Chapter 1

BASICS

This book is about algorithmic problems on point lattices, and their computational complexity. In this chapter we give some background about lattices and complexity theory.

1. Lattices

Let $\mathbb{R}^m$ be the m-dimensional Euclidean space. A *lattice* in $\mathbb{R}^m$ is the set

$$\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \colon x_i \in \mathbb{Z} \right\} \tag{1.1}$$

of all integral combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in $\mathbb{R}^m$ $(m \geq n)$. The integers n and m are called the *rank* and *dimension* of the lattice, respectively. The sequence of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is called a *lattice basis* and it is conveniently represented as a matrix

$$\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{m \times n} \tag{1.2}$$

having the basis vectors as columns. Using matrix notation, (1.1) can be rewritten in a more compact form as

$$\mathcal{L}(\mathbf{B}) = \{\mathbf{B}\mathbf{x} \colon \mathbf{x} \in \mathbb{Z}^n\} \tag{1.3}$$

where $\mathbf{Bx}$ is the usual matrix-vector multiplication.

Graphically, a lattice can be described as the set of intersection points of an infinite, regular (but not necessarily orthogonal) n-dimensional grid. A 2-dimensional example is shown in Figure 1.1. There, the basis vectors are

$$\mathbf{b}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \qquad \mathbf{b}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{1.4}$$

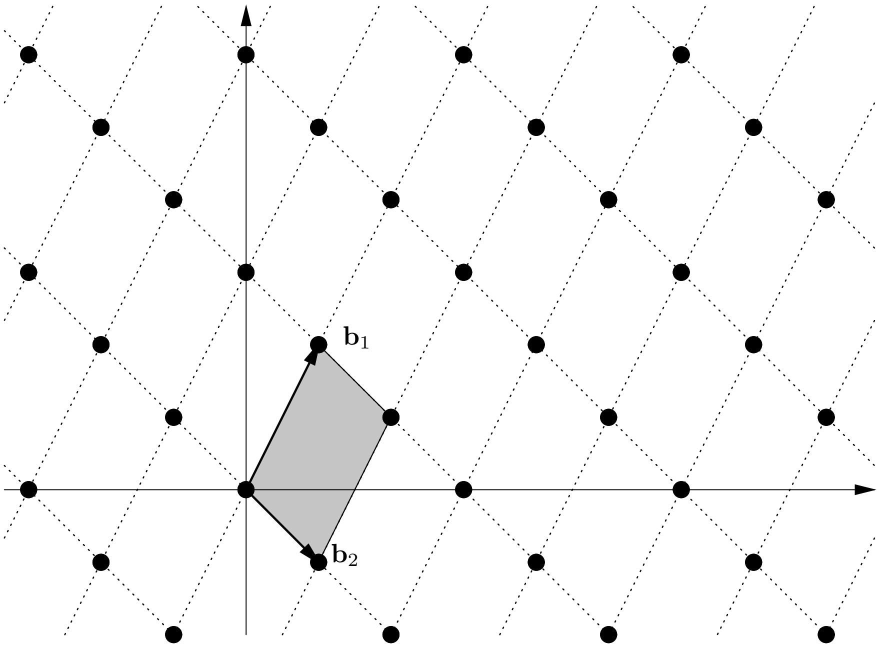

Figure 1.1. A lattice in $\mathbb{R}^2$

and they generate all the intersection points of the grid when combined with integer coefficients. The same lattice has many different bases. For example, vectors

$$\mathbf{b}_1' = \mathbf{b}_1 + \mathbf{b}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \qquad \mathbf{b}_2' = 2\mathbf{b}_1 + \mathbf{b}_2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \tag{1.5}$$

are also a basis for lattice $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2)$. The grid generated by $\mathbf{b}_1', \mathbf{b}_2'$ is shown in Figure 1.2. Notice that although the two grids are different, the set of intersection points is exactly the same, i.e., $\{\mathbf{b}_1, \mathbf{b}_2\}$ and $\{\mathbf{b}_1', \mathbf{b}_2'\}$ are two different bases for the same lattice $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2) = \mathcal{L}(\mathbf{b}_1', \mathbf{b}_2')$.

Throughout the book, we use the convention that lattice points are always represented as *column* vectors. Wherever vectors are more conveniently written as rows, we use transpose notation. For example, the definition of vector $\mathbf{b}_1, \mathbf{b}_2$ in (1.4) can equivalently be rewritten as $\mathbf{b}_1 = [1, 2]^T, \mathbf{b}_2 = [1, -1]^T$, where $\mathbf{A}^T$ denotes the transpose of matrix $\mathbf{A}$.

A simple example of n-dimensional lattice is given by the set $\mathbb{Z}^n$ of all vectors with integral coordinates. A possible basis is given by the

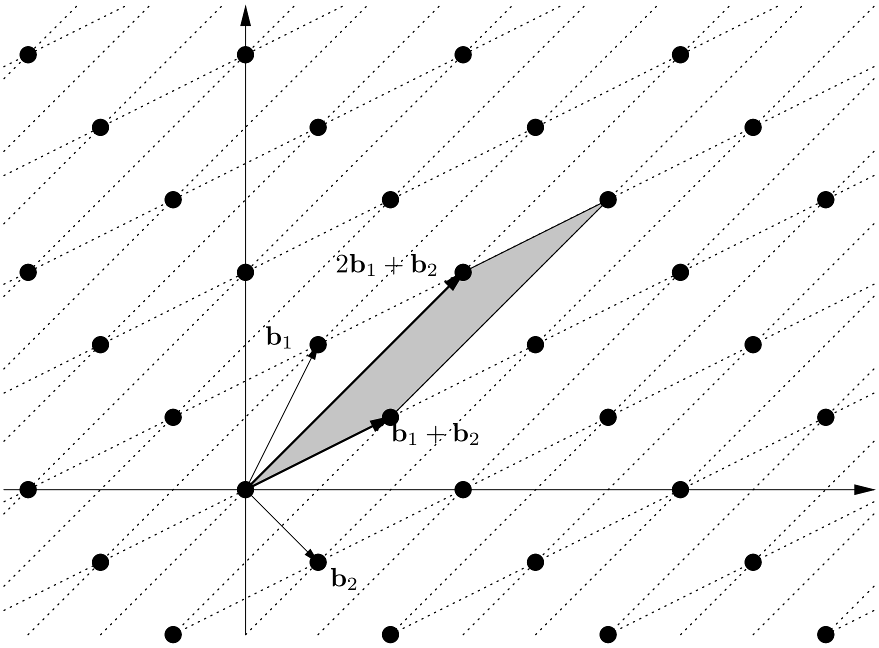

Figure 1.2. A different basis

standard unit vectors

$$\mathbf{e}_i = [\overbrace{\underbrace{0,\ldots,0,1}_{i},0,\ldots,0}^{n}]^T.$$

In matrix notation $\mathbb{Z}^n = \mathcal{L}(\mathbf{I})$ where $\mathbf{I} \in \mathbb{Z}^{n\times n}$ is the n-dimensional identity matrix, i.e., the $n \times n$ square matrix with 1's on the diagonal and 0's everywhere else.

When $n = m$, i.e., the number of basis vectors equals the number of coordinates, we say that $\mathcal{L}(\mathbf{B})$ is *full rank* or *full dimensional.* Equivalently, lattice $\mathcal{L}(\mathbf{B}) \subseteq \mathbb{R}^m$ is full rank if and only if the linear span of the basis vectors

$$\operatorname{span}(\mathbf{B}) = \{\mathbf{B}\mathbf{x} \colon \mathbf{x} \in \mathbb{R}^n\} \tag{1.6}$$

equals the entire space $\mathbb{R}^m$. The difference between (1.3) and (1.6) is that while in (1.6) one can use arbitrary real coefficients to combine the basis vectors, in (1.3) only integer coefficients are allowed. It is easy to see that $\operatorname{span}(\mathbf{B})$ does not depend on the particular basis $\mathbf{B}$, i.e., if $\mathbf{B}$ and $\mathbf{B}'$ generate the same lattice then $\operatorname{span}(\mathbf{B}) = \operatorname{span}(\mathbf{B}')$. So, for any lattice $\Lambda = \mathcal{L}(\mathbf{B})$, we can define the linear span of the lattice $\mathrm{span}(\Lambda)$, without reference to any specific basis. Notice that $\mathbf{B}$ is a basis of $\mathrm{span}(\mathbf{B})$ as a vector space. In particular, the rank of lattice $\mathcal{L}(\mathbf{B})$ equals the dimension of $\mathrm{span}(\mathbf{B})$ as a vector space over $\mathbb{R}$ and it is a lattice invariant, i.e., it does not depend on the choice of the basis.

Clearly, any set of n linearly independent lattice vectors $\mathbf{B}' \in \mathcal{L}(\mathbf{B})$ is a basis for $\mathrm{span}(\mathbf{B})$ as a vector space. However, $\mathbf{B}'$ is not necessarily a lattice basis for $\mathcal{L}(\mathbf{B})$. See Figure 1.3 for a 2-dimensional example. The picture shows the lattice $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2)$ generated by basis vectors (1.4) and the grid associated to lattice vectors

$$\mathbf{b}'_1 = \mathbf{b}_1 + \mathbf{b}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \qquad \mathbf{b}'_2 = \mathbf{b}_1 - \mathbf{b}_2 = \begin{bmatrix} 0 \\ 3 \end{bmatrix}. \tag{1.7}$$

Vectors $\mathbf{b}'_1$ and $\mathbf{b}'_2$ are linearly independent. Therefore, they are a basis for the plane $\mathbb{R}^2 = \mathrm{span}(\mathbf{b}_1, \mathbf{b}_2)$ as a vector space. However, they are not a basis for $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2)$ because lattice point $\mathbf{b}_1$ cannot be expressed as an *integer* linear combination of $\mathbf{b}'_1$ and $\mathbf{b}'_2$. There is a simple geometric characterization for linearly independent lattice vectors that generate the whole lattice. For any n linearly independent lattice vectors $\mathbf{B}' = [\mathbf{b}'_1, \ldots, \mathbf{b}'_n]$ (with $\mathbf{b}'_i \in \mathcal{L}(\mathbf{B}) \subset \mathbb{R}^m$ for all $i = 1, \ldots, n$) define the half open parallelepiped

$$\mathcal{P}(\mathbf{B}') = \{\mathbf{B}'\mathbf{x} \colon 0 \leq x_i < 1\}. \tag{1.8}$$

Then, $\mathbf{B}'$ is a basis for lattice $\mathcal{L}(\mathbf{B})$ if and only if $\mathcal{P}(\mathbf{B}')$ does not contain any lattice vector other than the origin. Figures 1.1, 1.2 and 1.3 illustrate the two cases. The lattice in Figures 1.2 and 1.3 is the same as the one in Figure 1.1. In Figure 1.2, the (half open) parallelepiped $\mathcal{P}(\mathbf{B}')$ does not contain any lattice point other than the origin, and therefore $\mathcal{L}(\mathbf{B}') = \mathcal{L}(\mathbf{B})$. In Figure 1.3, parallelepiped $\mathcal{P}(\mathbf{B}')$ contains lattice point $\mathbf{b}_1$. Therefore $\mathcal{L}(\mathbf{B}') \neq \mathcal{L}(\mathbf{B})$ and $\mathbf{B}'$ is not a basis for $\mathcal{L}(\mathbf{B})$.

Notice that since $\mathbf{B}'$ is a set of linearly independent vectors, $\mathcal{L}(\mathbf{B}')$ is a lattice and $\mathbf{B}'$ is a basis for $\mathcal{L}(\mathbf{B}')$. Clearly, $\mathcal{L}(\mathbf{B}') \subseteq \mathcal{L}(\mathbf{B})$, i.e., any point from lattice $\mathcal{L}(\mathbf{B}')$ belongs also to lattice $\mathcal{L}(\mathbf{B})$. When $\mathcal{L}(\mathbf{B}') \subseteq \mathcal{L}(\mathbf{B})$, we say that $\mathcal{L}(\mathbf{B}')$ is a *sublattice* of $\mathcal{L}(\mathbf{B})$. If $\mathcal{L}(\mathbf{B}') = \mathcal{L}(\mathbf{B})$ we say that bases $\mathbf{B}$ 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 bases $\mathbf{B}$ and $\mathbf{B}'$ are not equivalent, and $\mathcal{L}(\mathbf{B}')$ is a *proper* sublattice of $\mathcal{L}(\mathbf{B})$.

Equivalent bases (i.e., bases that generate the same lattice) can be algebraically characterized as follows. Two bases $\mathbf{B}, \mathbf{B}' \in \mathbb{R}^{m \times n}$ are equivalent if and only if there exists a unimodular matrix $\mathbf{U} \in \mathbb{Z}^{n \times n}$ (i.e., an integral matrix with determinant $\det(\mathbf{U}) = \pm 1$) such that $\mathbf{B}' = \mathbf{B}\mathbf{U}$. The simple proof is left to the reader as an exercise.

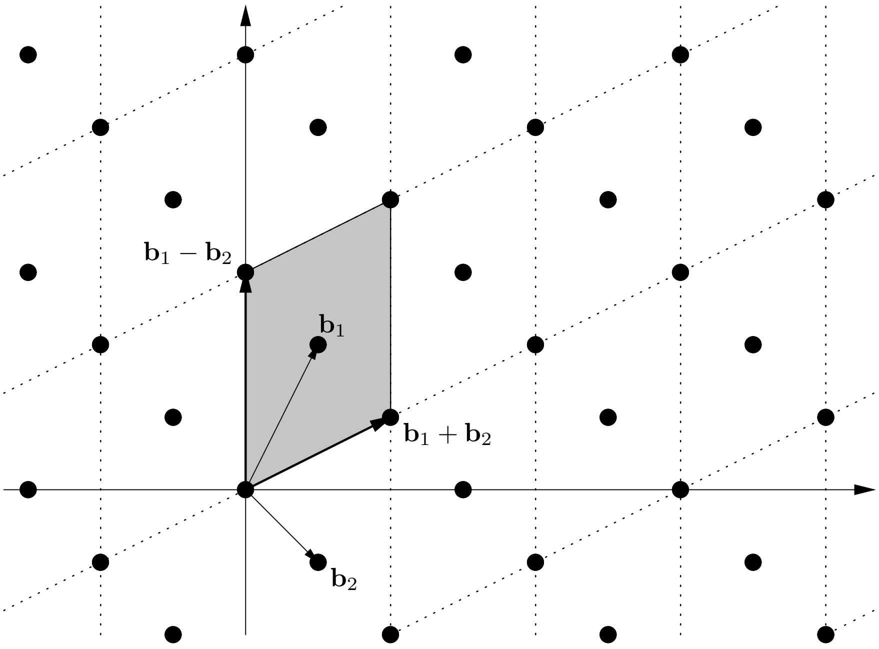

Figure 1.3. The sublattice generated by $\mathbf{b}_1 + \mathbf{b}_2$ and $\mathbf{b}_1 - \mathbf{b}_2$

When studying lattices from a computational point of view, it is customary to assume that the basis vectors (and therefore any lattice vector) have all rational coordinates. It is easy to see that rational lattices can be converted to integer lattices (i.e., sublattices of $\mathbb{Z}^n$) by multiplying all coordinates by an appropriate integer scaling factor. So, without loss of generality, in the rest of this book we concentrate on integer lattices, and, unless explicitly stated otherwise, we always assume that lattices are represented by a basis, i.e., a matrix with integer coordinates such that the columns are linearly independent.

Lattices can also be characterized without reference to any basis. A lattice can be defined as a discrete nonempty subset Λ of $\mathbb{R}^m$ which is closed under subtraction, i.e., if $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda$, then also $\mathbf{x} - \mathbf{y} \in \Lambda$. Here "discrete" means that there exists a positive real $\lambda > 0$ such that the distance between any two lattice vectors is at least λ. A typical example is the set $\Lambda = \{\mathbf{x} \in \mathbb{Z}^n \colon \mathbf{A}\mathbf{x} = \mathbf{0}\}$ of integer solutions of a system of homogeneous linear equations. Notice that Λ always contains the origin $\mathbf{0} = \mathbf{x} - \mathbf{x}$, it is closed under negation (i.e., if $\mathbf{x} \in \Lambda$ then $-\mathbf{x} = \mathbf{0} - \mathbf{x} \in \Lambda$), and addition (i.e., if $\mathbf{x}, \mathbf{y} \in \Lambda$ then $\mathbf{x} + \mathbf{y} = \mathbf{x} - (-\mathbf{y}) \in \Lambda$). In other words, Λ is a discrete additive subgroup of $\mathbb{R}^m$.

1.1 Determinant

The *determinant* of a lattice $\Lambda = \mathcal{L}(\mathbf{B})$, denoted $\det(\Lambda)$, is the n-dimensional volume of the fundamental parallelepiped $\mathcal{P}(\mathbf{B})$ spanned by the basis vectors. (See shaded areas in Figures 1.1 and 1.2.) The determinant is a lattice invariant, i.e., it does not depend on the particular basis used to compute it. This immediately follows from the characterization of equivalent bases as matrices $\mathbf{B}' = \mathbf{BU}$ related by a unimodular transformation $\mathbf{U}$. Geometrically, this corresponds to the intuition that the (n-dimensional) volume of the fundamental parallelepiped $\mathcal{P}(\mathbf{B})$ equals the inverse of the density of the lattice points in $\operatorname{span}(\mathbf{B})$. As an example consider the bases in Figures 1.1 and 1.2. The areas of the fundamental regions (i.e., the shaded parallelepipeds in the pictures) are exactly the same because the two bases generate the same lattice. However, the shaded parallelepiped in Figure 1.3 has a different area (namely, twice as much as the original lattice) because vectors (1.7) only generate a sublattice.

A possible way to compute the determinant is given by the usual *Gram-Schmidt orthogonalization* process. For any sequence of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$, define the corresponding Gram-Schmidt orthogonalized vectors $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ by

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^* \tag{1.9a}$$

$$\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle} \tag{1.9b}$$

where $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^m x_i y_i$ is the inner product in $\mathbb{R}^m$. For every i, $\mathbf{b}_i^*$ is the component of $\mathbf{b}_i$ orthogonal to $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$. In particular, $\operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_i) = \operatorname{span}(\mathbf{b}_1^*, \ldots, \mathbf{b}_i^*)$ and vectors $\mathbf{b}_i^*$ are pairwise orthogonal, i.e., $\langle \mathbf{b}_i^*, \mathbf{b}_j^* \rangle = 0$ for all $i \neq j$. The determinant of the lattice equals the product of the lengths of the orthogonalized vectors

$$\det(\mathcal{L}(\mathbf{B})) = \prod_{i=1}^n \|\mathbf{b}_i^*\| \tag{1.10}$$

where $\|\mathbf{x}\| = \sqrt{\sum_i x_i^2}$ is the usual Euclidean length. We remark that the definition of the orthogonalized vectors $\mathbf{b}_i^*$ depends on the order of the original basis vectors. Given basis matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$, we denote by $\mathbf{B}^*$ the matrix whose columns are the orthogonalized vectors $[\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*]$. Clearly, $\mathbf{B}^*$ is a basis of $\operatorname{span}(\mathbf{B})$ as a vector space. However, $\mathbf{B}^*$ is not usually a lattice basis for $\mathcal{L}(\mathbf{B})$. In particular, not every lattice has a basis consisting of mutually orthogonal vectors.

Notice that if the $\mathbf{b}_i$'s are rational vectors (i.e., vectors with rational coordinates), then also the orthogonalized vectors $\mathbf{b}_i^*$ are rationals. If lattice $\mathcal{L}(\mathbf{B})$ is full dimensional (i.e. $m = n$), then $\mathbf{B}$ is a nonsingular square matrix and $\det(\mathcal{L}(\mathbf{B}))$ equals the absolute value of the determinant of the basis matrix $\det(\mathbf{B})$. For integer lattices, $\mathbf{B}$ is a square integer matrix, and the lattice determinant $\det(\mathcal{L}(\mathbf{B})) = \det(\mathbf{B})$ is an integer. In general, the reader can easily verify that $\det(\mathcal{L}(\mathbf{B}))$ equals the square root of the determinant of the Gram matrix $\mathbf{B}^T\mathbf{B}$, i.e., the $n \times n$ matrix whose (i, j)th entry is the inner product $\langle \mathbf{b}_i, \mathbf{b}_j \rangle$:

$$\det(\mathcal{L}(\mathbf{B})) = \sqrt{\det(\mathbf{B}^T\mathbf{B})}. \tag{1.11}$$

This gives an alternative way to compute the determinant of a lattice (other than computing the Gram-Schmidt orthogonalized vectors), and shows that if $\mathbf{B}$ is an integer matrix, then the determinant of $\mathcal{L}(\mathbf{B})$ is always the square root of a positive integer, even if $\det(\mathcal{L}(\mathbf{B}))$ is not necessarily an integer when the lattice is not full rank.

1.2 Successive minima

Let $\mathcal{B}_m(\mathbf{0}, r) = \{\mathbf{x} \in \mathbb{R}^m : \|\mathbf{x}\| < r\}$ be the m-dimensional open ball of radius r centered in $\mathbf{0}$. When the dimension m is clear from the context, we omit the subscript m and simply write $\mathcal{B}(\mathbf{0}, r)$. Fundamental constants associated to any rank n lattice Λ are its successive minima $\lambda_1, \ldots, \lambda_n$. The ith minimum $\lambda_i(\Lambda)$ is the radius of the smallest sphere centered in the origin containing i linearly independent lattice vectors

$$\lambda_i(\Lambda) = \inf\{r \colon \dim(\mathrm{span}(\Lambda \cap \mathcal{B}(\mathbf{0}, r))) \geq i\}. \tag{1.12}$$

Successive minima can be defined with respect to any norm. A norm is a positive definite, homogeneous function that satisfies the triangle inequality, i.e., a function $\|\cdot\| \colon \mathbb{R}^n \to \mathbb{R}$ such that

- $\|\mathbf{x}\| \geq 0$ with equality only if $\mathbf{x} = \mathbf{0}$
- $\|\alpha\mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$
- $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$. An important family of norm functions is given by the ℓ_p norms. For any $p \geq 1$, the ℓ_p norm of a vector $\mathbf{x} \in \mathbb{R}^n$ is

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} x_i^p\right)^{1/p}. \tag{1.13a}$$

Important special cases are the l_1-norm

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|, \tag{1.13b}$$

the ℓ_2 norm (or Euclidean norm)

$$\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \sqrt{\sum_{i=1}^{n} x_i^2}, \tag{1.13c}$$

and the ℓ_∞ norm (or max-norm)

$$\|\mathbf{x}\|_\infty = \lim_{p \to \infty} \|\mathbf{x}\|_p = \max_{i=1}^{n} |x_i|. \tag{1.13d}$$

We remark that when $p < 1$, function (1.13) is not a norm because it does not satisfy the triangle inequality. Notice that the value of the successive minima $\lambda_1, \ldots, \lambda_n$, and the lattice vectors achieving them, depend on the norm being used. Consider for example the lattice

$$\Lambda = \{\mathbf{v} \in \mathbb{Z}^2 \colon v_1 + v_2 = 0 \bmod 2\} \tag{1.14}$$

generated by basis vectors

$$\mathbf{b}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \qquad \mathbf{b}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{1.15}$$

Lattice vector $\mathbf{b}_1$ is a shortest (nonzero) vector in $\mathcal{L}(\mathbf{b}_1, \mathbf{b}_2)$ with respect the ℓ_1 norm and $\lambda_1 = \|\mathbf{b}_1\|_1 = 2$ if the ℓ_1 norm is used. However, $\mathbf{b}_1$ is not a shortest vector with respect to the ℓ_2 or ℓ_∞ because in these norms lattice vector $\mathbf{b}_2$ is strictly shorter than $\mathbf{b}_1$ giving first minimum $\lambda_1 = \|\mathbf{b}_2\|_2 = \sqrt{2}$ and $\lambda_1 = \|\mathbf{b}_2\|_\infty = 1$, respectively. In this book we are primarily concerned with the ℓ_2 norm, which corresponds to the familiar Euclidean distance

$$\operatorname{dist}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1.16}$$

and will consider other norms only when it can be done without substantially complicating the exposition.

In the previous examples, we have seen that lattice (1.14) contains a vector $\mathbf{b}$ such that $\|\mathbf{b}\| = \lambda_1$. It turns out that this is true for every lattice. It easily follows from the characterization of lattices as discrete subgroups of $\mathbb{R}^n$ that there always exist vectors achieving the successive minima, i.e., there are linearly independent vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \Lambda$ such that $\|\mathbf{x}_i\| = \lambda_i$ for all $i = 1, \ldots, n$. So, the infimum in (1.12) is actually a minimum if $\mathcal{B}(\mathbf{0}, r)$ is replaced with the closed ball $\bar{\mathcal{B}}(\mathbf{0}, r) = \{\mathbf{x} \in \mathbb{R}^m : \|\mathbf{x}\| \leq r\}$. In particular, $\lambda_1(\Lambda)$ is the length of the shortest nonzero lattice vector and equals the minimum distance between any two distinct lattice points

$$\lambda_1(\Lambda) = \min_{\mathbf{x} \neq \mathbf{y} \in \Lambda} \|\mathbf{x} - \mathbf{y}\| = \min_{\mathbf{x} \in \Lambda \setminus \{\mathbf{0}\}} \|\mathbf{x}\|. \tag{1.17}$$

In the rest of this section we give a proof that any lattice contains nonzero vectors of minimal length. In doing so, we prove a lower bound for the first minimum that will be useful later on. The result is easily generalized to all successive minima to show that there are n linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ satisfying $\|\mathbf{v}_i\| = \lambda_i$ for all $i = 1, \ldots, n$. Fix some lattice $\mathcal{L}(\mathbf{B})$, and consider the first minimum

$$\lambda_1 = \inf\{\|\mathbf{v}\| : \mathbf{v} \in \mathcal{L}(\mathbf{B})/\{\mathbf{0}\}\}.$$

We want to prove that there exists a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v}\| = \lambda_1$. We first prove that λ_1 is strictly positive.

THEOREM 1.1 *Let $\mathbf{B}$ be a lattice basis, and let $\mathbf{B}^*$ be the corresponding Gram-Schmidt orthogonalization. Then, the first minimum of the lattice (in the ℓ_2 norm) satisfies*

$$\lambda_1 \geq \min_j \|\mathbf{b}_j^*\| > 0.$$

Proof: Consider a generic nonzero lattice vector $\mathbf{Bx}$ (where $\mathbf{x} \in \mathbb{Z}^n$ and $\mathbf{x} \neq \mathbf{0}$) and let i be the biggest index such that $x_i \neq 0$. We show that $\|\mathbf{Bx}\| \geq \|\mathbf{b}_i^*\| \geq \min_j \|\mathbf{b}_j^*\|$. It follows that the infimum $\lambda_1 = \inf \|\mathbf{Bx}\|$ also satisfies $\lambda_1 \geq \min_j \|\mathbf{b}_j^*\|$. From basic linear algebra we know that $|\langle \mathbf{x}, \mathbf{y}\rangle| \leq \|\mathbf{x}\| \cdot \|\mathbf{y}\|$ for any two vectors $\mathbf{x}, \mathbf{y}$. We prove that $|\langle \mathbf{Bx}, \mathbf{b}_i^*\rangle| \geq \|\mathbf{b}_i^*\|^2$, and therefore $\|\mathbf{Bx}\| \cdot \|\mathbf{b}_i^*\| \geq \|\mathbf{b}_i^*\|^2$. Since vectors $\mathbf{b}_i$'s are linearly independent, $\|\mathbf{b}_i^*\| \neq 0$ and $\|\mathbf{Bx}\| \geq \|\mathbf{b}_i^*\|$ follows.

So, let us prove that $|\langle \mathbf{Bx}, \mathbf{b}_i^*\rangle| \geq \|\mathbf{b}_i^*\|^2$. From the definition of i, we know that $\mathbf{Bx} = \sum_{j=1}^{i} \mathbf{b}_j x_j$. Using the definition of the orthogonalized vectors (1.9a) we get

$$\begin{aligned} \langle \mathbf{Bx}, \mathbf{b}_i^*\rangle &= \sum_{j=1}^{i} \langle \mathbf{b}_j, \mathbf{b}_i^*\rangle x_j \\ &= \langle \mathbf{b}_i, \mathbf{b}_i^*\rangle x_i \end{aligned}$$

$$\begin{aligned} &= \langle \mathbf{b}_i^* + \sum_{j<i} \mu_{ij}\mathbf{b}_j^*, \mathbf{b}_i^* \rangle x_i \\ &= \langle \mathbf{b}_i^*, \mathbf{b}_i^* \rangle x_i + \sum_{j<i} \mu_{ij} \langle \mathbf{b}_j^*, \mathbf{b}_i^* \rangle x_i \\ &= \|\mathbf{b}_i^*\|^2 x_i. \end{aligned}$$

Since x_i is a nonzero integer,

$$|\langle \mathbf{B}\mathbf{x}, \mathbf{b}_i^* \rangle| = \|\mathbf{b}_i^*\|^2 \cdot |x_i| \geq \|\mathbf{b}_i^*\|^2. \qquad \square$$

In particular, the theorem shows that $\lambda_1 > 0$. We now prove that there exists a nonzero lattice vector of length λ_1. By definition of λ_1, there exists a sequence of lattice vectors $\mathbf{v}_i \in \mathcal{L}(\mathbf{B})$ such that

$$\lim_{i\to\infty} \|\mathbf{v}_i\| = \lambda_1.$$

Since $\lambda_1 > 0$, for all sufficiently large i it must be $\|\mathbf{v}_i\| \leq 2\lambda_1$, i.e., lattice vector $\mathbf{v}_i$ belongs to the closed ball

$$\bar{\mathcal{B}}(\mathbf{0}, 2\lambda_1) = \{\mathbf{z} : \|\mathbf{z}\| \leq 2\lambda_1\}.$$

But set $\bar{\mathcal{B}}(0, 2\lambda)$ is compact, so, we can extract a convergent subsequence $\mathbf{v}_{i_j}$ with limit

$$\mathbf{w} = \lim_{j\to\infty} \mathbf{v}_{i_j}.$$

Clearly, $\|\mathbf{w}\| = \lim_{j\to\infty} \|\mathbf{v}_{i_j}\| = \lambda_1$. We want to prove that $\mathbf{w}$ is a lattice vector. By definition of $\mathbf{w}$ we have $\lim_{j\to\infty} \|\mathbf{v}_{i_j} - \mathbf{w}\| = 0$. Therefore for all sufficiently large j, $\|\mathbf{v}_{i_j} - \mathbf{w}\| < \lambda_1/2$. By triangle inequality, for a sufficiently large j and all $k > j$,

$$\|\mathbf{v}_{i_j} - \mathbf{v}_{i_k}\| \leq \|\mathbf{v}_{i_j} - \mathbf{w}\| + \|\mathbf{w} - \mathbf{v}_{i_k}\| < \lambda_1.$$

But $\mathbf{v}_{i_j} - \mathbf{v}_{i_k}$ is a lattice vector, and no nonzero lattice vector can have length strictly less than λ_1. This proves that $\mathbf{v}_{i_j} - \mathbf{v}_{i_k} = \mathbf{0}$, i.e., $\mathbf{v}_{i_k} = \mathbf{v}_{i_j}$ for all $k > j$. Therefore, $\mathbf{w} = \lim_k \mathbf{v}_{i_k} = \mathbf{v}_{i_j}$, and $\mathbf{w}$ is a lattice vector.

The above argument can be easily generalized to prove the following theorem about all successive minima of a lattice.

THEOREM 1.2 *Let Λ be a lattice of rank n with successive minima λ_1, $\ldots$, λ_n. Then there exist linearly independent lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \Lambda$ such that $\|\mathbf{v}_i\| = \lambda_i$ for all $i = 1, \ldots, n$.*

Interestingly, the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ achieving the minima are not necessarily a basis for Λ. Examples of lattices for which all bases must contain at least one vector strictly longer than λ_n are given in Chapter 7.

1.3 Minkowski's theorems

In this subsection we prove an important upper bound on the product of successive minima of any lattice. The bound is based on the following fundamental theorem.

THEOREM 1.3 (BLICHFELDT THEOREM.) *For any lattice* Λ *and for any measurable set* $S \subseteq$ *span*(Λ)*, if* S *has volume* $\mathrm{vol}(S) > \det(\Lambda)$*, then there exist two distinct points* $\mathbf{z}_1, \mathbf{z}_2 \in S$ *such that* $\mathbf{z}_1 - \mathbf{z}_2 \in \Lambda$*.*

Proof: Let $\Lambda = \mathcal{L}(\mathbf{B})$ be a lattice and S be any subset of $\mathrm{span}(\Lambda)$ such that $\mathrm{vol}(S) > \det(\mathbf{B})$. Partition S into a collection of disjoint regions as follows. For any lattice point $\mathbf{x} \in \Lambda$ define

$$S_{\mathbf{x}} = S \cap (\mathcal{P}(\mathbf{B}) + \mathbf{x}) \tag{1.18}$$

where $\mathcal{P}(\mathbf{B})$ is the half open parallelepiped (1.8). Here and below, for any set $A \subset \mathbb{R}^n$ and vector $\mathbf{x} \in \mathbb{R}^n$, expression $A + \mathbf{x}$ denotes the set $\{\mathbf{y} + \mathbf{x} \colon \mathbf{y} \in A\}$. Notice that sets $\mathcal{P}(\mathbf{B}) + \mathbf{x}$ (with $\mathbf{x} \in \Lambda$) partition $\mathrm{span}(\mathbf{B})$. Therefore sets $S_{\mathbf{x}}$ ($\mathbf{x} \in \Lambda$) form a partition of S, i.e., they are pairwise disjoint and

$$S = \bigcup_{\mathbf{x} \in \Lambda} S_{\mathbf{x}}.$$

In particular, since Λ is countable,

$$\mathrm{vol}(S) = \sum_{\mathbf{x} \in \Lambda} \mathrm{vol}(S_{\mathbf{x}}).$$

Define also translated sets

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

Notice that for all $\mathbf{x} \in \Lambda$, set $S'_{\mathbf{x}}$ is contained in $\mathcal{P}(\mathbf{B})$ and $\mathrm{vol}(S_{\mathbf{x}}) = \mathrm{vol}(S'_{\mathbf{x}})$. We claim that sets $S'_{\mathbf{x}}$ are not pairwise disjoint. Assume, for contradiction, they are. Then, we have

$$\sum_{\mathbf{x} \in \Lambda} \mathrm{vol}(S'_{\mathbf{x}}) = \mathrm{vol}\left(\bigcup_{\mathbf{x} \in \Lambda} S'_{\mathbf{x}}\right) \leq \mathrm{vol}(\mathcal{P}(\mathbf{B})). \tag{1.19}$$

We also know from the assumption in the theorem that

$$\sum_{\mathbf{x} \in \Lambda} \mathrm{vol}(S'_{\mathbf{x}}) = \sum_{\mathbf{x} \in \Lambda} \mathrm{vol}(S_{\mathbf{x}}) = \mathrm{vol}(S) > \det(\Lambda). \tag{1.20}$$

Combining (1.19) and (1.20) we get $\det(\Lambda) < \mathrm{vol}(\mathcal{P}(\mathbf{B}))$, which is a contradiction because $\det(\Lambda) = \mathrm{vol}(\mathcal{P}(\mathbf{B}))$ by the definition of lattice determinant.

This proves that set $S'_{\mathbf{y}}$ are not pairwise disjoint, i.e., there exist two sets $S'_{\mathbf{x}}$, $S'_{\mathbf{y}}$ (for $\mathbf{x}, \mathbf{y} \in \Lambda$) such that $S'_{\mathbf{x}} \cap S'_{\mathbf{y}} \neq \emptyset$. Let $\mathbf{z}$ be any vector in the (nonempty) intersection $S'_{\mathbf{x}} \cap S'_{\mathbf{y}}$ and define

$$\begin{aligned} \mathbf{z}_1 &= \mathbf{z} + \mathbf{x} \\ \mathbf{z}_2 &= \mathbf{z} + \mathbf{y}. \end{aligned}$$

From $\mathbf{z} \in S'_{\mathbf{x}}$ and $\mathbf{z} \in S'_{\mathbf{y}}$ we get $\mathbf{z}_1 \in S_{\mathbf{x}} \subseteq S$ and $\mathbf{z}_2 \in S_{\mathbf{y}} \subseteq S$. Moreover, $\mathbf{z}_1 \neq \mathbf{z}_2$ because $\mathbf{x} \neq \mathbf{y}$. Finally, the difference between $\mathbf{z}_1$ and $\mathbf{z}_2$ satisfies

$$\mathbf{z}_1 - \mathbf{z}_2 = \mathbf{x} - \mathbf{y} \in \Lambda, \tag{1.21}$$

completing the proof of the theorem. □

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

Theorem 1.4 (Convex body theorem) *For any lattice Λ of rank n and any convex set $S \subset span(\Lambda)$ symmetric about the origin, if* $\mathrm{vol}(S) > 2^n \det(\Lambda)$*, then S contains a nonzero lattice point* $\mathbf{v} \in S \cap \Lambda \setminus \{\mathbf{0}\}$.

Proof: Consider the set $S' = \{\mathbf{x} \colon 2\mathbf{x} \in S\}$. The volume of S' satisfies

$$\mathrm{vol}(S') = 2^{-n}\,\mathrm{vol}(S) > \det(\Lambda). \tag{1.22}$$

Therefore, by Blichfeldt theorem there exist two distinct points $\mathbf{z}_1, \mathbf{z}_2 \in S'$ such that $\mathbf{z}_1 - \mathbf{z}_2 \in \mathcal{L}(\Lambda)$. From the definition of S', we get $2\mathbf{z}_1, 2\mathbf{z}_2 \in S$ and since S is symmetric about the origin, we also have $-2\mathbf{z}_2 \in S$. Finally, by convexity, the midpoint of segment $[2\mathbf{z}_1, -2\mathbf{z}_2]$ also belongs to S, i.e.,

$$\frac{2\mathbf{z}_1 + (-2\mathbf{z}_2)}{2} = \mathbf{z}_1 - \mathbf{z}_2 \in S. \tag{1.23}$$

This proves that $\mathbf{v} = \mathbf{z}_1 - \mathbf{z}_2$ is a nonzero lattice point in S. □

Minkowski's convex body theorem can be used to bound the length of the shortest nonzero vector in an rank n lattice as follows. Let $S = \mathcal{B}(\mathbf{0}, \sqrt{n}\det(\Lambda)^{1/n}) \cap \mathrm{span}(\Lambda)$ be the open ball of radius $\sqrt{n}\det(\Lambda)^{1/n}$ in $\mathrm{span}(\Lambda)$. Notice that S has volume strictly bigger than $2^n \det(\Lambda)$ because it contains an n-dimensional hypercube with edges of length $2\det(\Lambda)^{1/n}$. By Minkowski's theorem there exists a nonzero lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B}) \setminus \{\mathbf{0}\}$ such that $\mathbf{v} \in S$, i.e., $\|\mathbf{v}\| < \sqrt{n}\det(\Lambda)^{1/n}$. This proves that for any rank n lattice Λ, the length of the shortest nonzero vector (in the ℓ_2 norm) satisfies

$$\lambda_1 < \sqrt{n}\det(\Lambda)^{1/n}. \tag{1.24}$$

This result (in a slightly stronger form) is the well known *Minkowski's first theorem.* Minkowski also proved a stronger result involving all successive minima, known as the *second theorem* of Minkowski. Namely, $\sqrt{n}\det(\Lambda)^{1/n}$ is an upper bound not only to the first minimum λ_1, but also to the the geometric mean of all successive minima. While Minkowski's first theorem is easily generalized to any norm, the proof of the second theorem for general norms is relatively complex. Here we prove the theorem only for the simple case of the Euclidean norm.

Theorem 1.5 (Minkowski's second theorem) *For any rank n lattice $\mathcal{L}(\mathbf{B})$, the successive minima (in the ℓ_2 norm) $\lambda_1, \ldots, \lambda_n$ satisfy*

$$\left(\prod_{i=1}^{n} \lambda_i\right)^{1/n} < \sqrt{n}\det(\mathbf{B})^{1/n}. \tag{1.25}$$

Proof: Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be linearly independent lattice vectors achieving the successive minima $\|\mathbf{x}_i\| = \lambda_i$ and assume for contradiction that $\prod_{i=1}^{n} \lambda_i \geq (\sqrt{n})^n \det(\mathbf{B})$. Consider the Gram-Schmidt orthogonalized vectors $\mathbf{x}_i^*$ and define the transformation

$$T\left(\sum c_i \mathbf{x}_i^*\right) = \sum \lambda_i c_i \mathbf{x}_i^* \tag{1.26}$$

that expands each coordinate $\mathbf{x}_i^*$ by a factor λ_i. Let $S = \mathcal{B}(\mathbf{0}, 1) \cap \operatorname{span}(\mathbf{B})$ be the n-dimensional open unit ball in $\operatorname{span}(\mathbf{B})$. If we apply T to S we get a symmetric convex body $T(S)$ of volume

$$\begin{aligned} \operatorname{vol}(T(S)) &= \left(\prod_i \lambda_i\right) \operatorname{vol}(S) \\ &\geq (\sqrt{n})^n \det(\mathbf{B}) \operatorname{vol}(S) \\ &= \operatorname{vol}(\sqrt{n}S) \det(\mathbf{B}) \end{aligned}$$

where $\sqrt{n}S$ is the ball of radius $\sqrt{n}$. The volume of $\sqrt{n}S$ is bigger than 2^n because $\sqrt{n}S$ contains a hypercube with edges of length 2. Therefore, $\operatorname{vol}(T(S)) > 2^n \det(\mathbf{B})$, and by Minkowski's convex body theorem $T(S)$ contains a lattice point $\mathbf{y}$ different from the origin. Since $\mathbf{y} \in T(S)$, it must be $\mathbf{y} = T(\mathbf{x})$ for some $\mathbf{x} \in S$. From the definition of S we get $\|\mathbf{x}\| < 1$. Now express $\mathbf{x}$ and $\mathbf{y}$ in terms of the orthogonalized basis

$$\begin{aligned} \mathbf{x} &= \sum_{i=1}^{n} c_i \mathbf{x}_i^* \\ \mathbf{y} &= \sum \lambda_i c_i \mathbf{x}_i^*. \end{aligned}$$

Since $\mathbf{y}$ is nonzero, some c_i is not zero. Let k be the largest index such that $c_i \neq 0$, and $k' \leq k$ the smallest index such that $\lambda_{k'} = \lambda_k$. Notice that $\mathbf{y}$ is linearly independent from $\mathbf{x}_1, \ldots, \mathbf{x}_{k'-1}$ because $\langle \mathbf{x}_k^*, \mathbf{y} \rangle = \lambda_k c_k \|\mathbf{x}_k^*\|^2 \neq 0$ and $\mathbf{x}_k^*$ is orthogonal to $\mathbf{x}_1, \ldots, \mathbf{x}_{k'-1}$. We now show that $\|\mathbf{y}\| < \lambda_k$.

$$\begin{aligned}
\|\mathbf{y}\|^2 &= \left\| \sum_{i \leq k} \lambda_i c_i \mathbf{x}_i^* \right\|^2 \\
&= \sum_{i \leq k} \lambda_i^2 c_i^2 \|\mathbf{x}_i^*\|^2 \\
&\leq \sum_{i \leq k} \lambda_k^2 c_i^2 \|\mathbf{x}_i^*\|^2 \\
&= \lambda_k^2 \left\| \sum_{i \leq k} c_i \mathbf{x}_i^* \right\|^2 \\
&= \lambda_k^2 \|\mathbf{x}\|^2 < \lambda_k^2.
\end{aligned}$$

This proves that $\mathbf{x}_1, \ldots, \mathbf{x}_{k'-1}, \mathbf{y}$ are k' linearly independent lattice vectors of length strictly less than $\lambda_k = \lambda_{k'}$, contradicting the definition of the k'th successive minimum $\lambda_{k'}$. □

2. Computational problems

Minkowski's first theorem gives a simple way to bound the length λ_1 of the shortest nonzero vector in a lattice $\mathcal{L}(\mathbf{B})$. Although Minkowski's bound is asymptotically tight in the worst case (i.e., there exist lattices such that $\lambda_1 > c\sqrt{n}\det(\mathbf{B})^{1/n}$ for some absolute constant c independent of n), in general λ_1 can be much smaller than $\sqrt{n}\det(\mathbf{B})^{1/n}$. For example, consider the two dimensional lattice generated by orthogonal vectors $\mathbf{b}_1 = \epsilon\mathbf{e}_1$ and $\mathbf{b}_1 = (1/\epsilon)\mathbf{e}_2$. The determinant of the lattice is 1, giving upper bound $\lambda_1 \leq \sqrt{2}$. However $\lambda_1 = \epsilon$ can be arbitrarily small.

Moreover, the proof of Minkowski's theorem is not constructive, in the sense that we know from the theorem that a short nonzero vector exists, but the proof does not give any computational method to efficiently find vectors of length bounded by $\sqrt{n}\det(\Lambda)^{1/n}$, leave alone vectors of length λ_1. The problem of finding a lattice vector of length λ_1 is the well known *Shortest Vector Problem.*

Definition 1.1 (Shortest Vector Problem, SVP) *Given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, find a nonzero lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$) such that $\|\mathbf{Bx}\| \leq \|\mathbf{By}\|$ for any other $\mathbf{y} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$.*

The lack of efficient algorithms to solve SVP has led computer scientists to consider approximation versions of the problem. In this book we study this and other lattice problems from a computational point of view. Throughout the book, we assume the standard computational model of deterministic Turing machines. The reader is referred to (van Emde Boas, 1990; Johnson, 1990) or any undergraduate level textbook on the subject for an introduction to the basic theory of computability and computational complexity. In the following subsection we simply recall some terminology and basic definitions. Then, in Subsection 2.2 we describe SVP and other lattice problems in their exact and approximation versions, and in Subsection 2.3 we give some background about the computational complexity of approximation problems.

2.1 Complexity Theory

An *alphabet* is a finite set of symbols Σ. A *string* (over Σ) is a finite sequence of symbols from Σ. The *length* of a string y is the number of symbols in y, and it is denoted $|y|$. The set of all strings over Σ is denoted Σ^*, and the set of all strings of length n is denoted Σ^n. A Turing machine M runs in time $t(n)$ if for every input string w of length n (over some fixed input alphabet Σ), $M(n)$ halts after at most $t(n)$ steps. We identify the notion of efficient computation with Turing machines that halt in time polynomial in the size of the input, i.e., Turing machines that run in time $t(n) = a + n^b$ for some constants a, b independent of n. A *decision problem* is the problem of deciding whether the input string satisfies or not some specified property. Formally, a decision problem is specified by a *language*, i.e., a set of strings $L \subseteq \Sigma^*$, and the problem is given an input string $w \in \Sigma^*$ decide whether $w \in L$ or not. The class of decision problems that can be solved by a deterministic Turing machine in polynomial time is called P. The class of decision problem that can be solved by a nondeterministic Turing machine in polynomial time is called NP. Equivalently, NP can be characterized as the set of all languages L for which there exists a relation $R \subseteq \Sigma^* \times \Sigma^*$ such that $(x, y) \in R$ can be checked in time polynomial in $|x|$, and $x \in L$ if and only if there exists a string y with $(x, y) \in R$. Such string y is called NP-witness or NP-certificate of membership of x in L. Clearly, P $\subseteq$ NP, but it is widely believed that P $\neq$ NP, i.e., there are NP problems that cannot be solved in deterministic polynomial time.

Let A and B be two decision problems. A *(Karp) reduction* from A to B is a polynomial time computable function $f : \Sigma^* \to \Sigma^*$ such that $x \in A$ if and only if $f(x) \in B$. Clearly, if A reduces to B and B can be solved in polynomial time, then also A can be solved in polynomial time. A (decision) problem A is NP-*hard* if any other NP problem B

reduces to A. If A is also in NP, then A is NP-*complete*. Clearly, if a problem A is NP-hard, then A cannot be solved in polynomial time unless P = NP. The standard technique to prove that a problem A is NP-hard (and therefore no polynomial time solution for A is likely to exists) is to reduce some other NP-hard problem B to A. Another notion of reduction which will be used in this book is that of *Cook reduction*. A Cook reduction from A to B is a polynomial time Turing machine $\mathcal{M}$ with access to an oracle that takes instances of problem B as input. $\mathcal{M}$ reduces A to B, if, given an oracle that correctly solves problem B, $\mathcal{M}$ correctly solves problem A. A problem A is NP-hard under Cook reductions if for any NP problem B there is a Cook reduction from B to A. If A is in NP, then we say that A is NP-complete under Cook reductions. NP-hardness under Cook reductions also gives evidence of the intractability of a problem, because if A can be solved in polynomial time then P = NP. The reader is referred to (Garey and Johnson, 1979) for an introduction to the theory of NP-completeness and various NP-complete problems that will be used throughout the book.

In the rest of this book algorithms and reductions between lattice problems are described using some informal high level language, and decision problems are described as sets of mathematical objects, like graphs, matrices, etc. In all cases, the translation to strings, languages and Turing machines is straightforward.

Occasionally, we will make use of other complexity classes and different notions of reductions, e.g., randomized complexity classes or nonuniform reductions. When needed, these notions will be briefly recalled, or references will be given.

Throughout the book, we use the standard asymptotic notation to describe the order of growth of functions: for any positive real valued functions $f(n)$ and $g(n)$ we write

- $f = O(g)$ if there exists two constants a, b such that $f(n) \leq a \cdot f(n)$ for all $n \geq b$.
- $f = o(g)$ if $\lim_{n\to\infty} f(n)/g(n) = 0$
- $f = \Omega(g)$ if $g = O(f)$
- $f = \omega(g)$ if $g = o(f)$
- $f = \Theta(g)$ if $f = O(g)$ and $g = O(f)$.

A function f is negligible if $f = o(1/g)$ for any polynomial $g(n) = n^c$.

2.2 Some lattice problems

To date, we do not know any polynomial time algorithm to solve SVP. In fact, we do not even know how to find nonzero lattice vectors of length within the Minkowski's bound $\|\mathbf{Bx}\| < \sqrt{n}\det(\mathbf{B})^{1/n}$. Another related problem for which no polynomial time solution is known is the *Closest Vector Problem* .

Definition 1.2 (Closest Vector Problem, CVP) *Given a lattice basis* $\mathbf{B} \in \mathbb{Z}^{m\times n}$ *and a target vector* $\mathbf{t} \in \mathbb{Z}^m$*, find a lattice vector* $\mathbf{Bx}$ *closest to the target* $\mathbf{t}$*, i.e., find an integer vector* $\mathbf{x} \in \mathbb{Z}^n$ *such that* $\|\mathbf{Bx} - \mathbf{t}\| \leq \|\mathbf{By} - \mathbf{t}\|$ *for any other* $\mathbf{y} \in \mathbb{Z}^n$*.*

Studying the computational complexity of these problems is the main subject of this book. Both for CVP and SVP one can consider different algorithmic tasks. These are (in decreasing order of difficulty):

- The *Search Problem*: Find a (nonzero) lattice vector $\mathbf{x} \in \Lambda$ such that $\|\mathbf{x} - \mathbf{t}\|$ (respectively, $\|\mathbf{x}\|$) is minimized.

- The *Optimization Problem*: Find the minimum of $\|\mathbf{x} - \mathbf{t}\|$ (respectively, $\|\mathbf{x}\|$) over $\mathbf{x} \in \Lambda$ (respectively, $\mathbf{x} \in \Lambda \setminus \{\mathbf{0}\}$).

- The *Decision Problem*: Given a rational $r > 0$, decide whether there is a (nonzero) lattice vector $\mathbf{x}$ such that $\|\mathbf{x} - \mathbf{t}\| \leq r$ (respectively,. $\|\mathbf{x}\| \leq r$).

We remark that to date virtually all known (exponential time) algorithms for SVP and CVP solve the search problem (and therefore also the associated optimization and decision problems), while all known hardness results hold for the decision problem (and therefore imply the hardness of the optimization and search problems as well). This suggests that the hardness of solving SVP and CVP is already captured by the decisional task of determining whether or not there exists a solution below some given threshold value. We will see in Chapter 3 that the decision problem associated to CVP is NP-complete, and therefore no algorithm can solve CVP in deterministic polynomial time, unless P = NP. A similar result holds (under randomized reductions) for SVP (see Chapter 4).

The hardness of solving SVP and CVP has led computer scientists to consider approximation versions of these problems. Approximation algorithms return solutions that are only guaranteed to be within some specified factor γ from the optimal. Approximation versions for the SVP and CVP search problems are defined below.

DEFINITION 1.3 (APPROXIMATE SVP) *Given a basis $\mathbf{B} \in \mathbb{Z}^{m\times n}$, find a nonzero lattice vector $\mathbf{Bx}$ ($\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$) such that $\|\mathbf{Bx}\| \leq \gamma \cdot \|\mathbf{By}\|$ for any other $\mathbf{y} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$.*

In the optimization version of approximate SVP, one only needs to find $\|\mathbf{Bx}\|$, i.e., a value d such that $\lambda_1(\mathbf{B}) \leq d < \gamma\lambda_1(\mathbf{B})$.

DEFINITION 1.4 (APPROXIMATE CVP) *Given a basis $\mathbf{B} \in \mathbb{Z}^{m\times n}$ and a target vector $\mathbf{t} \in \mathbb{Z}^m$, find a lattice vector $\mathbf{Bx}$ ($\mathbf{x} \in \mathbb{Z}^n$) such that $\|\mathbf{Bx} - \mathbf{t}\| \leq \gamma\|\mathbf{By} - \mathbf{t}\|$ for any other $\mathbf{y} \in \mathbb{Z}^n$.*

In the optimization version of approximate CVP, one only need to find $\|\mathbf{Bx} - \mathbf{t}\|$, i.e., a value d such that $\operatorname{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B})) \leq d < \gamma \operatorname{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$. Both in the approximate SVP and CVP, the approximation factor γ can be a function of any parameter associated to the lattice, typically its rank n, to capture the fact that the problem gets harder as this parameter increases. To date, the best known polynomial time (possibly randomized) approximation algorithms for SVP and CVP achieve worst case (over the choice of the input) approximation factors $\gamma(n)$ that are essentially exponential in the rank n. Finding algorithms that achieve polynomial approximation factors $\gamma(n) = n^c$ (for some constant c independent of the rank n) is one of the main open problems in this area.

SVP and CVP are the two main problems studied in this book. Chapter 2 describes efficient algorithms to find approximate solutions to these problems (for large approximation factors). The computational complexity of CVP is studied in Chapter 3. The strongest known hardness result for SVP is the subject of Chapters 4, 5 and 6. There are many other lattice problems which are thought to be computationally hard. Some of them, which come up in the construction of lattice based cryptographic functions, are discussed in Chapter 7. There are also many computational problems on lattices that can be efficiently solved (in deterministic polynomial time). Here we recall just a few of them. Finding polynomial time solutions to these problems is left to the reader as an exercise.

1 *Membership:* Given a basis $\mathbf{B}$ and a vector $\mathbf{x}$, decide whether $\mathbf{x}$ belongs to the lattice $\mathcal{L}(\mathbf{B})$. This problem is essentially equivalent to solving a system of linear equations over the integers. This can be done in polynomially many arithmetic operations, but some care is needed to make sure the numbers involved do not get exponentially large.

2 *Kernel:* Given an integral matrix $\mathbf{A} \in \mathbb{Z}^{n\times m}$, compute a basis for the lattice $\{\mathbf{x} \in \mathbb{Z}^m \colon \mathbf{Ax} = \mathbf{0}\}$. A similar problem is, given a modulus M

and a matrix $\mathbf{A} \in \mathbb{Z}_M^{n \times m}$, find a basis for the lattice $\{\mathbf{x} \in \mathbb{Z}^m \colon \mathbf{A}\mathbf{x} = \mathbf{0} \pmod{M}\}$. Again, this is equivalent to solving a system of (homogeneous) linear equations.

3 *Basis:* Given a set of possibly dependent integer vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$, find a basis of the lattice they generate. This can be done in a variety of ways, for example using the Hermite Normal Form. (See Chapter 8.)

4 *Union:* Given two integer lattices $\mathcal{L}(\mathbf{B}_1)$ and $\mathcal{L}(\mathbf{B}_2)$, compute a basis for the smallest lattice containing both $\mathcal{L}(\mathbf{B}_1)$ and $\mathcal{L}(\mathbf{B}_2)$. This immediately reduces to the problem of computing a basis for the lattice generated by a sequence of possibly dependent vectors.

5 *Dual:* Given a lattice $\mathcal{L}(\mathbf{B})$, compute a basis for the dual of $\mathcal{L}(\mathbf{B})$, i.e., the set of all vectors $\mathbf{y}$ in span($\mathbf{B}$) such that $\langle \mathbf{x}, \mathbf{y} \rangle$ is an integer for every lattice vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$. It is easy to see that a basis for the dual is given by $\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}$.

6 *Intersection:* Given two integer lattices $\mathcal{L}(\mathbf{B}_1)$ and $\mathcal{L}(\mathbf{B}_2)$, compute a basis for the intersection $\mathcal{L}(\mathbf{B}_1) \cap \mathcal{L}(\mathbf{B}_2)$. It is easy to see that $\mathcal{L}(\mathbf{B}_1) \cap \mathcal{L}(\mathbf{B}_2)$ is always a lattice. This problem is easily solved using dual lattices.

7 *Equivalence:* Given two bases $\mathbf{B}_1$ and $\mathbf{B}_2$, check if they generate the same lattice $\mathcal{L}(\mathbf{B}_1) = \mathcal{L}(\mathbf{B}_2)$. This can be solved by checking if each basis vector belongs to the lattice generated by the other matrix, however, more efficient solutions exist.

8 *Cyclic:* Given a lattice $\mathcal{L}(\mathbf{C})$, check if $\mathcal{L}(\mathbf{C})$ is cyclic, i.e., if for every lattice vector $\mathbf{x} \in \mathcal{L}(\mathbf{C})$, all the vectors obtained by cyclically rotating the coordinates of $\mathbf{x}$ also belong to the lattice. This problem is easily solved by rotating the coordinates of basis matrix $\mathbf{C}$ by one position, and checking if the resulting basis is equivalent to the original one.

2.3 Hardness of approximation

In studying the computational complexity of approximating lattice problems, it is convenient to formulate them as *promise problems.* These are a generalization of decision problems well suited to study the hardness of approximation. A promise problem is a pair $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ of disjoint languages, i.e., $\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}} \subseteq \Sigma^*$ and $\Pi_{\mathrm{YES}} \cap \Pi_{\mathrm{NO}} = \emptyset$. An algorithm solves the promise problem $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ if on input an instance $I \in \Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$ it correctly decides whether $I \in \Pi_{\mathrm{YES}}$ or $I \in \Pi_{\mathrm{NO}}$. The behavior of the algorithm when $I \notin \Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$ (i.e., when I does not satisfy the promise) is not specified, i.e., on input an instance outside the promise, the algorithm is allowed to return any answer.

Decision problems are a special case of promise problems, where the set $\Pi_{\mathrm{NO}} = \Sigma^* \setminus \Pi_{\mathrm{YES}}$ is implicitly specified and the promise $I \in \Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$ is vacuously true. We now define the promise problems associated to the approximate SVP and CVP. These are denoted $\textsc{GapSVP}_\gamma$ and $\textsc{GapCVP}_\gamma$.

Definition 1.5 *The promise problem* $\textsc{GapSVP}_\gamma$*, where* γ *(the gap function) is a function of the rank, is defined as follows:*

- YES *instances are pairs* $(\mathbf{B}, r)$ *where* $\mathbf{B} \in \mathbb{Z}^{m\times n}$ *is a lattice basis and* $r \in \mathbb{Q}$ *a rational number such that* $\|\mathbf{Bz}\| \le r$ *for some* $\mathbf{z} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$.

- NO *instances are pairs* $(\mathbf{B}, r)$ *where* $\mathbf{B} \in \mathbb{Z}^{m\times n}$ *is a lattice basis and* $r \in \mathbb{Q}$ *is a rational such that* $\|\mathbf{Bz}\| > \gamma r$ *for all* $\mathbf{z} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$.

Definition 1.6 *The promise problem* $\textsc{GapCVP}_\gamma$*, where* γ *(the gap function) is a function of the rank, is defined as follows:*

- YES *instances are triples* $(\mathbf{B}, \mathbf{t}, r)$ *where* $\mathbf{B} \in \mathbb{Z}^{m\times n}$ *is a lattice basis,* $\mathbf{t} \in \mathbb{Z}^m$ *is a vector and* $r \in \mathbb{Q}$ *is a rational number such that* $\|\mathbf{Bz} - \mathbf{t}\| \le r$ *for some* $\mathbf{z} \in \mathbb{Z}^n$.

- NO *instances are triples* $(\mathbf{B}, \mathbf{t}, r)$ *where* $\mathbf{B} \in \mathbb{Z}^{m\times n}$ *is a lattice,* $\mathbf{t} \in \mathbb{Z}^m$ *is a vector and* $r \in \mathbb{Q}$ *is a rational number such that* $\|\mathbf{Bz} - \mathbf{t}\| > \gamma r$ *for all* $\mathbf{z} \in \mathbb{Z}^n$.

Notice that when the approximation factor equals $\gamma = 1$, the promise problems $\textsc{GapSVP}_\gamma$ and $\textsc{GapCVP}_\gamma$ are equivalent to the decision problems associated to exact SVP and CVP. Occasionally, with slight abuse of notation, we consider instances $(\mathbf{B}, r)$ (or $(\mathbf{B}, \mathbf{t}, r)$) where r is a real number, e.g., $r = \sqrt{2}$. This is seldom a problem in practice, because r can always be replaced by a suitable rational approximation. For example, in the ℓ_2 norm, if $\mathbf{B}$ is an integer lattice then r can be substituted with any rational in the interval $[r, \sqrt{r^2+1})$. Promise problems $\textsc{GapSVP}_\gamma$ and $\textsc{GapCVP}_\gamma$ capture the computational task of approximating SVP and CVP within a factor γ in the following sense. Assume algorithm $\mathcal{A}$ approximately solves SVP within a factor γ, i.e., on input a lattice Λ, it finds a vector $\mathbf{x} \in \Lambda$ such that $\|\mathbf{x}\| \le \gamma\lambda_1(\Lambda)$. Then $\mathcal{A}$ can be used to solve $\textsc{GapSVP}_\gamma$ as follows. On input $(\mathbf{B}, r)$, run algorithm $\mathcal{A}$ on lattice $\mathcal{L}(\mathbf{B})$ to obtain an estimate $r' = \|\mathbf{x}\| \in [\lambda_1, \gamma\lambda_1]$ of the shortest vector length. If $r' > \gamma r$ then $\lambda_1 > r$, i.e., $(\mathbf{B}, r)$ is not a YES instance. Since $(\mathbf{B}, r) \in \Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$, $(\mathbf{B}, r)$ must be a NO instance. Conversely, if $r' < \gamma r$ then $\lambda_1 < \gamma r$ and from the promise $(\mathbf{B}, r) \in \Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$ one deduces that $(\mathbf{B}, r)$ is a YES instance. On the other hand, assume one has a decision oracle $\mathcal{A}$ that solves GAPSVP$_\gamma$. (By definition, when the input does not satisfy the promise, the oracle can return any answer.) Let $u \in \mathbb{Z}$ be an upper bound to $\lambda(\mathbf{B})^2$ (for example, let u be the squared length of any of the basis vectors). Notice that $\mathcal{A}(\mathbf{B}, \sqrt{u})$ always returns YES, while $\mathcal{A}(\mathbf{B}, 0)$ always returns NO. Using binary search find an integer $r \in \{0, \ldots, u\}$ such that $\mathcal{A}(\mathbf{B}, \sqrt{r}) =$ YES and $\mathcal{A}(\mathbf{B}, \sqrt{r-1}) =$ NO. Then, $\lambda_1(\mathbf{B})$ must lie in the interval $[\sqrt{r}, \gamma \cdot \sqrt{r})$. A similar argument holds for the closest vector problem.

The class NP is easily extended to include promise problems. We say that a promise problem $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ is in NP if there exists a relation $R \subseteq \Sigma^* \times \Sigma^*$ such that $(x, y) \in R$ can be decided in time polynomial in $|x|$, and for every $x \in \Pi_{\mathrm{YES}}$ there exists a y such that $(x, y) \in R$, while for every $y \in \Pi_{\mathrm{NO}}$ there is no y such that $(x, y) \in R$. If the input x does not satisfies the promise, then R may or may not contain a pair (x, y). The complement of a promise problem $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ is the promise problem $(\Pi_{\mathrm{NO}}, \Pi_{\mathrm{YES}})$. For decision problems, this is the same as taking the set complement of a language in Σ^*. The class of decision problems whose complement is in NP is denoted coNP. Also coNP can be extended to include the complements of all NP promise problems.

Reductions between promise problems are defined in the obvious way. A function $f\colon \Sigma^* \to \Sigma^*$ is a reduction from $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ to $(\Pi'_{\mathrm{YES}}, \Pi'_{\mathrm{NO}})$ if it maps YES instances to YES instances and NO instances to NO instances, i.e., $f(\Pi_{\mathrm{YES}}) \subseteq \Pi'_{\mathrm{YES}}$ and $f(\Pi_{\mathrm{NO}}) \subseteq \Pi'_{\mathrm{NO}}$. Clearly any algorithm $\mathcal{A}$ to solve $(\Pi'_{\mathrm{YES}}, \Pi'_{\mathrm{NO}})$ can be used to solve $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ as follows: on input $I \in \Pi_{\mathrm{YES}} \cup \Pi_{\mathrm{NO}}$, run $\mathcal{A}$ on $f(I)$ and output the result. Notice that $f(I)$ always satisfy the promise $f(I) \in \Pi'_{\mathrm{YES}} \cup \Pi'_{\mathrm{NO}}$, and $f(I)$ is a YES instance if and only if I is a YES instance. A promise problem A is NP-hard if any NP language (or, more generally, any NP promise problem) B can be efficiently reduced to A. As usual, proving that a promise problem is NP-hard shows that no polynomial time solution for the problem exists unless P = NP. In the case of Cook reductions, the oracle Turing machine $\mathcal{A}$ to solve problem $(\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ should work given *any* oracle that solves $(\Pi'_{\mathrm{YES}}, \Pi'_{\mathrm{NO}})$. In particular, $\mathcal{A}$ should work no matter how queries outside the promise are answered by the oracle.

3. Notes

For a general introduction to computational models and complexity classes as used in this book, the reader is referred to (van Emde Boas, 1990) and (Johnson, 1990), or any undergraduate level textbook on the subject. Classical references about lattices are (Cassels, 1971) and (Gruber and Lekerkerker, 1987). Another very good reference is (Siegel, 1989). The proof of Minkowski's second theorem presented in Subsection 1.3 is an adaption to the Euclidean norm of the proof given in (Siegel, 1989) for arbitrary norms. None of the above references address algorithmic issues related to lattice problems, and lattices are studied from a purely mathematical point of view. For a brief introduction to the applications of lattices in various areas of mathematics and science the reader is referred to (Lagarias, 1995) and (Gritzmann and Wills, 1993), which also touch some complexity and algorithmic issues. A very good survey of algorithmic application of lattices is (Kannan, 1987a).