}$$

From proof of Theorem 1.11(d) (p. 16) we have that

$$\|\mathbf{b}_i\|^2 = \|\tilde{\mathbf{b}}_i\|^2 + \sum_{j=1}^{i-1} \mu_{ij}^2 \|\tilde{\mathbf{b}}_j\|^2$$

So we have that,

$$\begin{aligned}
\|\mathbf{b}_i\|^2 &= \|\tilde{\mathbf{b}}_i\|^2 + \sum_{j=1}^{i-1} \mu_{ij}^2 \|\tilde{\mathbf{b}}_j\|^2 \\
&\leq \|\tilde{\mathbf{b}}_i\|^2 + \sum_{j=1}^{i-1} \frac{1}{4} \|\tilde{\mathbf{b}}_j\|^2 && \text{since } \mu_{ij}^2 = |\mu_{ij}^2| \leq \left(\frac{1}{2}\right)^2 = \frac{1}{4} \\
&\leq \|\tilde{\mathbf{b}}_i\|^2 + \sum_{j=1}^{i-1} \frac{1}{4} \beta^{i-j} \|\tilde{\mathbf{b}}_i\|^2 && \text{from (2.2)} \\
&= \|\tilde{\mathbf{b}}_i\|^2 \left(1 + \sum_{j=1}^{i-1} \frac{1}{4} \beta^{i-j}\right) \\
&= \|\tilde{\mathbf{b}}_i\|^2 \left(1 + \frac{1}{4} \sum_{j=1}^{i-1} \beta^{i-j}\right)
\end{aligned}$$

Using the summation formula for the geometric sequence $\sum_{j=1}^{i-1} \beta^{i-j}$ we obtain

$$\|\mathbf{b}_i\|^2 \leq \|\tilde{\mathbf{b}}_i\|^2 \left(1 + \frac{1}{4} \frac{\beta^i - \beta}{\beta - 1}\right)$$

We show by induction on i that

$$\left(1 + \frac{1}{4} \frac{\beta^i - \beta}{\beta - 1}\right) \leq \beta^{i-1}$$

hence,

$$\|\mathbf{b}_i\|^2 \leq \beta^{i-1} \|\tilde{\mathbf{b}}_i\|^2 \tag{2.3}$$

The basis case $i = 1$ gives $1 \leq 1$ which holds. For the inductive step we have that

$$\begin{aligned} 1 + \frac{1}{4}\frac{\beta^{i+1} - \beta}{\beta - 1} &\leq \beta^{(i+1)-1} && \Rightarrow \\ 1 + \beta^{i+1} - \beta &\leq \left(4\beta^{i+1} - 4\beta^{i}\right) && \Rightarrow \\ 1 - \beta &\leq 3\beta^{i+1} - 4\beta^{i} && \Rightarrow \\ 0 &\leq \frac{3\beta - 4}{\beta - 1}\beta^{i} \end{aligned}$$

which holds because from Definition 46 we have that $\beta > \frac{4}{3}$, so

$$\beta^{i} > 0 \quad \text{and} \quad 3\beta - 4 > 0 \quad \text{and} \quad \beta - 1 > 0$$

Combining (2.2) and (2.3) we have that

$$\|\tilde{\mathbf{b}}_j\|^2 \leq \beta^{j-1}\|\tilde{\mathbf{b}}_j\|^2 \leq \beta^{i-1}\|\tilde{\mathbf{b}}_i\|^2 \quad \text{for} \quad 1 \leq j \leq i \leq n$$

which proves (a).

(b) From Hadamard's inequality (Corollary 1.12, p. 18) we know that

$$det(\mathcal{L}) = \|\tilde{\mathbf{b}}_1\| \cdot \|\tilde{\mathbf{b}}_2\| \cdots \|\tilde{\mathbf{b}}_n\| \leq \|\mathbf{b}_1\| \cdot \|\mathbf{b}_2\| \cdots \|\mathbf{b}_n\|$$

which proves the left inequality in part (b). From (2.3) by taking the product over $i = 1, \ldots, n$ we have that

$$\begin{aligned} \|\mathbf{b}_1\|^2 \cdot \|\mathbf{b}_2\|^2 \cdots \|\mathbf{b}_n\|^2 &\leq \beta^{0+1+2+\cdots+(n-1)} \ \|\tilde{\mathbf{b}}_1\|^2 \cdot \|\tilde{\mathbf{b}}_2\|^2 \cdots \|\tilde{\mathbf{b}}_n\|^2 && \Rightarrow \\ \|\mathbf{b}_1\|^2 \cdot \|\mathbf{b}_2\|^2 \cdots \|\mathbf{b}_n\|^2 &\leq \beta^{n(n-1)/2} \ \|\tilde{\mathbf{b}}_1\|^2 \cdot \|\tilde{\mathbf{b}}_2\|^2 \cdots \|\tilde{\mathbf{b}}_n\|^2 && \Rightarrow \\ \sqrt{\|\mathbf{b}_1\|^2 \cdot \|\mathbf{b}_2\|^2 \cdots \|\mathbf{b}_n\|^2} &\leq \sqrt{\beta^{n(n-1)/2}} \ \sqrt{\|\tilde{\mathbf{b}}_1\|^2 \cdot \|\tilde{\mathbf{b}}_2\|^2 \cdots \|\tilde{\mathbf{b}}_n\|^2} && \Rightarrow \\ \|\mathbf{b}_1\| \cdot \|\mathbf{b}_2\| \cdots \|\mathbf{b}_n\| &\leq \left(\beta^{n(n-1)/2}\right)^{1/2} \ \|\tilde{\mathbf{b}}_1\| \cdot \|\tilde{\mathbf{b}}_2\| \cdots \|\tilde{\mathbf{b}}_n\| && \Rightarrow \\ \|\mathbf{b}_1\| \cdot \|\mathbf{b}_2\| \cdots \|\mathbf{b}_n\| &\leq \beta^{n(n-1)/4} \, \|\tilde{\mathbf{b}}_1\| \cdot \|\tilde{\mathbf{b}}_2\| \cdots \|\tilde{\mathbf{b}}_n\| = \beta^{n(n-1)/4} \, det(\mathcal{L}) && \Rightarrow \\ \|\mathbf{b}_1\| \cdot \|\mathbf{b}_2\| \cdots \|\mathbf{b}_n\| &\leq \beta^{n(n-1)/4} \, det(\mathcal{L}) \end{aligned}$$

which proves the right inequality in part (b).

(c) Setting $j = 1$ in part (a) gives

$$\|\mathbf{b}_1\| \leq \beta^{i-1}\|\tilde{\mathbf{b}}_i\| \quad \text{for } 1 \leq i \leq n$$

and taking product over $i = 1, \ldots, n$ we have that

$$\begin{aligned}
\overbrace{\|\mathbf{b}_1\|^2 \cdot \|\mathbf{b}_1\|^2 \cdots \|\mathbf{b}_1\|^2}^{n\text{-times}} \leq \beta^{0+1+2+\cdots+(n-1)} \; \|\tilde{\mathbf{b}}_1\|^2 \cdot \|\tilde{\mathbf{b}}_2\|^2 \cdots \|\tilde{\mathbf{b}}_n\|^2 & \quad \Rightarrow \\
\|\mathbf{b}_1\|^{2n} \leq \beta^{n(n-1)/2} \; \|\tilde{\mathbf{b}}_1\|^2 \cdot \|\tilde{\mathbf{b}}_2\|^2 \cdots \|\tilde{\mathbf{b}}_n\|^2 & \quad \Rightarrow \\
\sqrt{\|\mathbf{b}_1\|^{2n}} \leq \sqrt{\beta^{n(n-1)/2}} \; \sqrt{\|\tilde{\mathbf{b}}_1\|^2 \cdot \|\tilde{\mathbf{b}}_2\|^2 \cdots \|\tilde{\mathbf{b}}_n\|^2} & \quad \Rightarrow \\
\|\mathbf{b}_1\|^n \leq \left(\beta^{n(n-1)/2}\right)^{1/2} \; \|\tilde{\mathbf{b}}_1\| \cdot \|\tilde{\mathbf{b}}_2\| \cdots \|\tilde{\mathbf{b}}_n\| = \beta^{n(n-1)/4} \, det(\mathcal{L}) & \quad \Rightarrow \\
\sqrt[n]{\|\mathbf{b}_1\|^n} \leq \sqrt[n]{\beta^{n(n-1)/4}} \; \sqrt[n]{det(\mathcal{L})} & \quad \Rightarrow \\
\|\mathbf{b}_1\| \leq \beta^{(n-1)/4} \; det(\mathcal{L})^{1/n} &
\end{aligned}$$

which proves part (c).

□

The upper bound for $\mathbf{b}_1$ in the next result is exponential, but it depends only on δ and the dimension n, so it applies uniformly to all lattices of dimension n.

Theorem 2.3 (*LLL theorem*) *Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ be a δ-reduced basis of a lattice $\mathcal{L} \subset \mathbb{R}^n$. Then for any nonzero vector $\mathbf{v} \in \mathcal{L}$ we have that*

$$\|\mathbf{b}_1\| \leq \beta^{(n-1)/2} \; \|\mathbf{v}\|$$

In particular, $\mathbf{b}_1$ is no longer than $\beta^{(n-1)/2}$ times the shortest vector in $\mathcal{L}$.

Proof. Let $\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2, \ldots, \tilde{\mathbf{b}}_n$ be the Gram-Schmidt orthogonalization of $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$. Setting $j = 1$ in Proposition 2.2(a) gives

$$\begin{aligned}
&\|\mathbf{b}_1\|^2 \leq \beta^{i-1} \; \|\tilde{\mathbf{b}}_i\| \Rightarrow \|\tilde{\mathbf{b}}_i\|^2 \geq \frac{1}{\beta^{i-1}} \|\mathbf{b}_1\|^2 \quad \text{for } 1 \leq i \leq n \qquad \Rightarrow \\
&\sqrt{\|\tilde{\mathbf{b}}_i\|^2} \geq \sqrt{\frac{1}{\beta^{i-1}}} \sqrt{\|\mathbf{b}_1\|^2} \quad \text{for } 1 \leq i \leq n \qquad \Rightarrow \\
&\|\tilde{\mathbf{b}}_i\| \geq \frac{1}{\beta^{(i-1)/2}} \|\mathbf{b}_1\| \quad \text{for } 1 \leq i \leq n
\end{aligned}$$

Theorem 1.14 (p. 20) shows that for any nonzero vector $\mathbf{v} \in \mathcal{L}$

$$\|\mathbf{v}\| \geq \min_{1 \leq i \leq n} \|\tilde{\mathbf{b}}_i\| \geq \frac{1}{\beta^{(n-1)/2}} \|\mathbf{b}_1\|$$

and this completes the proof. □

There is a stronger result that gives upper bounds for the lengths of all the vectors in a δ-reduced basis.

Theorem 2.4 *Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ be a δ-reduced basis of a lattice $\mathcal{L} \subset \mathbb{R}^n$, and let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ be any m linearly independent vectors in $\mathcal{L}$. Then for $1 \leq j \leq m$ we have*

$$\|\mathbf{b}_j\| \leq \beta^{(n-1)/2}\, max\{\|\mathbf{v}_1\|, \|\mathbf{v}_2\|, \ldots, \|\mathbf{v}_m\|\}$$

Proof. We write each $\mathbf{v}_j$ as integral linear combination of the basis vectors,

$$\mathbf{v}_j = \sum_{i=1}^{n} r_{ij}\mathbf{b}_i \quad \text{with } r_{ij} \in \mathbb{Z},\ 1 \leq i \leq n,\ 1 \leq j \leq m$$

and for fixed j let $i(j)$ denote the largest i for which $r_{ij} \neq 0$. From the definition of Gram-Schmidt orthogonalization (Definition 21, p. 15) we have that

$$\tilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{k=1}^{i-1} \mu_{ik}\tilde{\mathbf{b}}_k \Rightarrow \mathbf{b}_i = \sum_{k=1}^{i} \mu_{ik}\tilde{\mathbf{b}}_k$$

therefore,

$$\mathbf{v}_j = \sum_{i=1}^{n} r_{ij}\mathbf{b}_i = \sum_{i=1}^{i(j)} r_{ij} \sum_{k=1}^{i} \mu_{ik}\tilde{\mathbf{b}}_k = \sum_{k=1}^{i(j)} \tilde{\mathbf{b}}_k \sum_{i=k}^{i(j)} r_{ij}\mu_{ik}$$

If we take the norm of both sides, because $\tilde{\mathbf{b}}_k$ are orthogonal we get that

$$\|\mathbf{v}_j\|^2 = \left\| \sum_{k=1}^{i(j)} \tilde{\mathbf{b}}_k \sum_{i=k}^{i(j)} r_{ij}\mu_{ik} \right\|^2 = \sum_{k=1}^{i(j)} \|\tilde{\mathbf{b}}_k\|^2 \sum_{i=k}^{i(j)} |r_{ij}\mu_{ik}|^2$$

For each $\tilde{\mathbf{b}}_k$ every term in the sum is nonnegative therefore for $k = i(j)$ observing that $\mu_{i(j),i(j)} = 1$ and $|r_{i(j),j}| \geq 1$ because $r_{i(j),j} \in \mathbb{Z}$ and $r_{i(j),j} \neq 0$, we have that

$$\|\mathbf{v}_j\|^2 \geq \|\tilde{\mathbf{b}}_{i(j)}\|^2 \quad \text{for } 1 \leq j \leq m \tag{2.4}$$

If $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ is an unordered set, then we may assume without loss of generality that

$$i(1) \leq i(2) \leq \cdots \leq i(j)$$

else we renumber each $\mathbf{v}_i$ for this property to hold.
We claim that $j \leq i(j)$ for $1 \leq j \leq m$. If not, then for some j with $i(j) < j$, the vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_j$ would all belong to the linear span of $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_j$, a

contradiction with the linear independence of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$. Combining Proposition 2.2(a) with $i = i(j)$ and (2.4) we get that

$$\|\mathbf{b}_j\|^2 \leq \beta^{i(j)-1} \|\tilde{\mathbf{b}}_{i(j)}\|^2 \leq \beta^{n-1} \|\tilde{\mathbf{b}}_{i(j)}\|^2 \leq \beta^{n-1} \|\mathbf{v}_j\|^2 \quad \text{for } 1 \leq j \leq m$$

Taking the square root of both sides gives

$$\|\mathbf{b}_j\| \leq \beta^{(n-1)/2} \|\mathbf{v}_j\| \leq \beta^{(n-1)/2} max\{\|\mathbf{v}_1\|, \|\mathbf{v}_2\|, \ldots, \|\mathbf{v}_m\|\} \quad \text{for } 1 \leq j \leq m$$

and this completes the proof. □

Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a δ-reduced basis of the lattice $\mathcal{L} \in \mathbb{R}^n$, and let $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ be its Gram-Schmidt orthogonalization . From Proposition 2.2(a) and Theorem 1.11(d), (p. 16) we have that

$$\begin{aligned} \|\mathbf{b}_j\|^2 \leq \beta^{i-1}\|\tilde{\mathbf{b}}_i\|^2 \quad \text{for } 1 \leq j \leq i \leq n \quad &\Rightarrow \\ \beta^{1-i}\|\mathbf{b}_j\|^2 \leq \|\tilde{\mathbf{b}}_i\|^2 \leq \|\mathbf{b}_i\|^2 \quad &\Rightarrow \\ \beta^{1-i} max\{\|\mathbf{b}_1\|^2, \|\mathbf{b}_2\|^2, \ldots, \|\mathbf{b}_i\|^2\} \leq \|\mathbf{b}_i\|^2 \quad \text{for } 1 \leq i \leq n & \end{aligned}$$

From the last inequality and Theorem 2.4 for $1 \leq i \leq n$ we have that

$$\beta^{1-i}\, max\{\|\mathbf{b}_1\|^2, \ldots, \|\mathbf{b}_i\|^2\} \leq \|\mathbf{b}_i\|^2 \leq \beta^{n-1}\, max\{\|\mathbf{b}_1\|^2, \ldots, \|\mathbf{b}_i\|^2\} \quad (2.5)$$

Now suppose that $\mathbf{v}_1 = \lambda_1(\mathcal{L}), \mathbf{v}_2 = \lambda_2(\mathcal{L}), \ldots, \mathbf{v}_i = \lambda_i(\mathcal{L})$ achieve the $i-th$ successive minimum and therefore are linearly independent. Clearly,

$$max\{\mathbf{v}_1, \ldots, \mathbf{v}_i\} \leq max\{\mathbf{b}_1, \ldots, \mathbf{b}_i\}$$

Using the last inequality for the leftmost term in (2.5) and Theorem 2.4 for the rightmost term and taking square roots, we obtain

$$\beta^{(1-i)/2}\, max\{\mathbf{v}_1, \ldots, \mathbf{v}_i\} \leq \|\mathbf{b}_i\| \leq \beta^{(n-1)/2}\, max\{\mathbf{v}_1, \ldots, \mathbf{v}_i\}$$

This can also be written as

$$\beta^{(1-n)/2}\, \|\mathbf{b}_i\| \leq max\{\mathbf{v}_1, \ldots, \mathbf{v}_i\} \leq \beta^{(i-1)/2}\, \|\mathbf{b}_i\|$$

This shows that $\|\mathbf{b}_i\|$ can be regarded as an approximation to the $i-th$ successive minimum of a lattice because the successive minima are weakly increasing:

$$\lambda_1(\mathcal{L}) \leq \lambda_2(\mathcal{L}) \leq \cdots \leq \lambda_i(\mathcal{L})$$

The algorithm presented next is the original LLL lattice reduction algorithm.

Algorithm 3: LLL lattice basis reduction.

Input : A parameter δ and basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ for the lattice $\mathcal{L} \subset \mathbb{R}^n$.
Output: A δ-reduced basis for the lattice $\mathcal{L}$.

1 $k \leftarrow 2$;
2 $\tilde{\mathbf{b}}_1 \leftarrow \mathbf{b}_1$;
3 **while** $k \leq n$ **do**
4   **for** $j = 1, 2, \ldots, k-1$ **do**
5   compute $\tilde{\mathbf{b}}_j$;
6         $\mathbf{b}_k \leftarrow \mathbf{b}_k - \lfloor \mu_{kj} \rceil \tilde{\mathbf{b}}_j$ `/* size reduction */;`
7   **end**
8   **if** $\|\tilde{\mathbf{b}}_k\|^2 \geq \left(\delta - \mu_{k,k-1}^2\right) \|\tilde{\mathbf{b}}_{k-1}\|^2$ **then** `/* Lovász condition */`
9         $k \leftarrow k+1$;
10   **else**
11   swap $\mathbf{b}_{k-1}$ and $\mathbf{b}_k$ `/* swap step */;`
12         $k \leftarrow max(k-1, 2)$;
13   **end**
14 **end**
15 **return** $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$;

At line 5 the vector $\tilde{\mathbf{b}}_j$ is obtained by applying Gram-Schmidt orthogonalization . For efficiency reasons the Gram-Schmidt orthogonalization can be done once before the main loop at line 3. Then if a size reduction (line 6) or swap (line 11) is done, we update the Gram-Schmidt orthogonalization coefficients accordingly (see [51] for details). At line 8 the size check is performed on the orthogonal projections of $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$ on the orthogonal of $span(\mathbf{b}_1, \ldots, \mathbf{b}_{k-2})$ in order to see if an iteration of the t-Gauss algorithm is necessary for $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$ (see [71], ch. 3).

Theorem 2.5 *Let* $\mathbf{b}_1, \ldots, \mathbf{b}_n$ *be a basis of a lattice* $\mathcal{L}$ *and* $\delta \in \left(\frac{1}{4}, 1\right)$. *Then the LLL algorithm (Algorithm 3) terminates in a polynomial number of step and returns a* δ*-reduced basis.*

Proof.*(sketch)*

For simplicity we consider $\delta = \frac{3}{4} \Rightarrow \beta = 2$ and $\mathcal{L} \subseteq \mathbb{Z}^n$.

Both the *for loop* at lines 4-7 and the fact that in order for the algorithm to terminate at line 9 we must have $k = n + 1$ therefore all vectors must pass the Lovász condition test at line 8, ensure that if the algorithm terminates then the basis returned satisfies the size condition and the Lovász condition respectively. So we have to show that the algorithm terminates.

Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a basis for $\mathcal{L}$, let $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ be its Gram-Schmidt orthogonalization , and for each $\ell = 1, \ldots, n$ let $\mathcal{L}_\ell$ be the lattice spanned by $\mathbf{b}_1, \ldots, \mathbf{b}_\ell$, i.e., $\mathcal{L}_\ell = \mathcal{L}(\{\mathbf{b}_1, \ldots, \mathbf{b}_\ell\})$.

We define the quantities d_ℓ and D as

$$d_\ell = \prod_{i=1}^{\ell} \|\tilde{\mathbf{b}}_i\|^2 \quad \text{and} \quad D = \prod_{\ell=1}^{n} d_\ell = \prod_{i=1}^{n} \|\tilde{\mathbf{b}}_i\|^{2(n+1-i)}$$

From Remark 1.10 (p. 16) we have that

$$det(\mathcal{L}_\ell) = \prod_{i=1}^{\ell} \|\tilde{\mathbf{b}}_i\| \Rightarrow \prod_{i=1}^{\ell} \|\tilde{\mathbf{b}}_i\|^2 = det(\mathcal{L}_\ell)^2 = d_\ell$$

During the execution of the algorithm, d_ℓ changes only if the swap step at line 11 is executed and that is when the value of D also changes. More precisely, only for $\ell = k-1$ the value of d_ℓ changes because only the values of $\tilde{\mathbf{b}}_{k-1}$ and $\tilde{\mathbf{b}}_k$ change. That is, for $\ell < k-1$ the terms $\tilde{\mathbf{b}}_{k-1}$ and $\tilde{\mathbf{b}}_k$ are not included in ℓ, and for $\ell \geq k$ both terms are included so if we swap them the product remains the same. To estimate that change in d_{k-1} note that the Lovász condition check fails at line 8, so we have

$$\|\tilde{\mathbf{b}}_k\|^2 < \left(\frac{3}{4} - \mu_{k,k-1}^2\right) \|\tilde{\mathbf{b}}_{k-1}\|^2 \leq \frac{3}{4} \|\tilde{\mathbf{b}}_{k-1}\|^2$$

and when we swap $\tilde{\mathbf{b}}_{k-1}$ and $\tilde{\mathbf{b}}_k$ we get a new d_{k-1} value

$$\begin{aligned}
d_{k-1}^{new} &= \|\tilde{\mathbf{b}}_1\|^2 \cdots \|\tilde{\mathbf{b}}_{k-2}\|^2 \cdot \|\tilde{\mathbf{b}}_k\|^2 \\
&= \|\tilde{\mathbf{b}}_1\|^2 \cdots \|\tilde{\mathbf{b}}_{k-2}\|^2 \cdot \frac{\|\tilde{\mathbf{b}}_{k-1}\|^2 \cdot \|\tilde{\mathbf{b}}_k\|^2}{\|\tilde{\mathbf{b}}_{k-1}\|^2} \\
&= d_{k-1}^{old} \cdot \frac{\|\tilde{\mathbf{b}}_k\|^2}{\|\tilde{\mathbf{b}}_{k-1}\|^2} \\
&\leq \frac{3}{4} d_{k-1}^{old}
\end{aligned}$$

Therefore if the swap step at line 11 is executed c times, the value of D is reduced by a factor of at least $\left(\frac{3}{4}\right)^c$, since each swap reduces the value of some ℓ by at least $\frac{3}{4}$.

Because $\mathcal{L} \subseteq \mathbb{Z}^n$ then $\lambda_1(\mathcal{L}) \geq 1$, and by Theorem 1.17 (p. 23) we have that,

$$\begin{aligned} 1 \leq \lambda_1(\mathcal{L}_\ell) \leq \sqrt{\ell}\ det(\lambda_1(\mathcal{L}_\ell))^{1/\ell} & \qquad \Rightarrow \\ 1 \leq \ell^{\ell/2}\ det(\lambda_1(\mathcal{L}_\ell)) & \qquad \Rightarrow \\ \ell^{-\ell/2} \leq det(\lambda_1(\mathcal{L}_\ell)) & \qquad \Rightarrow \\ \ell^{-\ell} \leq det(\lambda_1(\mathcal{L}_\ell))^2 & \end{aligned}$$

and thus, the product over all ℓ gives a lower bound for D (which is independent of each iteration of the algorithm),

$$D = \prod_{\ell=1}^{n} d_\ell \geq \prod_{\ell=1}^{n} \ell^{-\ell} \geq \prod_{\ell=1}^{n} \ell^{-n} = (n!)^{-n} \geq n^{-n} \geq n^{-n^2} > 0$$

At each iteration of the algorithm either we increase k by one at line 9, or we decrease it at line 12 after a swap is made. If we prove that the number of times that we decrease k is finite, say m, then we know that after m iterations the value of k will increase until it reaches the value of $n+1$ and the algorithm terminates.

Suppose that the number of times that the swap step is executed, which is c, is infinite. Then because the value of D is reduced by a factor of at least $\left(\frac{3}{4}\right)^c$ we have that

$$\lim_{c\to\infty} \left(\frac{3}{4}\right)^c = 0 \quad \text{because } \frac{3}{4} < 1 \quad \text{therefore } D = 0 \text{ as } c \to \infty$$

a contradiction because we have that $D \geq n^{-n^2} > 0$. This proves that the LLL algorithm terminates in a finite number of iterations.

We now give a upper bound for the run-time complexity. Let D_{init} denote the initial value of D for the original basis, let D_{final} denote the value of D for the basis that the algorithm return when it terminates, and as above, let c denote the number of times that the swap step at line 11 is executed. Notice that the *While* loop at line 3 is executed at most $2c+n$ times,so it suffices to find a bound for c. From the lower bound on D we have that

$$0 < n^{-n^2} \leq D_{final} \leq \left(\frac{3}{4}\right)^c D_{init}$$

Since $log\left(\frac{3}{4}\right) < 1$, by taking logarithms we have that

$$c = O\left(n^2 log(n) + log\left(D_{init}\right)\right)$$

To estimate D_{init} we have that

$$\begin{aligned}
D_{init} &= \prod_{i=1}^{n} \|\tilde{\mathbf{b}}_i\|^{2(n+1-i)} \\
&\leq \prod_{i=1}^{n} \|\mathbf{b}_i\|^{2(n+1-i)} && \text{because } \|\tilde{\mathbf{b}}_i\| \leq \|\mathbf{b}_i\| \\
&\leq \prod_{i=1}^{n} \left(\underset{1\leq i\leq n}{max} \|\mathbf{b}_i\| \right)^{2(n+1-i)} \\
&= \left(\underset{1\leq i\leq n}{max} \|\mathbf{b}_i\| \right)^{2(n+(n-1)+\cdots+1)} \\
&= \left(\underset{1\leq i\leq n}{max} \|\mathbf{b}_i\| \right)^{n^2+n}
\end{aligned}$$

therefore,

$$log\left(D_{init}\right) = O\left(n^2 log \left(\underset{1\leq i\leq n}{max} \|\mathbf{b}_i\| \right) \right)$$

from which we conclude that c is polynomial in the input size, and thus the LLL algorithm runs in polynomial time. $\square$

Let $B = \underset{1\leq i\leq n}{max} \|\mathbf{b}_i\|$.

It is proven in [51] that the number of bit operations needed by the LLL algorithm if we use the classical algorithms for arithmetic operations is $O\left(n^6 (logB)^3\right)$, which can be reduced to $O\left(n^{5+\epsilon}(logB)^{2+\epsilon}\right)$ for every $\epsilon > 0$, if we employ fast multiplication techniques.

The complexity can be improved using floating point numbers instead of rationals except for the basis vectors that are kept as integers, because is someone tries to keep the exact integer values of an integer lattice, as the dimension grows the intermediate calculations involve enormous number, thus it is generally necessary to use floating point approximations. Unfortunately, this is known to be unstable in the worst-case: the usual floating point LLL algorithm is not even guaranteed to terminate, and the output basis may not be reduced at all.

There have been many improvements to and generalization of the LLL algorithm. Some of them are described in [78], [74], [79] and [80].

From a theoretical point of view for a lattice of rank r and dimension n the fastest algorithm for for lattice reduction is described in [69] and has run-time complexity $O\left(nr^4(logB)^2\right)$.

2.5 Babai's algorithm

In this section we follow Babai [9] to show how the LLL algorithm can be used to find a good approximation of the closest vector problem (CVP).

Babai proposed two approximation algorithms to solve CVP. We consider Babai's “nearest plane” algorithm. The other one is the “round-off” algorithm. Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a basis for the lattice $\mathcal{L} \subset \mathbb{R}^n$, let $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ be its Gram-Schmidt orthogonalization .

Let

$$V = \sum_{i=1}^{n-1} r_i \mathbf{b}_i \quad \text{with } r_i \in \mathbb{R} \quad \text{for } 1 \leq i \leq n-1$$

be the linear subspace (hyperplane) generated by $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}$, and let

$$\mathcal{L}_{n-1} = \sum_{i=1}^{n-1} a_i \mathbf{b}_i \quad \text{with } a_i \in \mathbb{Z} \quad \text{for } 1 \leq i \leq n-1$$

be the corresponding sublattice, i.e., $\mathcal{L}_{n-1} = V \cap \mathcal{L}$.

We consider the following translations of V

$$V + \mathbf{x} = \{\mathbf{v} + \mathbf{x} : \mathbf{v} \in V\} \quad \text{with } \mathbf{x} \in \mathcal{L}$$

Given an arbitrary vector $\mathbf{t} \in \mathbb{R}^n$, the nearest plane algorithm says that we should find the vector $\mathbf{x} \in \mathcal{L}$ for which the orthogonal $dist(\mathbf{t}, V + \mathbf{x})$ is minimized. For this we use the following recursive procedure. We write $\mathbf{t}$ as a linear combination of $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$, i.e.,

$$\mathbf{t} = \sum_{i=1}^{n} c_i \tilde{\mathbf{b}}_i \quad \text{with } c_i \in \mathbb{R} \quad \text{for } 1 \leq i \leq n$$

define $\mathbf{w} = \lfloor c_n \rceil \mathbf{b}_n$ and $\mathbf{t}^\perp$ as

$$\mathbf{t}^\perp = \left(\sum_{i=1}^{n-1} c_i \tilde{\mathbf{b}}_i \right) + \mathbf{w}$$

Then $\mathbf{t}^\perp$ is the orthogonal projection of $\mathbf{t}$ onto the translated hyperplane $V + \mathbf{w}$. We have that $\mathbf{t}^\perp - \mathbf{w} \in V$, so recursively find the vector $\mathbf{x}_{n-1} \in \mathcal{L}_{n-1}$ closest to $\mathbf{t}^\perp - \mathbf{w}$ and set $\mathbf{x} = \mathbf{x}_{n-1} + \mathbf{w}$.

Theorem 2.6 (Babai's theorem) *Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a $\frac{3}{4}$-reduced LLL basis for the lattice $\mathcal{L} \subset \mathbb{R}^n$, and let $\mathbf{t} \in \mathbb{R}^n$ be an arbitrary vector. Then the lattice vector $\mathbf{x} \in \mathcal{L}$ produced by the nearest plane algorithm satisfies*

$$\|\mathbf{t} - \mathbf{x}\| \leq 2^{n/2} \|\mathbf{t} - \mathbf{v}\|$$

where $\mathbf{v} \in \mathcal{L}$ is the closest lattice vector to $\mathbf{t}$.

Proof. For $n = 1$ we find the closest integer multiple of one nonzero real number to another real number, which is the closest lattice vector.

For $n \geq 2$ we use induction on n. Observe that

$$\|\mathbf{t} - \mathbf{t}^{\perp}\| \leq \frac{\|\tilde{\mathbf{b}}_n\|}{2} \Rightarrow \|\mathbf{t} - \mathbf{t}^{\perp}\|^2 \leq \frac{\|\tilde{\mathbf{b}}_n\|^2}{4} \tag{2.6}$$

and

$$\|\mathbf{t} - \mathbf{t}^{\perp}\| \leq \|\mathbf{t} - \mathbf{v}\| \tag{2.7}$$

because the hyperplanes $V + \mathbf{x}$ where $\mathbf{x} \in \mathcal{L}$ are spaced at distance $\|\tilde{\mathbf{b}}_n\|$, and $\|\mathbf{t} - \mathbf{t}^{\perp}\|$ is the distance of $\mathbf{t}$ for the nearest such hyperplane.

From (2.6) with induction (corresponding to the recursion of the algorithm) we obtain

$$\|\mathbf{t} - \mathbf{x}\| \leq \frac{1}{4}\left(\|\tilde{\mathbf{b}}_1\|^2 + \cdots + \|\tilde{\mathbf{b}}_n\|^2\right) \tag{2.8}$$

Proposition 2.2a, (p. 40), for $i = n$ and for $\beta = 2$ since $\delta = \frac{3}{4}$, gives

$$\frac{1}{4}\sum_{i=1}^{n} \|\tilde{\mathbf{b}}_i\|^2 \leq \frac{1}{4}\sum_{i=1}^{n}\left(2^{n-i}\|\tilde{\mathbf{b}}_n\|^2\right) = \frac{1}{4}\left(2^n - 1\right)\|\tilde{\mathbf{b}}_n\|^2 < 2^{n-2}\|\tilde{\mathbf{b}}_n\|^2 \tag{2.9}$$

Combining (2.8) and (2.9) we get that

$$\|\mathbf{t} - \mathbf{x}\|^2 < 2^{n-2}\|\tilde{\mathbf{b}}_n\|^2 \Rightarrow \|\mathbf{t} - \mathbf{x}\| < 2^{\frac{n}{2}-1}\|\tilde{\mathbf{b}}_n\| \tag{2.10}$$

We now have to consider two cases, corresponding to whether the closest vector $\mathbf{v} \in \mathcal{L}$ does or does not belong to $V + \mathbf{w}$.

(a) Case ($\mathbf{v} \in V + \mathbf{w}$):
In this case $\mathbf{v} - \mathbf{w} \in \mathcal{L}$ is the closest vector to the sublattice $\mathcal{L}_{n-1}$ to the vector $\mathbf{t}^{\perp} - \mathbf{w} \in V$. Therefore the inductive hypothesis gives

$$\begin{aligned}
\|\mathbf{t}^{\perp} - \mathbf{x}\| &= \|\mathbf{t}^{\perp} - (\mathbf{x}_{n-1} + \mathbf{w})\| \\
&\leq 2^{(n-1)/2}\|\mathbf{t}^{\perp} - (\mathbf{v} - \mathbf{w} + \mathbf{w})\| \\
&= 2^{(n-1)/2}\|\mathbf{t}^{\perp} - \mathbf{v}\| \\
&\leq 2^{(n-1)/2}\|\mathbf{t} - \mathbf{v}\|
\end{aligned}$$

Combining this with (2.7) we have that

$$\begin{aligned}
\|\mathbf{t}-\mathbf{x}\| &= \sqrt{\|(\mathbf{t}-\mathbf{t}^{\perp})\|^2 + \|(\mathbf{t}^{\perp}-\mathbf{x})\|^2} \\
&\leq \sqrt{\|(\mathbf{t}-\mathbf{v})\|^2 + 2^{n-1}\|(\mathbf{t}^{\perp}-\mathbf{v})\|^2} \\
&= 2^{n/2}\|(\mathbf{t}^{\perp}-\mathbf{v})\| \\
&\leq 2^{n/2}\|(\mathbf{t}-\mathbf{v})\|
\end{aligned}$$

Thus, $\|\mathbf{t}-\mathbf{x}\| \leq 2^{n/2}\|\mathbf{t}-\mathbf{v}\|$.

(b) Case ($\mathbf{v} \notin V + \mathbf{w}$):
In this case we must have

$$\|\mathbf{t}-\mathbf{v}\| \geq \frac{\|\tilde{\mathbf{b}}_n\|}{2}$$

Combining this with (2.10) we again have that $\|\mathbf{t}-\mathbf{x}\| < 2^{n/2}\|\mathbf{t}-\mathbf{v}\|$ and this completes the proof.

□

It is clear that the next algorithm runs in polynomial time.

Algorithm 4: Babai's nearest plane algorithm.

Input : A $\frac{3}{4}$-reduced LLL basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ for the lattice $\mathcal{L} \subseteq \mathbb{Z}^n$, its $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_n$ Gram-Schmidt orthogonalization , and a target vector $\mathbf{t} \in \mathbb{Z}^n$

Output: A vector in $\mathcal{L}$ that is closer to **t** within an approximation ration of $2^{n/2}$.

```
v ← t;
for i = n, ..., 1 do
    v ← v − ⌈⟨v, b̃_i⟩ / ||b̃_i||²⌋ b_i;
end
return (t − v);
```

Lattice reduction algorithms and Babai's algorithm have been used for cryptanalysis of various knapsack based schemes, the *Goldreich-Goldwasser-Halevi* cryptosystem [29] and the *NTRU signature* scheme [41].

For cryptanalysis of knapsack based schemes see the survey papers [45] (also cryptanalysis of Knuth's truncated linear congruential generators), [67] and [73], for the Goldreich-Goldwasser-Halevi cryptosystem see [66] and for NTRU signatures see [68].

3

Complexity of lattice problems

In this chapter we present some complexity results for lattice problems. A lot of reductions for lattice problems use the (decisional) subset sum problem which is a known NP-complete problem (see [25]).

Definition 47 *Given* $a_1, a_2, \ldots, a_n, b \in \mathbb{N}$ *decide whether there exist* $x_1, x_2, \ldots, x_n \in \{0, 1\}$ *such that*

$$\sum_{i=1}^{n} a_i x_i = b$$

In the next two sections we give complexity results for SVP and CVP. Without giving any details we must note that the decision versions of both the Shortest Independent Vectors Problem and the Shortest Basis Problem are NP-complete. The decision version of the Closest Vector Problem with Preprocessing is NP-complete in the following sense: there is a polynomial time reduction from a SAT instance ϕ to CVPP instance $(\mathbf{t}, \mathcal{L}(\mathbf{B}))$ such as the lattice $\mathcal{L}\mathbf{B}$ depends only on $|\phi|$ and not ϕ itself. This implies that if there is a polynomial time algorithm for CVPP, the SAT has polynomial size circuits and thus, the polynomial time hierarchy collapses. The Closest Radius Problem is in Π_2 but not known to be NP-hard. See [71, ch.14] and [61, ch.7] for more on these problems.

3.1 Shortest vector problem

SVP is the most famous and widely studied problem for lattices. The NP-hardness of SVP in the Euclidean norm was conjectured by Peter van Emde Boas in 1981 [84], and remained an open problem until 1998, for almost twenty years, when Ajtai [3] proved that solving SVP exactly is NP-hard under *randomized* reductions.

Immediately following Ajtai's breakthrough work, the problem received renewed attention. In [3], Ajtai had already observed that hardness for the exact version implies weak inapproximability results for approximation factors of the form $1 + 1/2^{n^c}$ and this was slightly improved by Cai and Nerurkar [85] to factors $1 + 1/n^c$, where n is the lattice dimension, still approaching 1 as the lattice dimension grows but at a slower rate. Micciancio [56] significantly strengthened Ajtai's result by showing NP-hardness for SVP by a reduction from a variant of CVP for any constant factor smaller than $\sqrt{2}$ (as we will see later, CVP is known to be NP-hard).

The strongest inapproximability results to date are from Khot [48] who showed that SVP is NP-hard to approximate within any constant factor $O(1)$, and from Haviv and Regev [37] who showed that SVP cannot be approximated within some factor $n^{1/O(log\, log\, n)}$ unless NP is in random subexponential time, i.e., $NP \subseteq RSUBEXP = \cap_{\delta>0} RTIME\left(2^{n^\delta}\right)$.

However, all of the above results employ randomization, and little progress has been made towards a deterministic reduction. In fact, the most recent and quantitatively strongest results [48, 37] achieve larger approximation factors than [56] at the cost on introducing even more randomness, have two-sided error whereas [56] has one-sided error, and due to their construct they seem more difficult to derandomize. In 2012, Micciancio [60] presented a new and simpler proof that SVP is NP-hard to approximate within some constant factor and that SVP cannot be approximated within some factor $n^{1/O(log\, log\, n)}$ unless NP is in random subexponential time, and thus matching the best currently known results [48, 37], but under probabilistic reductions with one-sider error.

Proving that SVP is NP-hard under *deterministic* reductions is still an open problem for both the exact and the approximate version of the problem.

In general, there are three approaches to solve SVP: enumeration algorithms, probabilistic sieving algorithms and Voronoi cell based algorithms. The majority of the algorithmic work on SVP and CVP has focused on the ℓ_2 norm and therefore there has been a lot of progress for the ℓ_2 norm, progress on the more general norms has been much slower. For some practical problems the solution strategy is to approximate the problem via a reduction to the ℓ_2 norm but in some cases the error introduced by such a reduction yields unusable results or worst case runtime. Because of the practical interest in SVP we use the experimental values of the constants for run-time complexity.

Enumeration algorithms in [35, 75], solve SVP in the ℓ_2 norm deterministically in asymptotic time $2^{O(n\, logn)}$ where n is the dimension of the lattice. These algorithms do an exhaustive search by exploring all lattice vectors of a bounded search region and require polynomial space. Enumeration algorithms can be rendered probabilistic using an extreme pruning strategy [24], which allows for an exponential speedup and makes enumeration the fastest algorithm for solving SVP in practice. Furthermore the parallelization of enumeration algorithms has been investigated in [39, 19].

Sieving algorithms were first presented in 2001 by Ajtai, Kumar, and Sivakumar in [5]. The randomized sieving approach consists of sampling an exponential number of "perturbed" lattice points, and then iteratively clustering and combining them to give shorter and shorter lattice points. The run-time and space requirement were proven to be $2^{O(n)}$ where n is the lattice dimension. Nguyen and Vidick did an analysis of this algorithm in [72] and showed that the run-time is $2^{O(5.9n+o(n))}$ and the space required is $2^{O(2.95n+o(n))}$ where n is the lattice dimension. The authors also presented a heuristic variant of the algorithm without perturbations whose running time is $(4/3+\epsilon)^n$ polynomial-time operations, and whose space requirement is $(4/3+\epsilon)^{n/2}$ polynomially many bits but as they mention this algorithm becomes problematic for $n > 50$ in terms of space requirement.

In 2010, Micciancio and Voulgaris [63] presented a provable sieving variant called ListSieve and a more practical, heuristic variant called GaussSieve. ListSieve has $2^{O(3.199n+o(n))}$ run-time and $2^{O(1.325n+o(n))}$ space requirement where n is the lattice dimension. For GaussSieve for run-time no upper bound is currently known and it requires $2^{O(0.41n)}$ space. Pujol and Stehlé in [76] using the birthday paradox improved the bounds of ListSieve to $2^{O(2.465n+o(n))}$ for run-time and $2^{O(1.233n+o(n))}$ for space complexity. Finally, the work of Blomër and Naewe in [11] deals with all ℓ_p norms, generalizing the Ajtai-Kumar-Sivakumar sieve.

Using heuristics like extreme pruning in [24], enumeration algorithms outperform sieving algorithms again, as it can be seen in the SVP challenge at `http://www.latticechallenge.org/svp-challenge/`.

The Voronoi cell based algorithms were introduced in a breakthrough work [62] by Micciancio and Voulgaris. The Voronoi cell $\mathcal{V}(\mathcal{L})$ of a lattice $\mathcal{L}$ is the set of vectors closer to the origin than to any other lattice point:

$$\mathcal{V}(\mathcal{L}) = \{\mathbf{x} : \forall \mathbf{c} \in \mathcal{L}, \|\mathbf{x}\| < \|\mathbf{c} - \mathbf{x}\|\}.$$

Stated differently the Voronoi cell is the interior of a polytope.

Although the previous definition of the Voronoi cell involves an infinite number of inequalities, for a lattice $\mathcal{L} \subset \mathbb{R}^m$ there exists a minimal set of vectors $\{\mathbf{v}_j\}_{j \leq m} \in \mathcal{L}$ that suffices to define the Voronoi cell:

$$\mathcal{V}(\mathcal{L}) = \{\mathbf{x} : \forall j \leq m, \|\mathbf{x}\| < \|\mathbf{v}_j - \mathbf{x}\|\}.$$

We call there vectors the relevant vectors of $\mathcal{L}$. Assume that we know the relevant vectors, we can use them to solve SVP (and CVP). It is the first deterministic single exponential algorithm for exact SVP under the ℓ_2 norm in $2^{O(n)}$ time and space where n is the dimension of the lattice.

Recall that given a basis for a lattice and a parameter d, $GapSVP_\gamma$ (Definition 30, p. 32) is the promise problem of answering whether or not the given lattice has a shortest vector less that d or doesn't have any vector shorter than $\gamma \cdot d$. Summing up, the best known algorithms for $GapSVP_\gamma$ ([79, 62]), require at least $2^{\tilde{\Omega}(n/log\gamma)}$ time.

Finally, we must mention that any algorithm that solves SVP_γ can be used to solve $GapSVP_\gamma$ as well, but the converse is an open problem.

Next, we show that there is a polynomial time reduction from Subset sum to SVP with respect to the ℓ_∞ norm.

Proposition 3.1 $Subset\ sum \leq_{pol} SVP_{\ell_\infty}$

Proof. Let $\mathbf{B} \in \mathbb{Z}^{(n+2)\times(n+1)}$ defined as

$$\mathbf{B} = \begin{pmatrix} | & & | \\ \mathbf{b}_1 & \dots & \mathbf{b}_{n+1} \\ | & & | \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 2 & 0 & \cdots & 0 & 1 \\ \vdots & & \ddots & & & \vdots \\ 0 & 0 & 0 & \cdots & 2 & 1 \\ 2a_1 & 2a_2 & 2a_3 & \cdots & 2a_n & 2b \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Clearly, $\mathbf{B}$ can be constructed in polynomial time. It is easy to see that $rank(\mathbf{B}) = n+1$ therefore the columns of $\mathbf{B}$, namely $\mathbf{b}_1, \dots, \mathbf{b}_{n+1}$, are linearly independent vectors and can form a lattice basis.

To show that $rank(\mathbf{B}) = n+1$ we use elementary row operations to zero out the $(n+1)-th$ line by multiplying each of the first n lines with $-a_i$ for $i = 1, \dots, n$ and adding them to the $(n+1)-th$ line, multiply the $(n+2)-th$ line with $-b$ and add it to the $(n+1)-th$ line and then exchange lines $n+1$ and $n+2$. In this way we reduce matrix $\mathbf{B}$ to row-echelon form with a zero row and due to the form of $\mathbf{B}$ we cannot zero out any more rows. Finally, because $rank(\mathbf{B}^\intercal) = rank(\mathbf{B})$ we have that $rank(\mathbf{B}) = n+1$.

We will now show that,

$$(a_1, \dots, a_n, b) \in Subset\ Sum \Leftrightarrow \|\lambda_1(\mathcal{L}(\mathbf{B}))\|_\infty = 1$$

(a) ("$\Rightarrow$"): Let $(x_1, \dots, x_n) \in \{0,1\}^n$ such that $\sum_{i=1}^n x_i a_i = b$.

Then for $\mathbf{x} = (x_1, \ldots, x_n, -1)^\mathsf{T}$ we have that

$$\mathbf{Bx} = \sum_{i=1}^{n} x_i \mathbf{b}_i - \mathbf{b}_{n+1} = \begin{pmatrix} 2x_1 - 1 \\ \vdots \\ 2x_n - 1 \\ (\sum_{i=1}^{n} 2x_i a_i) - 2b \\ -1 \end{pmatrix} \Rightarrow \|\mathbf{Bx}\|_\infty = 1$$

because:

- $|2x_i - 1| = 1$ for all $x_i \in \{0, 1\}$
- $(\sum_{i=1}^{n} 2x_i a_i) - 2b = 2\left(\sum_{i=1}^{n} x_i a_i - b\right) = 0$
- $|-1| = 1$

(b) ("$\Leftarrow$"): Suppose that $\|\lambda_1(\mathcal{L}(\mathbf{B}))\|_\infty = 1$, i.e.,

$$\left\| \sum_{i=1}^{n+1} x_i \mathbf{b}_i \right\|_\infty = 1 \quad \text{where } x_i \in \mathbb{Z}$$

Then we have that $|2x_i + x_{n+1}| \leq 1$ for $i = 1, \ldots, n$. From the last line of $\mathbf{B}$ we conclude that $|x_{n+1}| = 1 \Rightarrow x_{n+1} = \pm 1$. Without loss of generality assume that $x_{n+1} = -1 \Rightarrow |2x_i - 1| \leq 1$, otherwise we can multiply with -1 the $\lambda_1(\mathcal{L}(\mathbf{B}))$ and have again $|2x_i - 1| \leq 1$ because $\|-\lambda_1(\mathcal{L}(\mathbf{B}))\|_\infty = \|\lambda_1(\mathcal{L}(\mathbf{B}))\|_\infty$.

Because $x_i \in \mathbb{Z}$ then either $|2x_i - 1| = 1$ or $|2x_i - 1| = 0$. For $|2x_i - 1| = 0$ we get that $x_i = \frac{1}{2}$ a contradiction to the fact that $x_i \in \mathbb{Z}$, hence

$$|2x_i - 1| = 1 \Rightarrow 2x_i - 1 = \pm 1 \Rightarrow x_i = 0 \text{ or } x_i = 1 \Rightarrow x_i \in \{0, 1\} \text{ for } i = 1, \ldots, n$$

From the $n - th$ line of matrix $\mathbf{B}$ we have that $|\sum_{i=1}^{n} 2x_i a_i - 2b| \leq 1 \Rightarrow |\sum_{i=1}^{n} x_i a_i - b| \leq \frac{1}{2} \Rightarrow |\sum_{i=1}^{n} x_i a_i - b| = 0$ because $\mathcal{L}(\mathbf{B})$ is an integral lattice. Therefore, we have that

$$\sum_{i=1}^{n} x_i a_i - b = 0 \Rightarrow \sum_{i=1}^{n} x_i a_i = b$$

and this completes the proof.

$\square$

3.2 Closest vector problem

The *Closest Vector Problem* has been investigated for more than a century but it has attracted less attention than SVP which is its homogeneous counterpart. Today much is known about the computational complexity of CVP in both its exact and approximation version. For some of the algorithms below or their extensions/improvements we also mention their usage for solving/approximating SVP.

CVP is NP-hard to approximate to within $n^{c/log\,log\,n}$ factors for some $c > 0$ [8, 21, 20], where n is the dimension of the lattice. Therefore, as with SVP, we do not expect to solve (or even closely approximate) CVP efficiently in high dimensions.

As with SVP, there are three approaches to solve CVP: enumeration algorithms, probabilistic sieving algorithms and Voronoi cell based algorithms.

Before we continue we must mention that the lattice basis reduction algorithms such as the LLL basis reduction algorithm [51] and some of its first extensions [9, 79] give $2^{poly(logn)}$ approximations to SVP and CVP in the ℓ_2 norm in $poly(n)$ time.

Enumeration algorithms such as Kannan's algorithm [46] and further improvements [38, 35] can be used to solve exact SVP and CVP in the ℓ_2 norm in $2^{O(n\,logn)}$ time and $poly(n)$ space. As with SVP, also for CVP enumeration algorithms remain the most practical solver for these two problems and much effort has been spend on optimizing them as we saw on the previous section (see [24]).

The randomized sieving algorithm of Ajtai, Kumar and Sivakumar [5] was furthered used to create a $1/\epsilon^n$ time and space algorithm for the $(1+\epsilon)$-CVP unde the ℓ_2 norm [6, 11], ℓ_p norms [11], near symmetric norms [17], and in [22] Eisenbrand, Hähnle and Niemeier show that we can solve $(1+\epsilon)$-CVP under the ℓ_∞ norm using $O(ln\frac{1}{\epsilon}^n)$ calls to any 2-approximate solver. The Ajtai, Kumar and Sivakumar sieve based algorithms are the only algorithms currently available for solving $(1+\epsilon)$-CVP under non-euclidean norms.

The work of Micciancio and Voulgaris in [62] gave a deterministic $2^{O(n)}$ time and space algorithm for exact CVP under the ℓ_2 norm where n is the dimension of the lattice.

Finally, we must mention that the search and decisional versions of the exact Closest Vector Problem are polynomially equivalent and that any algorithm that solves CVP_γ can be used to solve $GapCVP_\gamma$ as well (see [61, ch. 3]).

Next, we show that there is a polynomial time reduction from Subset sum to CVP with respect to the ℓ_∞ norm.

Proposition 3.2 $Subset\ sum \leq_{pol} CVP_{\ell_\infty}$

Proof.

Let $\mathbf{B}' \in \mathbb{Z}^{(n+1)\times(n+1)}$ defined as

$$\mathbf{B}' = \begin{pmatrix} | & | \\ \mathbf{B} & \mathbf{y} \\ | & | \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 2 & 0 & \cdots & 0 & 1 \\ \vdots & & \ddots & & & \vdots \\ 0 & 0 & 0 & \cdots & 2 & 1 \\ 2a_1 & 2a_2 & 2a_3 & \cdots & 2a_n & 2b \end{pmatrix}$$

where $\mathbf{B} \in \mathbb{Z}^{(n+1)\times n}$ and $\mathbf{y} \in \mathbb{Z}^{(n+1)\times 1}$, i.e., $\mathbf{B}$ is the matrix that consists of the first n columns of $\mathbf{B}'$ and $\mathbf{y}$ is the last column of $\mathbf{B}'$, namely

$$\mathbf{B} = \begin{pmatrix} 2 & 0 & 0 & \cdots & 0 \\ 0 & 2 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 2 \\ 2a_1 & 2a_2 & 2a_3 & \cdots & 2a_n \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 2b \end{pmatrix}$$

Clearly, $\mathbf{B}'$ can be constructed in polynomial time. It is easy to see that $rank(\mathbf{B}) = n$, therefore the columns of $\mathbf{B}$ are linearly independent vectors and can form a lattice basis.

To show that $rank(\mathbf{B}) = n$ we use elementary row operations to zero out the $(n+1)-th$ line by multiplying each of the first n lines with $-a_i$ for $i = 1, \ldots, n$ and adding them to the $(n+1)-th$ line. In this way we reduce matrix $\mathbf{B}$ to row-echelon form with a zero row and due to the form of $\mathbf{B}$ we cannot zero out any more rows. Finally, because $rank(\mathbf{B}^\intercal) = rank(\mathbf{B})$ we have that $rank(\mathbf{B}) = n$.

We will now show that,

$$(a_1, \ldots, a_n, b) \in Subset\ Sum \Leftrightarrow \exists \mathbf{x} \in \mathbb{Z}^n \text{ such that } \|\mathbf{Bx} - \mathbf{y}\|_\infty = 1$$

(a) ("$\Rightarrow$"): Let $(x_1, \ldots, x_n) \in \{0,1\}^n$ such that $\sum_{i=1}^n x_i a_i = b$.
Then for $\mathbf{x} = (x_1, \ldots, x_n)^\intercal$ we have that

$$\mathbf{Bx} - \mathbf{y} = \begin{pmatrix} 2x_1 - 1 \\ \vdots \\ 2x_n - 1 \\ (\sum_{i=1}^n 2x_i a_i) - 2b \end{pmatrix} \Rightarrow \|\mathbf{Bx} - \mathbf{y}\|_\infty = 1$$

because:

- $|2x_i - 1| = 1$ for all $x_i \in \{0,1\}$

- $\left(\sum_{i=1}^{n} 2x_i a_i\right) - 2b = 2\left(\sum_{i=1}^{n} x_i a_i - b\right) = 0$

(b) ("$\Leftarrow$"): Suppose that $\|\mathbf{Bx} - \mathbf{y}\|_\infty = 1$ where $\mathbf{x} \in \mathbb{Z}^n$, i.e.,

$$\|\mathbf{Bx} - \mathbf{y}\|_\infty = \left\| \begin{array}{c} 2x_1 - 1 \\ \vdots \\ 2x_n - 1 \\ \left(\sum_{i=1}^{n} 2x_i a_i\right) - 2b \end{array} \right\|_\infty = 1$$

Because $x_i \in \mathbb{Z}$ then either $|2x_i - 1| = 1$ or $|2x_i - 1| = 0$. For $|2x_i - 1| = 0$ we get that $x_i = \frac{1}{2}$ a contradiction to the fact that $x_i \in \mathbb{Z}$, hence

$|2x_i - 1| = 1 \Rightarrow 2x_i - 1 = \pm 1 \Rightarrow x_i = 0$ or $x_i = 1 \Rightarrow x_i \in \{0, 1\}$ for $i = 1, \ldots, n$

From the $(n+1) - th$ line of $\mathbf{Bx} - \mathbf{y}$ we have that $|\sum_{i=1}^{n} 2x_i a_i - 2b| \leq 1 \Rightarrow |\sum_{i=1}^{n} x_i a_i - b| \leq \frac{1}{2} \Rightarrow |\sum_{i=1}^{n} x_i a_i - b| = 0$ because $\mathcal{L}(\mathbf{B})$ is an integral lattice. Therefore, we have that

$$\sum_{i=1}^{n} x_i a_i - b = 0 \Rightarrow \sum_{i=1}^{n} x_i a_i = b$$

and this completes the proof.

$\square$

Finally, we show that there is a polynomial time reduction from Subset sum to CVP with respect to the ℓ_2 norm.

Proposition 3.3 $Subset\ sum \leq_{pol} CVP_{\ell_2}$

Proof. As in the previous proof let $\mathbf{B}' \in \mathbb{Z}^{(n+1)\times(n+1)}$ defined as

$$\mathbf{B}' = \begin{pmatrix} | & | \\ \mathbf{B} & \mathbf{y} \\ | & | \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 2 & 0 & \cdots & 0 & 1 \\ \vdots & & \ddots & & & \vdots \\ 0 & 0 & 0 & \cdots & 2 & 1 \\ 2a_1 & 2a_2 & 2a_3 & \cdots & 2a_n & 2b \end{pmatrix}$$

where $\mathbf{B} \in \mathbb{Z}^{(n+1)\times n}$ and $\mathbf{y} \in \mathbb{Z}^{(n+1)\times 1}$, i.e., $\mathbf{B}$ is the matrix that consists of the first n columns of $\mathbf{B}'$ and $\mathbf{y}$ is the last column of $\mathbf{B}'$, namely

$$\mathbf{B} = \begin{pmatrix} | & & | \\ \mathbf{b}_1 & \dots & \mathbf{b}_n \\ | & & | \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & \cdots & 0 \\ 0 & 2 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 2 \\ 2a_1 & 2a_2 & 2a_3 & \cdots & 2a_n \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 2b \end{pmatrix}$$

Clearly, $\mathbf{B}'$ can be constructed in polynomial time. It is easy to see that $rank(\mathbf{B}) = n$, therefore the columns of $\mathbf{B}$, namely $\mathbf{b}_1, \dots, \mathbf{b}_n$, are linearly independent vectors and can form a lattice basis.

We will now show that,

$$(a_1, \dots, a_n, b) \in Subset\ Sum \Leftrightarrow \exists \mathbf{x} \in \mathbb{Z}^n \text{ such that } \|\mathbf{Bx} - \mathbf{y}\| = \sqrt{n}$$

(a) ("$\Rightarrow$"): Let $(x_1, \dots, x_n) \in \{0, 1\}^n$ such that $\sum_{i=1}^n x_i a_i = b$.
Then for $\mathbf{x} = (x_1, \dots, x_n)^\intercal$ we have that

$$\mathbf{Bx} - \mathbf{y} = \begin{pmatrix} 2x_1 - 1 \\ \vdots \\ 2x_n - 1 \\ (\sum_{i=1}^n 2x_i a_i) - 2b \end{pmatrix} = \begin{pmatrix} 2x_1 - 1 \\ \vdots \\ 2x_n - 1 \\ 0 \end{pmatrix}$$

therefore,

$$\|\mathbf{Bx} - \mathbf{y}\| = \|(\pm 1, \dots, \pm 1, 0)\| = \sqrt{n}$$

(b) ("$\Leftarrow$"): Suppose that $\|\mathbf{Bx} - \mathbf{y}\| = \sqrt{n}$ where $\mathbf{x} \in \mathbb{Z}^n$, i.e.,

$$\|\mathbf{Bx} - \mathbf{y}\| = \left\| \begin{matrix} 2x_1 - 1 \\ \vdots \\ 2x_n - 1 \\ (\sum_{i=1}^n 2x_i a_i) - 2b \end{matrix} \right\| = \sqrt{n} \Rightarrow$$

$$\|\mathbf{Bx} - \mathbf{y}\| = \sqrt{\sum_{i=1}^n (2x_i - 1)^2 + \left(\left(\sum_{i=1}^n 2x_i a_i \right) - 2b \right)^2} = \sqrt{n} \Rightarrow$$

$$\sum_{i=1}^n (2x_i - 1)^2 + \left(\left(\sum_{i=1}^n 2x_i a_i \right) - 2b \right)^2 = n$$

Therefore we have that,

$$\underbrace{\left(\left(\sum_{i=1}^{n} 2x_i a_i\right) - 2b\right)^2}_{\text{is } \geq 0} = \underbrace{n - \sum_{i=1}^{n}(2x_i - 1)^2}_{\text{must be } \geq 0}$$

Because $x_i \in \mathbb{Z}$ we have that $\sum_{i=1}^{n}(2x_i - 1)^2 \in \mathbb{Z}$. Since we are subtracting from n a sum of nonnegative values and we want the result to be also nonnegative, and from the fact that $x_i \in \mathbb{Z}$ it follows that $(2x_i - 1)^2 = 1$ for $i = 1, \ldots, n$. Therefore we have that $|2x_i - 1| = 1$ from which we get that $x_i \in \{0, 1\}$ for $i = 1, \ldots, n$ again because $x_i \in \mathbb{Z}$.

Hence, for $x_i \in \{0, 1\}$ we get that,

$$n - \sum_{i=1}^{n}(2x_i - 1)^2 = n - \sum_{i=1}^{n}|\pm 1| = 0$$

therefore we have that,

$$\begin{aligned}
\left(\left(\sum_{i=1}^{n} 2x_i a_i\right) - 2b\right)^2 &= 0 && \Rightarrow \\
\left(\sum_{i=1}^{n} 2x_i a_i\right) - 2b &= 0 && \Rightarrow \\
\sum_{i=1}^{n} 2x_i a_i &= 2b && \Rightarrow \\
\sum_{i=1}^{n} x_i a_i &= b
\end{aligned}$$

and this completes the proof.

□

The NP-completeness reduction for CVP can be generalized for any ℓ_p norm ($p \geq 1$), see [84]. We note that the reductions we presented for both SVP and CVP are for the decision version of these problems.

There is a special case of $GapCVP_\gamma$ which is of particular interest in cryptography. If in the input of $GapCVP_\gamma$ we have that $d < \lambda_1(\mathcal{L})/(2 \cdot \gamma(n))$ then the problem is called $GapBDD_\gamma$ where BDD stands for Bounded Distance Decoding.

The search approximation version of this problem is defined as follows,

Definition 48 (γ-Bounded Distance Decoding (BDD_γ)) *Given a lattice basis* $\mathbf{B}$ *and vector* $\mathbf{t}$ *such that* $dist(\mathbf{t}, \mathcal{L}(\mathbf{B})) \leq \gamma(n) \cdot \lambda_1(\mathcal{L}(\mathbf{B}))$ *the task is to find the lattice vector* $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ *closest to* $\mathbf{t}$*.*

3.3 Reducing approximate SVP to approximate CVP

In this section we follow the work of Goldreich, Micciancio, Safra and Seifert in [30] to show that there is a Cook reduction from approximate SVP to approximate CVP for any ℓ_p norm ($p \geq 1, p = \infty$).

One could think that we could set the target vector in CVP to be the zero vector and use this as an oracle to solve SVP for a lattice $\mathcal{L} \subset \mathbb{R}^n$. This would not work because in SVP we are searching for a nonzero lattice vector whereas in CVP the target vector can be the solution if it is a lattice vector itself and the zero vector is always a lattice vector. To avoid this situation we run an CVP oracle on a sublattice $\mathcal{L}' \subset \mathcal{L}$ not containing the target vector and thus the problem now is how to select a sublattice without removing all the lattice vectors closest to the target vector. Details follow.

Proposition 3.4 *Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a basis for the lattice $\mathcal{L}$ and let $\mathbf{v} = \sum_{i=1}^n c_i \mathbf{b}_i$ where $c_i \in \mathbb{Z}$, be a shortest nonzero vector in $\mathcal{L}$. Then, there exists an i such that c_i is odd.*

Proof. Assume that all i are even. Then,

$$\frac{1}{2}\mathbf{v} = \sum_{i=1}^{n} \frac{c_i}{2}\mathbf{b}_i$$

is a shorter vector in $\mathcal{L}$ contradicting the fact that $\mathbf{v}$ is a shortest vector. □

Next we show how to reduce SVP to solving n instances of CVP.

Given a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ of the lattice $\mathcal{L}$ we construct the $j-th$ instance of CVP for $j = 1, \ldots, n$ with lattice basis

$$\mathbf{B}^{(j)} = \begin{pmatrix} | & & | & | & | & & | \\ \mathbf{b}_1 & \ldots & \mathbf{b}_{j-1} & 2\mathbf{b}_j & \mathbf{b}_{j+1} & \ldots & \mathbf{b}_n \\ | & & | & | & | & & | \end{pmatrix}$$

and target vector $\mathbf{b}_j$.

Proposition 3.5 *For $j = 1, \ldots, n$ we have $\mathcal{L}\left(\mathbf{B}^{(j)}\right) \subset \mathcal{L}(\mathbf{B})$.*

Proof. For every $\mathbf{v} \in \mathcal{L}\left(\mathbf{B}^{(j)}\right)$ we have that

$$\mathbf{v} = \sum_{i \neq j} c_j \mathbf{b}_i + (2c_j)\mathbf{b}_j = \sum_{i=1}^{n} a_i \mathbf{b}_i \Rightarrow \mathbf{v} \in \mathcal{L}(\mathbf{B})$$

where $a_i = c_i$ for $i \neq j$ and $a_i = 2c_i$ for $i = j$.

On the other hand we have that $\mathbf{b}_j \in \mathcal{L}(\mathbf{B})$. Assume, for contradiction, that $\mathbf{b}_j \in \mathcal{L}\left(\mathbf{B}^{(j)}\right)$. Then we would have for $a_i \in \mathbb{Z}$,

$$\begin{aligned} a_1\mathbf{b}_1 + \ldots + 2a_j\mathbf{b}_j + \ldots + a_n\mathbf{b}_n = \mathbf{b}_j & \qquad \Rightarrow \\ a_1\mathbf{b}_1 + \ldots + (2a_j - 1)\mathbf{b}_j + \ldots + a_n\mathbf{b}_n = 0 & \end{aligned}$$

contradicting the linear independence of $\mathbf{b}_1, \ldots, \mathbf{b}_n$ since $a_j \in \mathbb{Z}$. Therefore $\mathbf{b}_j \notin \mathcal{L}\left(\mathbf{B}^{(j)}\right)$ and this completes the proof. $\square$

Proposition 3.6 *Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a basis for the lattice $\mathcal{L}$ and let $\mathbf{v} = \sum_{i=1}^{n} c_i\mathbf{b}_i$ where $c_i \in \mathbb{Z}$, be a lattice vector in $\mathcal{L}$. Then*

$$\mathbf{u} = \frac{c_j + 1}{2} 2\mathbf{b}_j + \sum_{i \neq j} c_i\mathbf{b}_i$$

is a lattice vector in $\mathcal{L}\left(\mathbf{B}^{(j)}\right)$ and $dist(\mathbf{u}, \mathbf{b}_j) = \|\mathbf{v}\|$ where $\mathbf{b}_j$ is the target vector.

Proof. Since c_j is odd then $\frac{c_j+1}{2}$ is an integer and thus $\mathbf{u}$ is a lattice vector in $\mathcal{L}\left(\mathbf{B}^{(j)}\right)$. So we have that,

$$\mathbf{u} - \mathbf{b}_j = \frac{c_j + 1}{2} 2\mathbf{b}_j + \sum_{i \neq j} c_i\mathbf{b}_i - \mathbf{b}_j = c_j\mathbf{b}_j + \sum_{i \neq j} c_i\mathbf{b}_i = \mathbf{v}$$

and the proposition follows. $\square$

Proposition 3.7 *Let $\mathbf{u} = 2c'_j\mathbf{b}_j + \sum_{i \neq j} c_i\mathbf{b}_i$ be a lattice vector in $\mathcal{L}\left(\mathbf{B}^{(j)}\right)$. Then $\mathbf{v} = (2c'_j - 1)\mathbf{b}_j + \sum_{i \neq j} c_i\mathbf{b}_i$ is a nonzero lattice vector in $\mathcal{L}(\mathbf{B})$ and $\|\mathbf{v}\| = dist(\mathbf{u}, \mathbf{b}_j)$.*

Proof. Since $c'_j \in \mathbb{Z}$ then $2c'_j - 1$ cannot be zero and in fact is an odd integer, and thus $\mathbf{v}$ is a nonzero vector. Then we have that,

$$\mathbf{v} = (2c'_j - 1)\mathbf{b}_j + \sum_{i \neq j} c_i\mathbf{b}_i = 2c'_j\mathbf{b}_j + \sum_{i \neq j} c_i\mathbf{b}_i - \mathbf{b}_j = \mathbf{u} - \mathbf{b}_j$$

and the proposition follows. $\square$

Theorem 3.8 *For every function $\gamma(n) \geq 1$ with $n \in \mathbb{N}$, SVP_γ (resp. $GapSVP_\gamma$) is Cook-reducible to CVP_γ (resp. $GapCVP_\gamma$). Furthermore, the reduction is non-adaptive, and all queries maintain the rank of the input instance.*

Proof. We present a proof for both the decision and the search version.

Decision: Let $(\mathbf{B}, d)$ be a $GapSVP_\gamma$ instance, and define $GapCVP_\gamma$ instances $\left(\mathbf{B}^{(j)}, \mathbf{b}_j, d\right)$ for $j = 1, \ldots, n$. We want to prove that if $(\mathbf{B}, d)$ is a YES instance, then $\left(\mathbf{B}^{(j)}, \mathbf{b}_j, d\right)$ is a YES instance for some $j = 1, \ldots, n$ and if $(\mathbf{B}, d)$ is a NO instance, then $\left(\mathbf{B}^{(j)}, \mathbf{b}_j, d\right)$ is a NO instance for every $j = 1, \ldots, n$

First assume $(\mathbf{B}, d)$ is a YES instance and let $\mathbf{v} = \sum_{i=1}^{n}$ be a shortest nonzero lattice vector in $\mathcal{L}(\mathbf{B})$. So we have that $\|\mathbf{v}\| \leq d$ and by Proposition 3.4 c_j is odd for some j. The vector $\mathbf{u}$ as defined in Proposition 3.6 is in $\mathcal{L}\left(\mathbf{B}^{(j)}\right)$ and satisfies $dist(\mathbf{u}, \mathbf{b}_j) = \|\mathbf{v}\| \leq d$, proving that $\left(\mathbf{B}^{(j)}, \mathbf{b}_j, d\right)$ is a YES instance.

For the NO instances we prove the contrapositive. Assume $\left(\mathbf{B}^{(j)}, \mathbf{b}_j, d\right)$ is not a NO instance for some j. Then there exists a vector $\mathbf{u}$ in $\mathcal{L}\left(\mathbf{B}^{(j)}\right)$ such that $dist(\mathbf{u}, \mathbf{b}_j) \leq \gamma(n) \cdot d$. The vector $\mathbf{v}$ as defined in Proposition 3.7 is a nonzero lattice vector in $\mathcal{L}(\mathbf{B})$ and satisfies $\|\mathbf{v}\| = dist(\mathbf{u}, \mathbf{b}_j) \leq \gamma(n) \cdot d$, proving that $(\mathbf{B}, d)$ is not a NO instance.

Search: In the search version we make n queries to the CVP_γ oracle with input $\left(\mathbf{B}^{(j)}, \mathbf{b}_j\right)$ for $j = 1, \ldots, n$. Let $\mathbf{u}_j$ be the oracle answer for the $j - th$ query. By Proposition 3.6 $\mathbf{v}_j = \mathbf{u}_j - \mathbf{b}_j$ is in $\mathcal{L}(\mathbf{B})$, so it remains to show that one of them is a shortest vector.

Now suppose that $\mathbf{v}$ is the shortest vector in $\mathcal{L}(\mathbf{B})$. Then we have that $\mathbf{v} = \sum_{i=1}^{n} c_j \mathbf{b}_i$ and by Proposition 3.4 there exists a j such that c_j is an odd integer. By Proposition 3.6 we have that $\mathbf{u}_j = \mathbf{v} + \mathbf{b}_j$ is the closest vector to $\mathbf{b}_j$ in $\mathcal{L}\left(\mathbf{B}^{(j)}\right)$ and $\mathbf{u}_j$ is the shortest among all $\mathbf{u}_i$ for $i = 1, \ldots, n$ exactly because $\mathbf{v}$ is a shortest vector of $\mathcal{L}(\mathbf{B})$. So the oracle query $CVP_\gamma\left(\mathbf{B}^{(j)}, \mathbf{b}_j\right)$ will respond with the vector $\mathbf{u}_j$ and thus we can get $\mathbf{u}_j = \mathbf{v} + \mathbf{b}_j \Rightarrow \mathbf{v} = \mathbf{u}_j - \mathbf{b}_j$. To summarize we have that, $\lambda_1(\mathcal{L}(\mathbf{B})) = \underset{1 \leq i \leq n}{min}\, dist(\mathbf{u}_i, \mathbf{b}_i)$ where $\mathbf{u}_i$ is the answer to the $i - th$ CVP_γ oracle query with input the pair $\left(\mathbf{B}^{(j)}, \mathbf{b}_j\right)$. □

The advantages of the previous reduction are that it is *gap* and *rank* preserving. One drawback of the reduction is that it is a Cook reduction, i.e., more than one oracle query needs to be made. Furthermore, with similar ideas also in [30] there is a *randomized* Karp reduction from SVP_γ (resp. $GapSVP_\gamma$) to CVP_γ (resp. $GapCVP_\gamma$) that maps YES instances to YES instances with probability at least $1/2$, and NO instances are always mapped to NO instances. This randomized reduction is also *gap* and *rank* preserving. It is an open problem whether there exists a *deterministic* Karp reduction for that matter.

Proposition 3.9 *Let $\mathcal{L}(\mathbf{B})$ be a lattice of dimension n. Then for any $\gamma(n) \geq 1$, $GapCVP_\gamma$ is in NP, therefore $GapSVP_\gamma$ is also in NP for the same $\gamma(n)$.*

Proof. Let $(\mathbf{B}, \mathbf{t}, d)$ be a $GapCVP_\gamma$ instance. A witness is a vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v} - \mathbf{t}\| \leq d$. Since vector $\mathbf{v}$ is of polynomial size because its length is at most $\|\mathbf{t}\| + d$ and can be verified in polynomial time by checking that $\|\mathbf{v} - \mathbf{t}\| \leq d$, so the proposition follows. $\square$

Proposition 3.10 *Let $\mathcal{L}(\mathbf{B})$ be a lattice of dimension n. Then for any $\gamma(n) \geq 1$, $GapSVP_\gamma$ in NP.*

Proof. Follows from Theorem 3.8 and Proposition 3.9. $\square$

3.4 Limits to inapproximability

As we saw in sections 3.1 and 3.2 even for constant approximation factors, no efficient algorithm is known for SVP (resp. GapSVP) or CVP (resp. GapCVP). GapSIVP is NP-hard to approximate to within any constant factor, and no polynomial time algorithm exists for any $2^{log^{1-\epsilon} n}$ factor unless $NP \subseteq DTIME\left(2^{poly(logn)}\right)$.

Haviv and Regev [36] showed that for GapCRP, for all sufficiently large $p \leq \infty$, there is a constant $c_p > 1$ such that GapCRP in the ℓ_p norm is Π_2-hard to approximate to within any factor less that c_p and in particular for $p = \infty$ it is $c_\infty = 3/2$ which gets closer to the factor 2 beyond which the problem is not believed to be Π_2-hard (see [31]). It is an open question where $GapCRP_\gamma$ is Π_2-hard with respect to the ℓ_p norm for small values of $p \geq 1$. The covering radius problem can be approximated within any constant factor $\gamma(n) > 1$ in random exponential time $2^{O(n)}$ (see [31]).

Khot, Popat and Vishnoi [49] showed for an arbitrarily small constant $\epsilon > 0$, assuming $NP \nsubseteq DTIME\left(2^{log^{O(1/\epsilon)} n}\right)$, CVPP is hard to approximate within a factor better than $2^{log^{1-\epsilon} n}$ improving the previous hardness factor of $log^{\delta} n$ for some $\delta > 0$ due to Alekhnovich, Khot, Kindler and Vishnoi [7].

One might hope to increase the factors in the hardness results above, however there seem to be strict limits to any such improvements. We note that AM is the complexity class of languages that have a constant round interactive proof system. A well-known complexity theoretic result is that if $NP \subseteq coAM$, then the polynomial hierarchy collapses (see Boppana, Håstad and Zachos [12]).

In general, proving that for some approximation factor $\gamma(n)$ a certain problem is in a class not believed to be in NP such as coNP or coAM, implies that for that approximation factor the problem is not NP-hard, assuming that the polynomial hierarchy does not collapse. From Propositions 3.9 and 3.10 we have that for any $\gamma(n) \geq 1$, $GapCVP_\gamma$ and $GapSVP_\gamma$ are in NP.

Lagarias, Lenstra and Schnorr in [50] showed that for $\gamma(n) = n^{3/2}$, $GapSVP_\gamma$ and $GapCVP_\gamma$ are in coNP. Banaszczyk [10] improved this to $\gamma(n) = n$. Goldreich and Goldwasser in [26] showed that for some $\gamma(n) = O\left(\sqrt{n/logn}\right)$, $GapSVP_\gamma$ and $GapCVP_\gamma$ are in coAM.

Aharov and Regev in [1] showed that for some $\gamma(n) = O\left(\sqrt{n}\right)$, $GapSVP_\gamma$ and $GapCVP_\gamma$ are in NP $\cap$ coNP but their result for gaps between $\sqrt{n/logn}$ and $\sqrt{n}$ does not apply, and so containment in NP $\cap$ coNP is not known to hold.

Therefore for some $\gamma = O\left(\sqrt{n}\right)$ $GapSIVP_\gamma$ and $GapCRP_\gamma$ have been placed in coNP and for $\gamma(n) = 2$ in AM (see [1]). For some $\gamma(n) = O\left(\sqrt{n/logn}\right)$ $GapCRP_\gamma$ has been placed in coAM (see [31]).

$GapCVPP_\gamma$ has been known to be computable in polynomial time (not including the arbitrary preprocessing stage) for $\gamma(n) = O\left(\sqrt{n/logn}\right)$ (see [1]).

The approximation version of the Bounded Distance Decoding problem, namely BDD_γ, has been shown to be NP-hard for $\gamma \geq \frac{1}{\sqrt{2}}$ by Liu, Lyubashevsky and Micciancio in [52] and is an open question whether it is hard for smaller γ. We note that the BDD_γ problem becomes harder as γ becomes larger. In the same paper the authors showed that for $\gamma = O\left(\sqrt{(log\, n)/n}\right)$ BDD_γ with preprocessing can be solved in polynomial time. For a connection of the Bounded Distance Decoding with other lattice problems see Lyubashevsky and Micciancio [53].

4

Lattice-based cryptography

In this chapter we talk about lattice-based cryptographic constructions and lattice-based public-key encryption schemes based on the *Learning With Errors* problem. Before we do that, we formally define what is a public-key encryption scheme.

Definition 49 *A* ***public-key encryption scheme*** *is a tuple of probabilistic polynomial time algorithms* (Gen, Enc, Dec) *such that:*

(1) The key generation algorithm Gen takes as input the security parameter 1^n and outputs a pair of keys (pk, sk), the public key and the private key respectively.

(2) The encryption algorithm Enc takes as input a public key pk and message m from some underlying plaintext space (that may depend on pk),and it outputs a ciphertext c.

(3) The decryption algorithm Dec takes as input a private key sk and a ciphertext c, and outputs a message m or a special symbol $\perp$ denoting failure. We assume without loss of generality that Dec is deterministic.

4.1 Early lattice-based cryptography

Lattice-based cryptography began with the seminal work of Ajtai [2] who showed that random instances of a certain problem are at least as hard to solve as worst-case instances of lattice problems.

The average-case / worst-case connection is of particular interest in cryptography. For example consider a cryptographic scheme in which one can prove that breaking the scheme implies factoring some natural number N. Hence, one must choose a number N that is computationally difficult to factor. But how can we do that? Certainly not by choosing N in a range at random because with probability

$1/2$ the number will be even. Maybe choosing two large primes p, q and setting $N = pq$ will make N hard to factor but one must be careful in how to choose the two primes so as to not make their product easy to factor for some specialized algorithms.

On the other hand, lattice-based schemes, do not have this problem. Showing that if uniformly random instances of a certain problem Π can be solved then certain other hard problems can be solved for all lattices, is a very useful feature for cryptography if we base the security of a cryptographic scheme on the hardness of problem Π. Notice that coming up with a hard instance of problem Π is now easy - just generate a random instance of it. That way one can build cryptographic schemes based on the hardness of random instances of problem Π which in turn are as difficult to solve (and thus break the scheme) as worst-case lattice problems.

Briefly, Ajtai created a family $\mathcal{H}$ of collision-resistant functions $h_{\mathbf{A}}$ indexed by $\mathbf{A} \in \mathbb{Z}_p^{n\times k}$ where $k > nlogp$ and the input to the functions is a vector $\mathbf{x}$ in $\{0,1\}^k$. The output is $h_{\mathbf{A}}(\mathbf{x}) = \mathbf{Ax} \mod p$. Ajtai showed that finding two distinct vectors $\mathbf{x}, \mathbf{x}'$ such that $h_{\mathbf{A}}(\mathbf{x}) = h_{\mathbf{A}}(\mathbf{x}')$ for random $\mathbf{A}$, is as hard as solving certain lattice problems for all lattices (see [2, 27]).

The first cryptosystem that was based on the worst-case hardness of lattice problems was the Ajtai-Dwork cryptosystem [4] (the second one). The security of this system was based on the worst-case hardness of the approximate "unique" Shortest Vector Problem $uSVP_{O(n^8)}$. Recall that in $uSVP_{\gamma(n)}$ the task is to find the shortest vector in a lattice in which the shortest vector is guaranteed to be at least $\gamma(n)$ times smaller than the next shortest (nonparallel) vector. Although the system was not presented using lattices, in the security proof they showed that every instance of uSVP could be transformed into a random instance of their cryptosystem with high probability. However the fact that this cryptosystem is not efficient enough to be practical and secure at the same time was confirmed by Nguyen and Stern [70] in their cryptanalysis of the Ajtai-Dwork cryptosystem. Goldreich, Goldwasser and Halevi [28] proposed a modified version of the Ajtai-Dwork cryptosystem. In their version, they eliminated decryption errors that may appear with small probability (inversely proportional to the security parameter). For both these cryptosystems, CCA1 attacks were presented in [32, 44].

In 1997, Goldreich, Goldwasser and Halevi [29] proposed a public-key cryptosystem (encryption and signatures) inspired by McEliece cryptosystem [54] (which is based on error-correcting codes) and relying on the hardness of CVP. Roughly, their public-key encryption scheme works as follows: The secret key $\mathbf{A}$ is a "good" basis for a random lattice $\mathcal{L}$ and the public key is a "bad" basis $\mathbf{B}$ for the same lattice $\mathcal{L}$. The plaintext message is encoded in vector $\mathbf{s}$ and the ciphertext is $\mathbf{c} = \mathbf{Bs} + \mathbf{e}$ where $\mathbf{e}$ is a small random error vector. That way of creating the ciphertext resembles the McEliece cryptosystem. To decrypt $\mathbf{c}$ first apply Babai's round-off algorithm so that $\mathbf{d} \leftarrow \lfloor \mathbf{A}^{-1}\mathbf{c} \rceil$. Then $\mathbf{d}$ with be $\mathbf{A}^{-1}\mathbf{Bs}$ since the error $\mathbf{e}$ is small, Babai's round-off algorithm will remove it. Finally, compute $\mathbf{B}^{-1}\mathbf{AA}^{-1}\mathbf{Bs}$ to recover the original plaintext $\mathbf{s}$. In 1999, Nguyen [66] showed that the proposed selection of the error vector $\mathbf{e}$ had as a result the leakage of information on the plaintext, and this information leakage allows an attacker to reduce the problem of decrypting ciphertexts to solving particular CVP instances which are much easier that the general problem. Namely, for these instances, the given vector is very close to the lattice, which makes it possible in practice to find the closest vector by standard techniques. Nguyen suggested modifications to fix the encryption process, but estimate that, even modified, the scheme cannot provide security without being impractical, compared to existing schemes. Learning the results of Nguyen's cryptanalysis, one of the authors declared the scheme “dead” [66, p. 3].

In 1998, Hoffstein, Pipher and Silverman [42] proposed a public-key cryptosystem named NTRUEncrypt (original name is NTRU) which was based on the algebraic structures of certain polynomial rings. The hard problem underlying NTRUEncrypt is SVP, although the initial description of NTRUEncrypt does not involve lattices. We use the name NTRUEncrypt to distinguish this cryptosystem from a public-key digital signature cryptosystem named NTRUSign. Since its first release NTRUEncrypt has undergone changes especially in way the parameters are chosen. The latest version is of 2008 and the system is fully accepted to IEEE P1363 standards under the specifications for lattice-based public-key cryptography. In April 2011, NTRUEncrypt was accepted as a X9.98 Standard, for use in the financial services industry. Many attacks have been proposed for NTRUEncrypt,see [16, 86, 33, 43, 55, 23], but so far, the NTRUEncrypt cryptosystem remains strong.

Bibliography

[1] Dorit Aharonov and Oded Regev. Lattice Problems in NP $\cap$ coNP. In *FOCS*, page 362–371. IEEE Computer Society, 2004.

[2] Miklós Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In Gary L. Miller, editor, *STOC*, page 99–108. ACM, 1996.

[3] Miklós Ajtai. The Shortest Vector Problem in L2 is NP-hard for Randomized Reductions (Extended Abstract). In Jeffrey Scott Vitter, editor, *STOC*, page 10–19. ACM, 1998.

[4] Miklós Ajtai and Cynthia Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. In Frank Thomson Leighton and Peter W. Shor, editors, *STOC*, page 284–293. ACM, 1997.

[5] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *STOC*, page 601–610. ACM, 2001.

[6] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. Sampling Short Lattice Vectors and the Closest Lattice Vector Problem. In *IEEE Conference on Computational Complexity*, page 53–57, 2002.

[7] Mikhail Alekhnovich, Subhash Khot, Guy Kindler, and Nisheeth K. Vishnoi. Hardness of Approximating the Closest Vector Problem with Pre-Processing. In *FOCS*, page 216–225. IEEE Computer Society, 2005.

[8] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In *Proceedings of the 1993 IEEE 34th Annual Foundations of Computer Science*, SFCS '93, page 724–733, Washington, DC, USA, 1993. IEEE Computer Society.

[9] László Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[10] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993. 10.1007/BF01445125.

[11] Johannes Blömer and Stefanie Naewe. Sampling Methods for Shortest Vectors, Closest Vectors and Successive Minima. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *ICALP*, volume 4596 of *Lecture Notes in Computer Science*, page 65–77. Springer, 2007.

[12] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP Have Short Interactive Proofs? *Inf. Process. Lett.*, 25(2):127–132, 1987.

[13] M.R. Bremner. *Lattice Basis Reduction: An Introduction to the LLL Algorithm and Its Applications*. Pure and Applied Mathematics. CRC Press, 2011.

[14] A.M. Cohen, H. Cuypers, and H. Sterk. *Some Tapas of Computer Algebra*. Number τ. 13 in Algorithms and Computation in Mathematics. Springer, 1998.

[15] H. Cohen. *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics. Springer, 2000.

[16] Don Coppersmith and Adi Shamir. Lattice Attacks on NTRU. In Walter Fumy, editor, *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, page 52–61. Springer, 1997.

[17] Daniel Dadush. A $O(1/\epsilon^2)$-Time Sieving Algorithm for Approximate Integer Programming. In David Fernández-Baca, editor, *LATIN*, volume 7256 of *Lecture Notes in Computer Science*, page 207–218. Springer, 2012.

[18] Hervé Daudé, Philippe Flajolet, and Brigitte Vallée. An Average-Case Analysis of the Gaussian Algorithm for Lattice Reduction. *Combinatorics, Probability & Computing*, 6(4):397–433, 1997.

[19] Jérémie Detrey, Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Accelerating Lattice Reduction with FPGAs. In Michel Abdalla and Paulo S. L. M. Barreto, editors, *LATINCRYPT*, volume 6212 of *Lecture Notes in Computer Science*, page 124–143. Springer, 2010.

[20] Irit Dinur. Approximating SVP_∞ to within Almost-Polynomial Factors Is NP-Hard. In Gian Carlo Bongiovanni, Giorgio Gambosi, and Rossella Petreschi, editors, *CIAC*, volume 1767 of *Lecture Notes in Computer Science*, page 263–276. Springer, 2000.

[21] Irit Dinur, Guy Kindler, and Shmuel Safra. Approximating-CVP to Within Almost-Polynomial Factors is NP-Hard. In *FOCS*, page 99–111. IEEE Computer Society, 1998.

[22] Friedrich Eisenbrand, Nicolai Hähnle, and Martin Niemeier. Covering cubes and the closest vector problem. In *Proceedings of the 27th annual ACM symposium on Computational geometry*, SoCG '11, page 417–423, New York, NY, USA, 2011. ACM.

[23] Nicolas Gama and Phong Q. Nguyen. New Chosen-Ciphertext Attacks on NTRU. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, page 89–106. Springer, 2007.

[24] Nicolas Gama, Phong Q. Nguyen, and Oded Regev. Lattice Enumeration Using Extreme Pruning. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, page 257–278. Springer, 2010.

[25] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., San Francisco, CA, 1979.

[26] Oded Goldreich and Shafi Goldwasser. On the Limits of Nonapproximability of Lattice Problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000.

[27] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-Free Hashing from Lattice Problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(42), 1996.

[28] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Eliminating Decryption Errors in the Ajtai-Dwork Cryptosystem. In Burton S. Kaliski Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, page 105–111. Springer, 1997.

[29] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-Key Cryptosystems from Lattice Reduction Problems. In Burton S. Kaliski Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, page 112–131. Springer, 1997.

[30] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. Approximating Shortest Lattice Vectors is not Harder than Approximating Closest Lattice Vectors. *Inf. Process. Lett.*, 71(2):55–61, 1999.

[31] Venkatesan Guruswami, Daniele Micciancio, and Oded Regev. The Complexity of the Covering Radius Problem on Lattices and Codes. In *IEEE Conference on Computational Complexity*, page 161–173. IEEE Computer Society, 2004.

[32] Chris Hall, Ian Goldberg, and Bruce Schneier. Reaction Attacks against several Public-Key Cryptosystems. In Vijay Varadharajan and Yi Mu, editors, *ICICS*, volume 1726 of *Lecture Notes in Computer Science*, page 2–12. Springer, 1999.

[33] Daewan Han, Jin Hong, Jae Woo Han, and Daesung Kwon. Key Recovery Attacks on NTRU without Ciphertext Validation Routine. In Reihaneh Safavi-Naini and Jennifer Seberry, editors, *ACISP*, volume 2727 of *Lecture Notes in Computer Science*, page 274–284. Springer, 2003.

[34] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Algorithms for the Shortest and Closest Lattice Vector Problems. In Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing, editors, *IWCC*, volume 6639 of *Lecture Notes in Computer Science*, page 159–190. Springer, 2011.

[35] Guillaume Hanrot and Damien Stehlé. Improved Analysis of Kannan's Shortest Lattice Vector Algorithm. In Alfred Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, page 170–186. Springer, 2007.

[36] Ishay Haviv and Oded Regev. Hardness of the Covering Radius Problem on Lattices. In *IEEE Conference on Computational Complexity*, page 145–158. IEEE Computer Society, 2006.

[37] Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In David S. Johnson and Uriel Feige, editors, *STOC*, page 469–477. ACM, 2007.

[38] Bettina Helfrich. Algorithms to construct Minkowski reduced and Hermite reduced lattice bases. *Theor. Comput. Sci.*, 41(2-3):125–139, December 1985.

[39] Jens Hermans, Michael Schneider 0002, Johannes Buchmann, Frederik Vercauteren, and Bart Preneel. Parallel Shortest Lattice Vector Enumeration on Graphics Cards. In Daniel J. Bernstein and Tanja Lange, editors,

AFRICACRYPT, volume 6055 of *Lecture Notes in Computer Science*, page 52–68. Springer, 2010.

[40] J. Hoffstein, J. Pipher, and J.H. Silverman. *An Introduction to Mathematical Cryptography*. Undergraduate Texts in Mathematics. Springer, New York, NJ, USA, 2008.

[41] Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSIGN: Digital Signatures Using the NTRU Lattice. In Marc Joye, editor, *CT-RSA*, volume 2612 of *Lecture Notes in Computer Science*, page 122–140. Springer, 2003.

[42] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. In Joe Buhler, editor, *ANTS*, volume 1423 of *Lecture Notes in Computer Science*, page 267–288. Springer, 1998.

[43] Nick Howgrave-Graham, Phong Q. Nguyen, David Pointcheval, John Proos, Joseph H. Silverman, Ari Singer, and William Whyte. The Impact of Decryption Failures on the Security of NTRU Encryption. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, page 226–246. Springer, 2003.

[44] Oleg Izmerly and Tal Mor. Chosen ciphertext attacks on lattice-based public key encryption and modern (non-quantum) cryptography in a quantum environment. *Theor. Comput. Sci.*, 367(3):308–323, 2006.

[45] Antoine Joux and Jacques Stern. Lattice Reduction: A Toolbox for the Cryptanalyst. *J. Cryptology*, 11(3):161–185, 1998.

[46] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, August 1987.

[47] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC Press, 2007.

[48] Subhash Khot. Hardness of Approximating the Shortest Vector Problem in Lattices. In *FOCS*, page 126–135. IEEE Computer Society, 2004.

[49] Subhash A. Khot, Preyas Popat, and Nisheeth K. Vishnoi. $2^{log^{1-\epsilon} n}$ hardness for the closest vector problem with preprocessing. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, page 277–288, New York, NY, USA, 2012. ACM.

[50] J. Lagarias, H. Lenstra, and C. Schnorr. Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10:333–348, 1990. 10.1007/BF02128669.

[51] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982. 10.1007/BF01457454.

[52] Yi-Kai Liu, Vadim Lyubashevsky, and Daniele Micciancio. On Bounded Distance Decoding for General Lattices. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *APPROX-RANDOM*, volume 4110 of *Lecture Notes in Computer Science*, page 450–461. Springer, 2006.

[53] Vadim Lyubashevsky and Daniele Micciancio. On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, page 577–594. Springer, 2009.

[54] Robert J. Mceliece. A public-key cryptosystem based on algebraic coding theory. Technical report, Jet Propulsion Lab Deep Space Network Progress report, 1978.

[55] Tommi Meskanen and Ari Renvall. A wrap error attack against NTRUEncrypt. *Discrete Appl. Math.*, 154(2):382–391, February 2006.

[56] Daniele Micciancio. The Shortest Vector in a Lattice is Hard to Approximate to within Some Constant. *SIAM J. Comput.*, 30(6):2008–2035, 2000.

[57] Daniele Micciancio. Closest Vector Problem. In Henk C. A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security (2nd Ed.)*, page 212–214. Springer, 2011.

[58] Daniele Micciancio. Lattice-Based Cryptography. In Henk C. A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security (2nd Ed.)*, page 713–715. Springer, 2011.

[59] Daniele Micciancio. Shortest Vector Problem. In Henk C. A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security (2nd Ed.)*, page 1196–1197. Springer, 2011.

[60] Daniele Micciancio. Inapproximability of the Shortest Vector Problem: Toward a Deterministic Reduction. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:20, 2012.

[61] 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, 2002.

[62] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In Leonard J. Schulman, editor, *STOC*, page 351–358. ACM, 2010.

[63] Daniele Micciancio and Panagiotis Voulgaris. Faster Exponential Time Algorithms for the Shortest Vector Problem. In Moses Charikar, editor, *SODA*, page 1468–1480. SIAM, 2010.

[64] R.A. Mollin. *Advanced Number Theory with Applications*. Discrete Mathematics and Its Applications. Taylor & Francis, 2010.

[65] Jean-Michel Muller, Nicolas Brisebarre, Florent de Dinechin, Claude-Pierre Jeannerod, Vincent Lefèvre, Guillaume Melquiond, Nathalie Revol, Damien Stehlé, and Serge Torres. *Handbook of Floating-Point Arithmetic.* Birkhäuser, 2010.

[66] Phong Q. Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, page 288–304. Springer, 1999.

[67] Phong Q. Nguyen. The Two Faces of Lattices in Cryptology. In Serge Vaudenay and Amr M. Youssef, editors, *Selected Areas in Cryptography*, volume 2259 of *Lecture Notes in Computer Science*, page 313. Springer, 2001.

[68] Phong Q. Nguyen and Oded Regev. Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures. *J. Cryptology*, 22(2):139–160, 2009.

[69] Phong Q. Nguyen and Damien Stehlé. Floating-Point LLL Revisited. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, page 215–233. Springer, 2005.

[70] Phong Q. Nguyen and Jacques Stern. Cryptanalysis of the Ajtai-Dwork Cryptosystem. In Hugo Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, page 223–242. Springer, 1998.

[71] Phong Q. Nguyen and Brigitte Vallée, editors. *The LLL Algorithm: Survey and Applications*. Springer Berlin Heidelberg, 2009.

[72] Phong Q. Nguyen and Thomas Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2(2):181–207, September 2008.

[73] A. M. Odlyzko. The rise and fall of knapsack cryptosystems. In *In Cryptology and Computational Number Theory*, page 75–88. A.M.S, 1990.

[74] Michael Pohst. A Modification of the LLL Reduction Algorithm. *J. Symb. Comput.*, 4(1):123–127, 1987.

[75] Xavier Pujol and Damien Stehlé. Rigorous and Efficient Short Lattice Vectors Enumeration. In *Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '08, page 390–405, Berlin, Heidelberg, 2008. Springer-Verlag.

[76] Xavier Pujol and Damien Stehlé. Solving the Shortest Lattice Vector Problem in Time $2^{2.465n}$. *IACR Cryptology ePrint Archive*, 2009:605, 2009.

[77] Michael Schneider. Sieving for Shortest Vectors in Ideal Lattices. *IACR Cryptology ePrint Archive*, 2011:458, 2011.

[78] C. P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 1994. 10.1007/BF01581144.

[79] Claus-Peter Schnorr. A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

[80] Claus-Peter Schnorr. Fast LLL-type lattice reduction. *Inf. Comput.*, 204(1):1–25, 2006.

[81] Claus-Peter Schnorr. Gitter und Kryptographie (lecture notes), October 2011. Available at http://www.mi.informatik.uni-frankfurt.de/teaching/lecture_notes/index.html.

[82] C.C. Sims. *Computation with finitely presented groups*. Cambridge University Press, 1994.

[83] Brigitte Vallée. Gauss' Algorithm Revisited. *J. Algorithms*, 12(4):556–572, 1991.

[84] Peter van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, Mathematische Instituut, University of Amsterdam, 1981. Available at http://staff.science.uva.nl/ peter/vectors/mi8104c.html.

[85] Jin yi Cai and Ajay Nerurkar. Approximating the SVP to within a factor $(1 + 1/dim^{\epsilon})$ Is NP-Hard under Randomized Reductions. *J. Comput. Syst. Sci.*, 59(2):221–239, 1999.

[86] Éliane Jaulmes and Antoine Joux. A Chosen-Ciphertext Attack against NTRU. In Mihir Bellare, editor, *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, page 20–35. Springer, 2000.

Index