Lattice cryptography: from cryptanalysis to New Foundations

Woerden, W.P.J. van

Part IV

The Lattice Isomorphism Problem, Remarkable Lattices and Cryptography

CHAPTER 9

The Lattice Isomorphism Problem

This chapter gives an introduction to the Lattice Isomorphism Problem. It contains results from the joint work 'A canonical form for positive definite matrices', with Mathieu Dutour Sikirić, Anna Haensch, John Voight, published at ANTS 2020. In addition it shows direct applications of this work to ongoing joint work on Perfect form and C-type enumeration, with Mathieu Dutour Sikirić.

9.1 Introduction

When are two lattices the same? Two distinct bases $\mathbf{B}, \mathbf{B}' \in \mathcal{GL}_n(\mathbb{R})$ can generate exactly the same lattice $\mathcal{L}(\mathbf{B}) = \mathcal{L}(\mathbf{B}')$, and they do if and only if $\mathbf{B}' = \mathbf{B} \cdot \mathbf{U}$ for some unimodular matrix $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$. What about a lattice $\mathcal{L}$ and a rotation $\mathcal{L}'$ of the same lattice? These lattices are not (necessarily) the same as sets, but they do carry the same geometric information. We call such lattices isomorphic. More precisely we say that a lattice $\mathcal{L}$ and a lattice $\mathcal{L}'$ are *isomorphic* if

there is an orthonormal transformation $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$, i.e., an isometry, such that $\mathcal{L}' = \mathbf{O} \cdot \mathcal{L} = \{\mathbf{Ov} : \mathbf{v} \in \mathcal{L}\}$.

The study of isomorphisms between lattices is as old as the use of lattices, for example by Gauss, Lagrange, Minkowski and Voronoi from the 18th century to the beginning of the 20th century. Isomorphisms naturally arise in the search for interesting lattices[1], as a lattice is only really 'new' if it is not already isomorphic to a known one. Furthermore, isomorphisms between a lattice and itself, enable a study of the symmetries of a lattice.

Computing an isometry $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ between two isomorphic lattices, known as the Lattice Isomorphism Problem (LIP), or even deciding if two lattices are isomorphic, is seemingly a hard problem. Almost all algorithms require as a first step the enumeration of (many) short lattice vectors, which in itself is a hard problem. This includes the best proven algorithm, that runs in time $n^{O(n)}$ [HR14], as well as the best practical algorithms [PP85; PS97]. Moreover among the interesting lattices there are many with a high kissing number, and LIP seems even harder for those. For practical computations we are thus restricted to somewhat low dimensions, and this chapter must be considered in this context. For higher dimensions the hardness of LIP directly motivates its use in cryptography, and in Chapter 10 we show how to leverage LIP to construct a new type of lattice-based cryptography.

In this chapter we give an overview of the literature on LIP. First we show that this problem is more natural to state in the quadratic form setting. Next, we discuss the presence of natural invariants, that can be used to answer LIP in the negative, or guide the search for an isomorphism. We show how so-called characteristic sets play a fundamental role in solving LIP, and even in constructing a canonical representative inside an isomorphism class. We end with some interesting applications of this canonical representative, such as computationally solving the lattice packing problem by perfect form enumeration.

[1] Examples are lattices that are good packings, coverings or have a high kissing number, perfect lattices or those arising from number theory.

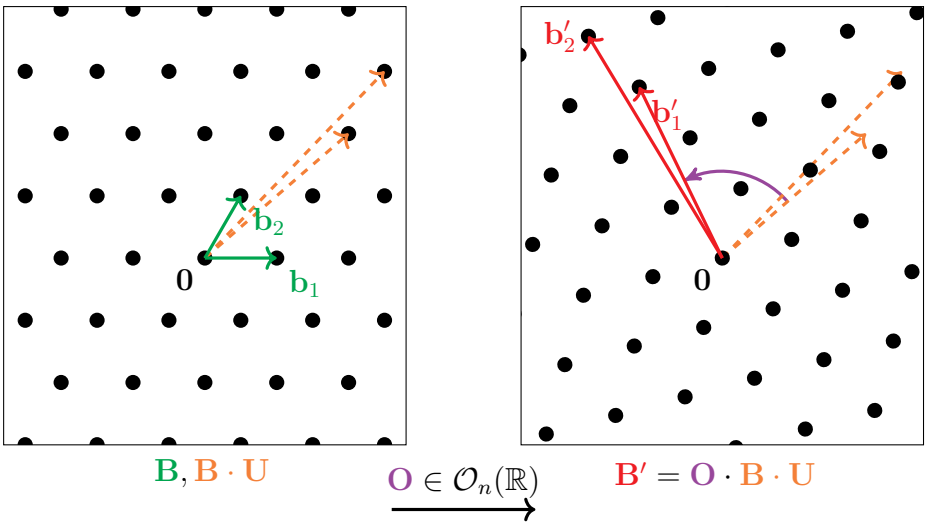

Figure 9.1: Illustration of two isomorphic lattices.

9.2 LIP & quadratic forms

Abstractly, the set of full rank, n-dimensional lattices can be thought of as the homogeneous space[2] $\mathcal{GL}_n(\mathbb{R})/\mathcal{GL}_n(\mathbb{Z})$: two bases $\mathbf{B}, \mathbf{B}' \in \mathcal{GL}_n(\mathbb{R})$ generate the same lattice if and only if there exists a unimodular matrix $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ such that $\mathbf{B}' = \mathbf{BU}$. Two lattices $\mathcal{L}, \mathcal{L}'$ are isomorphic if and only if there exists an orthonormal transformation $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ such that $\mathcal{L}' = \mathbf{O} \cdot \mathcal{L}$. The set of lattices up to isomorphism can thus be thought of as the double quotient

$$\mathcal{O}_n(\mathbb{R}) \backslash \mathcal{GL}_n(\mathbb{R}) / \mathcal{GL}_n(\mathbb{Z}).$$

Reconstructing equivalence in this double quotient is known as the Lattice Isomorphism Problem (LIP).

Definition 143 (LIP, lattice version). *Given bases $\mathbf{B}, \mathbf{B}' \subset \mathbb{R}^n$ of two isomorphic lattices, compute an orthonormal transformation $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$, and an unimodular transformation $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ such that $\mathbf{B}' = \mathbf{OBU}$.*

[2] This quotient should read as the quotient of a *set* by the action of group, and not a group quotient. Indeed $\mathcal{GL}_n(\mathbb{Z})$ is *not* a normal subgroup of $\mathcal{GL}_n(\mathbb{R})$ for $n > 1$.

If either $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ or $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ is known, recovering the other reduces to a matrix inversion. The presence of *both* the unknown orthonormal and the unknown unimodular transformation is what makes LIP seemingly a hard problem. In other words, reconstructing (or even testing) equivalence in either quotient $\mathcal{GL}_n(\mathbb{R})/\mathcal{GL}_n(\mathbb{Z})$ or $\mathcal{O}_n(\mathbb{R})\backslash\mathcal{GL}_n(\mathbb{R})$ is easy, doing so in the double quotient appears to be hard.

The real-valued coordinates of the basis and orthonormal transformation can be inconvenient and inefficient to work with. We can alleviate some of these concerns by moving to the (equivalent) quadratic form setting, where instead of a basis $\mathbf{B}$ we focus on the Gram matrix $\mathbf{Q} = \mathbf{B}^\top\mathbf{B} = (\langle \mathbf{b}_i, \mathbf{b}_j \rangle)_{i,j}$.

Quadratic forms and arithmetical equivalence

The idea of the Quadratic Form point of view on LIP is to consider the quotient in the opposite order than in the lattice point of view: first on the left by $\mathcal{O}_n(\mathbb{R})$ and then only on the right by $\mathcal{GL}_n(\mathbb{Z})$.

Definition 144 (Quadratic Form). *A quadratic form of rank n is a positive definite real*[3] *symmetric matrix $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$.*

Quadratic forms can be thought of as bases modulo rotation; they realize the quotient $\mathcal{O}_n(\mathbb{R})\backslash\mathcal{GL}_n(\mathbb{R})$. More precisely, consider the surjective Gram map $\gamma : \mathcal{GL}_n(\mathbb{R}) \to \mathcal{S}_n^{>0}(\mathbb{R})$ sending a lattice basis $\mathbf{B}$ to the quadratic form $\mathbf{Q} = \mathbf{B}^\top\mathbf{B}$. The preimages of $\gamma(\mathbf{B})$ are precisely the $\mathbf{OB}$ for $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$.

Lemma 145. *The Gram map $\gamma : \mathcal{GL}_n(\mathbb{R}) \to \mathcal{S}_n^{>0}(\mathbb{R})$ induces a bijection*

$$\begin{aligned} \mathcal{O}_n(\mathbb{R})\backslash\mathcal{GL}_n(\mathbb{R}) &\longleftrightarrow \mathcal{S}_n^{>0}(\mathbb{R}), \\ \mathcal{O}_n(\mathbb{R}) \cdot \mathbf{B} &\longmapsto \mathbf{B}^\top\mathbf{B}. \end{aligned}$$

Proof. Because $(\mathbf{OB})^\top(\mathbf{OB}) = \mathbf{B}^\top\mathbf{O}^\top\mathbf{OB} = \mathbf{B}^\top\mathbf{B}$ the map is well defined. The matrix $\mathbf{B}^\top\mathbf{B}$ is symmetric and positive definite as for any

[3] In constrast to the standard definition, our quadratic form is always real and positive definite. Furthermore we define the quadratic form as a symmetric matrix, and not by the associated degree two polynomial $f(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top\mathbf{Q}\mathbf{y}$.

nonzero $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ the vector $\mathbf{Bx}$ is nonzero, and thus $\mathbf{x}^\top(\mathbf{B}^\top\mathbf{B})\mathbf{x} = \|\mathbf{Bx}\|^2 > 0$. The map is injective because if for any $\mathbf{A}, \mathbf{B} \in \mathcal{GL}_n(\mathbb{R})$ we have $\mathbf{A}^\top\mathbf{A} = \mathbf{B}^\top\mathbf{B}$, then $(\mathbf{BA}^{-1})^\top(\mathbf{BA}^{-1}) = \mathbf{I}_n$ and thus $\mathbf{BA}^{-1} \in \mathcal{O}_n(\mathbb{R})$. The map is surjective because for every quadratic form $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$ the Cholesky decomposition gives a unique upper-triangular lattice basis $\mathbf{B_Q}$ with positive diagonal such that $\mathbf{Q} = \mathbf{B_Q^\top B_Q}$. □

For a lattice basis $\mathbf{B}$ the Gram matrix $\mathbf{Q} = \mathbf{B}^\top\mathbf{B}$ naturally gives a quadratic form. We can now translate the notions we have for lattices to those of quadratic forms. In the quadratic form setting lattice vectors $\mathbf{Bx} \in \mathbb{R}^n$ are represented by their integral basis coefficients $\mathbf{x} \in \mathbb{Z}^n$. The inner product with respect to a quadratic form is naturally given by $\langle \mathbf{x}, \mathbf{y} \rangle_\mathbf{Q} := \mathbf{x}^\top\mathbf{Qy}$, and the norm by $\|\mathbf{x}\|_\mathbf{Q}^2 := \mathbf{x}^\top\mathbf{Qx}$. Note that this perfectly coincides with the geometry between the original lattice vectors. We denote the $\mathbf{Q}$-ball of radius r by $\mathcal{B}_\mathbf{Q}(r) := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_\mathbf{Q} \leq r\}$. Translating the lattice definition, one gets the *first minimum*[4] $\lambda_1(\mathbf{Q})$ defined by

$$\lambda_1(\mathbf{Q}) := \min_{\mathbf{x} \in \mathbb{Z}^n \setminus \{0\}} \|\mathbf{x}\|_\mathbf{Q},$$

and more generally the i-th succesive minimum $\lambda_i(\mathbf{Q})$ defined as the smallest $r > 0$ such that $\mathbb{Z}^n \cap \mathcal{B}_\mathbf{Q}(r)$ spans a space of dimension at least i.

In this realization $\mathcal{S}_n^{>0}(\mathbb{R})$ of the quotient $\mathcal{O}_n(\mathbb{R})\backslash\mathcal{GL}_n(\mathbb{R})$, the (right) action of $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ is given by $\mathbf{Q} \mapsto \mathbf{U}^\top\mathbf{QU}$. We may now rephrase lattice isomorphisms in terms of quadratic forms, moving the real-valued orthonormal transform $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ out of the picture.

Definition 146 (Arithmetical Equivalence). *Two quadratic forms $\mathbf{Q}, \mathbf{Q}' \in \mathcal{S}_n^{>0}(\mathbb{R})$ are called equivalent, denoted $\mathbf{Q} \cong \mathbf{Q}'$, if there exists a unimodular $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ such that $\mathbf{Q}' = \mathbf{U}^\top\mathbf{QU}$.*

For two equivalent forms $\mathbf{Q}, \mathbf{Q}'$ we denote the set of all such unimodular transformations by $\mathrm{Isom}(\mathbf{Q}, \mathbf{Q}') := \{\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z}) : \mathbf{Q}' = \mathbf{U}^\top\mathbf{QU}\}$.

Lemma 147. *Two lattice bases $\mathbf{B}, \mathbf{B}'$ are isomorphic if and only if their Gram matrices are equivalent.*

[4] In the setting of quadratic forms the first minimum often refers to the squared norm, i.e., the value $\lambda_1(\mathbf{Q})^2$. We chose to stay in line with the lattice definition.

Proof. Let $\mathbf{B}, \mathbf{B}' \in \mathcal{GL}_n(\mathbb{R})$ be isomorphic bases, i.e., there exists an $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$, and $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ such that $\mathbf{B}' = \mathbf{OBU}$, then for $\mathbf{Q}' = \mathbf{B}'^\top \mathbf{B}'$ and $\mathbf{Q} := \mathbf{B}^\top \mathbf{B}$ we have:

$$\mathbf{Q}' := (\mathbf{B}')^\top \mathbf{B}' = \mathbf{U}^\top \mathbf{B}^\top \mathbf{O}^\top \mathbf{OBU} = \mathbf{U}^\top \mathbf{B}^\top \mathbf{BU} = \mathbf{U}^\top \mathbf{QU},$$

and thus $\mathbf{Q}'$ and $\mathbf{Q}$ are equivalent.

For the other direction, assume that $\mathbf{Q}' = \mathbf{U}^\top \mathbf{QU}$ for some $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$. This implies that $(\mathbf{B}')^\top \mathbf{B}' = (\mathbf{BU})^\top (\mathbf{BU})$, and thus by Lemma 145 there exists some $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ such that $\mathbf{B}' = \mathbf{OBU}$ and thus $\mathbf{B}$ and $\mathbf{B}'$ are isomorphic. □

We denote the equivalence class, or orbit, of a quadratic form $\mathbf{Q}$ by $[\mathbf{Q}] := \{\mathbf{U}^\top \mathbf{QU} : \mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})\}$. Many remarkable lattices attain a rational-valued Gram matrix $\mathbf{Q}$, removing the need for real-valued or approximate arithmetic. We will often restrict ourself to the set of integer-valued (positive-definite) quadratic forms denoted by $\mathcal{S}_n^{>0}(\mathbb{Z})$.

Remark 148 (Isometry). Generally the term *isometry* is used for a distance preserving bijective map between normed vector spaces, and *linear isometry* for norm preserving bijective linear maps. In this thesis we simply refer to *isometry* for any bijective inner product preserving map between (subsets of) inner product spaces. In particular for two equivalent forms $\mathbf{Q}, \mathbf{Q}' \in \mathcal{S}_n^{>0}(\mathbb{R})$, any unimodular $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ such that $\mathbf{Q}' = \mathbf{U}^\top \mathbf{QU}$ corresponds to an isometry $\mathbf{x} \mapsto \mathbf{U}^{-1}\mathbf{x}$ between $\mathbb{Z}^n \subset \mathbb{R}^n$ w.r.t. (the inner product given by) $\mathbf{Q}$ and $\mathbb{Z}^n \subset \mathbb{R}^n$ w.r.t. $\mathbf{Q}'$.

The Lattice Isomorphism Problem, quadratic form formulation

The Lattice Isomorphism Problem can now be restated. We start by properly defining the worst-case problems, in both a search and distinguishing variant.

Definition 149 (Worst-case search LIP: wc-sLIP$^{\mathbf{Q}}$). *For a quadratic form $\mathbf{Q} \in \mathcal{S}_n^{>0}$ the problem* wc-sLIP^{Q} *is, given any quadratic form $\mathbf{Q}' \in [\mathbf{Q}]$, to compute a unimodular $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ such that $\mathbf{Q}' = \mathbf{U}^\top \mathbf{QU}$.*

Note that the problem is equivalent to the original LIP problem as we can still extract an orthonormal transformation by computing $\mathbf{O} = \mathbf{B}'(\mathbf{BU})^{-1}$. We also consider a *distinguishing* variant of LIP, denoted wc-ΔLIP. It is not to be confused with the *decisional* version of LIP (which we will refer to as dLIP).[5]

Definition 150 (Worst-case distinguishing LIP: wc-$\Delta\mathrm{LIP}^{\mathbf{Q}_0,\mathbf{Q}_1}$). *For two quadratic forms $\mathbf{Q}_0, \mathbf{Q}_1 \in \mathcal{S}_n^{>0}$ the problem* wc-$\Delta\mathrm{LIP}^{\mathbf{Q}_0,\mathbf{Q}_1}$ *is, given any quadratic form $\mathbf{Q}' \in [\mathbf{Q}_b]$, where $b \in \{0,1\}$ is a uniform random bit, to recover b*[6].

The distinguishing problem wc-$\Delta\mathrm{LIP}^{\mathbf{Q}_0,\mathbf{Q}_1}$ is worst-case w.r.t. the choice of $\mathbf{Q}'$, but also the fact that the challenge $b \in \{0,1\}$ is sampled uniformly is worst-case: any additional bias would only make the problem easier for an attacker.

Automorphisms

Isomorphisms between a lattice and itself are called automorphisms. They represent the internal symmetry of the lattice. In the quadratic form setting, these are exactly the unimodular transformations that act invariantly, i.e., thet form the stabilizer group of $\mathbf{Q}$ under the action of $\mathcal{GL}_n(\mathbb{Z})$. This group is called the automorphism group of $\mathbf{Q}$.

Definition 151 (Automorphisms). *For a quadratic form $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$, the automorphism group* $\mathrm{Aut}(\mathbf{Q})$ *of $\mathbf{Q}$ is defined by*

$$\mathrm{Aut}(\mathbf{Q}) := \{\mathbf{V} \in \mathcal{GL}_n(\mathbb{Z}) : \mathbf{V}^\top \mathbf{Q} \mathbf{V} = \mathbf{Q}\}.$$

The automorphism group is finite, as will become clear in Section 9.5. For any $\mathbf{Q}' = \mathbf{U}^\top \mathbf{Q} \mathbf{U} \in [\mathbf{Q}]$ we have the natural stabilizer conjugacy $\mathrm{Aut}(\mathbf{Q}') = \mathbf{U}^{-1} \mathrm{Aut}(\mathbf{Q}) \mathbf{U}$. Additionally, we obtain a bijection between the right cosets $\mathrm{Aut}(\mathbf{Q}) \setminus \mathcal{GL}_n(\mathbb{Z})$ and the orbit $[\mathbf{Q}]$, and thus

[5] In $\mathrm{dLIP}^{\mathbf{Q}_0}$ one is given an arbitrary $\mathbf{Q}'$ and must decide whether $\mathbf{Q}'$ belongs to $[\mathbf{Q}_0]$. The distinguishing version is potentially easier in that $\mathbf{Q}'$ is promised to belong to either $[\mathbf{Q}_0]$ or $[\mathbf{Q}_1]$ for some known fixed $[\mathbf{Q}_1]$.

[6] The distinguishing problem should be interpreted as a game: a challenger samples $b \in \{0,1\}$ uniformly and responds with any $\mathbf{Q}' \in [\mathbf{Q}_b]$, the prover should respond with a bit b' and wins if $b = b'$. The prover solves the problem if it can win with non-negligible advantage (over the trivial win probability of 0.5).

for any isomorphism $\mathbf{U} \in \mathrm{Isom}(\mathbf{Q}, \mathbf{Q}')$, the full set of isomorphisms is given by $\mathrm{Isom}(\mathbf{Q}, \mathbf{Q}') = \mathrm{Aut}(\mathbf{Q}) \cdot \mathbf{U}$.

9.3 Invariants

Equivalent quadratic forms $\mathbf{Q}, \mathbf{Q}' := \mathbf{U}^\top \mathbf{Q} \mathbf{U}$ (for some $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$) share many common properties, and these invariants can be used to decide that two quadratic forms cannot be equivalent, or can guide the search for an isomorphism. We start with some arithmetic invariants that are efficiently computable, and then discuss some geometric invariants which are only efficiently computable in low dimensions. We restrict ourself to integral quadratic forms $\mathbf{Q}, \mathbf{Q}' \in \mathcal{S}_n^{>0}(\mathbb{Z})$, as those are common in practice and allow for the most invariants.

9.3.1 Arithmetic invariants

Firstly we have $\det(\mathbf{U}) = \pm 1$, and thus for two equivalent quadratic forms we have

$$\det(\mathbf{Q}') = \det(\mathbf{U}^\top) \det(\mathbf{Q}) \det(\mathbf{U}) = \det(\mathbf{Q}).$$

Secondly because $\mathbf{U}$ and $\mathbf{U}^{-1}$ are both integral, the quantity $\gcd(\mathbf{Q}) := \gcd\{\mathbf{Q}_{ij} : 1 \le i, j \le n\}$ also gives an invariant.

A third and less obvious invariant is the parity of the quadratic form. The notion is standard for unimodular lattices: it is called even if all norms are even, and odd otherwise. More generally, writing $\|\mathbf{x}\|_{\mathbf{Q}}^2 = \sum_i \mathbf{Q}_{ii} x_i^2 + 2 \sum_{i<j} x_j \mathbf{Q}_{ij} x_i$ one gets that $\gcd\{\|\mathbf{x}\|_{\mathbf{Q}}^2 : x \in \mathbb{Z}^n\} \in \{1, 2\} \cdot \gcd(\mathbf{Q})$. We call this factor $\mathrm{par}(\mathbf{Q}) \in \{1, 2\}$ the parity of $\mathbf{Q}$. It is also efficiently computable by noting that $\mathrm{par}(\mathbf{Q}) = \gcd(\{\mathbf{Q}_{ii} : 1 \le i \le n\} \cup \{2 \gcd(\mathbf{Q})\}) / \gcd(\mathbf{Q})$.

These invariants are all a bit ad hoc, and possibly incomplete. All these invariants (and more) are captured by the following.

Weaker equivalence (genus)

The study of integral equivalence of quadratic forms is classically approached via weaker notions, namely, equivalence over larger rings [CS13, Ch. 15, Sec. 4] $R \supset \mathbb{Z}$. In particular we consider $R =$

$\mathbb{R}, \mathbb{Q}, \mathbb{Q}_p, \mathbb{Z}_p$, where $\mathbb{Q}_p$ denotes the rational and $\mathbb{Z}_p$ the integer p-adic numbers for a prime p. For any ring R we denote the equivalence class of a quadratic form $\mathbf{Q}$ by $[\mathbf{Q}]_R := \{\mathbf{U}^\top \mathbf{Q}\mathbf{U} : \mathbf{U} \in \mathcal{GL}_n(R)\}$. These equivalences are coarser than integral equivalence, $[\mathbf{Q}] = [\mathbf{Q}'] \Rightarrow [\mathbf{Q}]_R = [\mathbf{Q}']_R$. The R-equivalence class is thus an invariant under integral equivalence, which, in contrast to $R = \mathbb{Z}$, might be efficiently computable.

Over the reals one can always diagonalize a quadratic form $\mathbf{D} := \mathbf{U}^\top \mathbf{Q}\mathbf{U}$ with $\mathbf{U} \in \mathcal{GL}_n(\mathbb{R})$ such that $\mathbf{D}_{ii} \in \{-1, 0, 1\}$. Let $n_j = \#\{i : \mathbf{D}_{ii} = j\}$ for $j\{-1, 0, 1\}$ denote the number of occurrences of j on the diagonal. Then (n_1, n_0, n_{-1}), also called the *signature*, uniquely represents the $\mathbb{R}$-equivalence class. A positive definite quadratic form of rank n always has a signature of $(n, 0, 0)$, and thus all quadratic forms of fixed rank that we consider fall in the same class.

We now turn our focus to the p-adic integral equivalence for all primes p. For (positive definite) quadratic forms this equivalence over all primes p is nicely summarized as the genus.

Definition 152 (Genus). *The genus* $\text{genus}(\mathbf{Q})$ *of a (positive definite) quadratic form* $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ *is defined as the collection of* p*-adic integral equivalence classes* $([\mathbf{Q}]_{\mathbb{Z}_p})_p$ *for all primes* p*.*

Two quadratic forms $\mathbf{Q}, \mathbf{Q}' \in \mathcal{S}_n^{>0}(\mathbb{Z})$ have the same genus if and only if they are p-adic integral equivalent for all primes p. In general, the genus is defined to also cover $\mathbb{R}$-equivalence, but for positive definite forms that would be redundant. The genus turns out to cover all arithmetic invariants mentioned so far, while, as we see later, still being efficient to compute.

Theorem 153 (Hasse-Minkowski). *If two (positive definite) quadratic forms are* p*-adic equivalent for all primes* p*, then they are equivalent over the rationals, and their invariants* $\det, \gcd$*, and* par *match.*

Proof. Note that $\mathbb{Z}_p \subset \mathbb{Q}_p$, and that all positive definite quadratic forms of the same rank are $\mathbb{R}$-equivalent. So the first statement follows from the Hasse-Minkowski theorem. The second part of the proof follows from Remark 155. □

We shortly discuss how p-adic integral equivalence can be classified efficiently by a complete system of invariants by Conway [CS13, Ch.15]. Let $p \geq 2$ be any prime. By appropriate (near) diagonalizing any form $\mathbf{Q}$ can be decomposed over the p-adic integers as a block diagonal matrix

$$\mathbf{Q} = \mathbf{Q}_1 \oplus p\mathbf{Q}_p \oplus p^2\mathbf{Q}_{p^2} \oplus \cdots \oplus q\mathbf{Q}_q \oplus \cdots , \tag{9.1}$$

in which each $\mathbf{Q}_q$ is a p-adic integral form whose determinant is coprime to p. For an odd prime $p \neq 2$ the (finite) set of values of q occurring in eq. (9.1), together with the dimensions $n_q := \dim(\mathbf{Q}_q)$ and signs

$$\epsilon_q := \left(\frac{\det(\mathbf{Q}_q)}{p}\right),$$

form a complete set of invariants for the p-adic integral equivalence of $\mathbf{Q}$.

Theorem 154 ([CS13, Ch. 15, Theorem 9]). *For any odd prime $p \neq 2$, two quadratic forms $\mathbf{Q}, \mathbf{Q}'$ are equivalent over the p-adic integers if and only if they have the same invariants n_q, ϵ_q for each power q of p.*

For any odd prime $p \nmid \det(\mathbf{Q})$ we have $n_q = n$, and ϵ_q is fully determined by the determinant $\det(\mathbf{Q})$. As such primes can be ignored, we only have to consider the finite number of odd primes $p \mid \det(\mathbf{Q})$. For such odd primes p there are at most n powers $q_1, \ldots, q_k \leq \det(\mathbf{Q})$ of p with a non-trivial block dimension $n_{q_i} > 0$, and thus $\{(q, n_{q_i}, \epsilon_{q_i})\}_{i=1,\ldots,k}$ determines canonically the p-adic integral equivalence. So the genus (except for $p = 2$) can be described in a finite and canonical way.

For $p = 2$ there are some additional complexities. For example for $\mathbf{Q}_q$ with q a power of 2, the existence (or not) of an odd entry on the diagonal gives an additional invariant, which is related to the earlier defined parity of a quadratic form. If there is an odd entry on the diagonal, the trace t_q mod 8 of $\mathbf{Q}_q$, better known as the oddity, is needed to completely specify the 2-adic equivalence. These oddities are not an invariant, but together with the other invariants they can efficiently be turned into a canonical description (see [CS13, Ch. 15, Sec. 7].

Maybe unsurprisingly, the genus covers all the previously defined invariants.

Remark 155 (Genus invariants). The genus of a quadratic form $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ completely determines the det, gcd and par of $\mathbf{Q}$.

Proof. For any p and the p-adic decomposition eq. (9.1) the maximal power of p dividing the determinant $\det(\mathbf{Q})$ is given by $\prod_{q=p^i} q^{n_q}$. The maximal power of p dividing $\gcd(\mathbf{Q})$ is given by the minimal $q = p^i$ such that $n_q \geq 1$. The parity is fully determined by the decomposition at $p = 2$. Namely, let $q = 2^i$ be the minimal power of 2 such that $n_q \geq 1$, then $\mathrm{par}(\mathbf{Q}) = 1$ if and only if there is an odd value on the diagonal of a matrix representing $\mathbf{Q}_q$. □

The hull

In the literature for linear code equivalence, a relevant notion is that of the efficiently computable hull $C \cap C^\perp$ of a code $C \subset \mathbb{F}_q^n$. Properties such as the rank of the hull are invariant under equivalence, and a small rank even allows to efficiently find the isometry [Sen00].

For a lattice $\mathcal{L}$ and its dual $\mathcal{L}^*$ we could define the hull as $\mathcal{L} \cap \mathcal{L}^*$. For integral lattices (i.e., if the associated quadratic form is integral) the hull does not present us with new information, since we always have $\mathcal{L} \subset \mathcal{L}^*$ and thus $\mathcal{L} \cap \mathcal{L}^* = \mathcal{L}$. We could generalize the definition to consider $\mathcal{L} \cap (k \cdot \mathcal{L}^*)$ for rational $k \in \mathbb{Q}_{\neq 0}$. However, even for such a generalized construction the genus of the resulting lattice is completely determined by the genus of the original lattice $\mathcal{L}$ [DG22]. To conclude, the (generalized) hull does not appear to give us any new arithmetic invariant.

One possibility is that the generalized hull could have certain geometric properties that could be exploited (more so than the original lattice or its dual). This is a question that deserves further investigation, but is beyond the scope of this work.

9.3.2 Geometric invariants

The defining property of a unimodular transformation $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ is that it gives a bijection $\mathbb{Z}^n \to \mathbb{Z}^n$ by $\mathbf{x} \mapsto \mathbf{U}\mathbf{x}$ (or $\mathbf{x} \mapsto \mathbf{U}^{-1}\mathbf{x}$). With respect to the quadratic forms $\mathbf{Q}, \mathbf{Q}' := \mathbf{U}^\top \mathbf{Q} \mathbf{U}$ this even gives an isometry (from $\mathbf{Q}'$ to $\mathbf{Q}$) as

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{Q}'} = \mathbf{x}^\top \mathbf{Q}' \mathbf{y} = \mathbf{x}^\top \mathbf{U}^\top \mathbf{Q} \mathbf{U} \mathbf{y} = \langle \mathbf{U}\mathbf{x}, \mathbf{U}\mathbf{y} \rangle_{\mathbf{Q}} \text{ for } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

This isometry results in several natural geometric invariants related to the norms and inner products of integral vectors. We have already seen some, namely the first minimum $\lambda_1(\mathbf{Q})$ and the i-th successive minimum $\lambda_i(\mathbf{Q})$. More geometric invariants can be defined, such as the kissing number $\kappa(\mathbf{Q}) = |\operatorname{Min}(\mathbf{Q})|$ where

$$\operatorname{Min}(\mathbf{Q}) := \{\mathbf{x} \in \mathbb{Z}^n : \|\mathbf{x}\|_{\mathbf{Q}} = \lambda_1(\mathbf{Q})\},$$

and more generally the (formal) Theta-series $\Theta_{\mathbf{Q}}(q) = \sum_{\ell \geq 0} N_\ell q^\ell$ associated to $\mathbf{Q}$, where $N_\ell = |\{\mathbf{x} \in \mathbb{Z}^n : \|\mathbf{x}\|_{\mathbf{Q}}^2 = \ell\}|$. Going back to the inner products one could also consider the (multi-)set of all pairwise inner products $\{\langle \mathbf{x}, \mathbf{y} \rangle : \mathbf{x}, \mathbf{y} \in S\}$, where e.g., $S = \operatorname{Min}(\mathbf{Q})$ or $S = \{\mathbf{x} \in \mathbb{Z}^n : \|\mathbf{x}\|_{\mathbf{Q}}^2 = \ell\}$. One could also consider pairwise inner products between two of such sets that are distinct.

Two equivalent forms also share symmetries $\operatorname{Aut}(\mathbf{Q}') \cong \operatorname{Aut}(\mathbf{Q})$, which for example implies that $|\operatorname{Aut}(\mathbf{Q}')| = |\operatorname{Aut}(\mathbf{Q})|$.

The computation of all these geometric invariants appears to involve finding or even enumerating short vectors, or computing automorphisms; in particular they seem hard to compute in general.

9.4 Characteristic sets

To solve the Lattice Isomorphism Problem we need to recover a right unimodular transformation $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ such that $\mathbf{Q}' = \mathbf{U}^\top \mathbf{Q} \mathbf{U}$. Since the group $\mathcal{GL}_n(\mathbb{Z})$ of such transformations is infinite, it is a priori not clear how even a brute-force approach would work. Instead, we follow the framework introduced by Plesken and Souvignier [PS97] based on so-called *characteristic sets*.

Definition 156 (Characteristic Set). *A characteristic vector set function $\mathcal{V}$ is a map that assigns to every $n \geq 1$ and quadratic form $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$, a finite subset of vectors $\mathcal{V}(\mathbf{Q}) \subset \mathbb{Z}^n$ such that*

(i) $\mathcal{V}(\mathbf{Q})$ generates $\mathbb{Z}^n$ (as a $\mathbb{Z}$-module); and

(ii) for all $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$, we have $\mathbf{U}^{-1}\mathcal{V}(\mathbf{Q}) = \mathcal{V}(\mathbf{U}^\top \mathbf{Q} \mathbf{U})$.

The transformation property (ii) can be interpreted as a canonicity property under equivalence of quadratic forms, or stated differently,

that the map $\mathcal{V}$ is equivariant (for each $n \geq 1$). For example the minimal vectors function $\mathbf{Q} \mapsto \mathrm{Min}(\mathbf{Q})$ satisfies this property; minimal vectors are mapped to minimal vectors by any isometry. Characteristic sets allow us to limit the search space to a finite one; as, by property (ii), each solution of LIP between two quadratic forms $\mathbf{Q}, \mathbf{Q}'$ induces an isometry between their characteristic sets $\mathcal{V}(\mathbf{Q}), \mathcal{V}(\mathbf{Q}')$. Property (i) is necessary to obtain the correspondence in the other direction.

Lemma 157 (Isometries). *Let $\mathcal{V}$ be a characteristic function, and let $\mathbf{Q}, \mathbf{Q}' \in \mathcal{S}_n^{>0}(\mathbb{R})$ be quadratic forms. There is a bijection*

$$\begin{aligned} \mathrm{Isom}(\mathbf{Q}, \mathbf{Q}') &\leftrightarrow \mathrm{Isom}(\mathcal{V}(\mathbf{Q}), \mathcal{V}(\mathbf{Q}')), \\ \mathbf{U} &\mapsto (\psi_{\mathbf{U}} : \mathbf{x} \mapsto \mathbf{U}^{-1}\mathbf{x}), \end{aligned}$$

where $\mathrm{Isom}(\mathcal{V}(\mathbf{Q}), \mathcal{V}(\mathbf{Q}'))$ *is the set of isometries between* $\mathcal{V}(\mathbf{Q})$ *w.r.t.* $\mathbf{Q}$*, and* $\mathcal{V}(\mathbf{Q}')$ *w.r.t.* $\mathbf{Q}'$*. If* $\mathbf{Q} = \mathbf{Q}'$*, then* $\mathrm{Isom}(\mathbf{Q}, \mathbf{Q}') = \mathrm{Aut}(\mathbf{Q})$ *and* $\mathrm{Isom}(\mathcal{V}(\mathbf{Q}), \mathcal{V}(\mathbf{Q}'))$ *are groups under function composition and the bijection is an isomorphism.*

Proof. Let $\mathbf{U} \in \mathrm{Isom}(\mathbf{Q}, \mathbf{Q}')$ be such that $\mathbf{Q}' = \mathbf{U}^\top \mathbf{Q} \mathbf{U}$. Then by definition of the characteristic set we have $\mathcal{V}(\mathbf{Q}') = \mathbf{U}^{-1}\mathcal{V}(\mathbf{Q})$, and thus $\psi_{\mathbf{U}}$ induces a bijection from $\mathcal{V}(\mathbf{Q})$ to $\mathcal{V}(\mathbf{Q}')$. Furthermore for any $\mathbf{x}, \mathbf{y} \in \mathcal{V}(\mathbf{Q})$ we have $\langle \psi_{\mathbf{U}}(\mathbf{x}), \psi_{\mathbf{U}}(\mathbf{y}) \rangle_{\mathbf{Q}'} = \langle \mathbf{U}^{-1}\mathbf{x}, \mathbf{U}^{-1}\mathbf{y} \rangle_{\mathbf{U}^\top \mathbf{Q} \mathbf{U}} = \langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{Q}}$, and thus $\psi_{\mathbf{U}}$ is actually an isometry.

For the injectivity let $\mathbf{U}, \mathbf{V} \in \mathrm{Isom}(\mathbf{Q}, \mathbf{Q}')$ be such that $\psi_{\mathbf{U}} = \psi_{\mathbf{V}}$. This implies that $\mathbf{U}^{-1}\mathbf{x} = \mathbf{V}^{-1}\mathbf{x}$ for all $\mathbf{x} \in \mathcal{V}(\mathbf{Q})$, and because $\mathcal{V}(\mathbf{Q})$ is of full rank we have $\mathbf{U}^{-1} = \mathbf{V}^{-1}$, and thus $\mathbf{U} = \mathbf{V}$.

For the surjectivity let $\psi : \mathcal{V}(\mathbf{Q}) \to \mathcal{V}(\mathbf{Q}')$ be an isometry. Both characteristic sets generate $\mathbb{Z}^n$ as a $\mathbb{Z}$-module, and thus in particular we can find n $\mathbb{R}$-linear independent vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{V}(\mathbf{Q})$. Because ψ is an isometry, the vectors $\psi(\mathbf{x}_1), \ldots, \psi(\mathbf{x}_n)$ must also be linearly independent, and thus we obtain a *unique* linear map $\mathbf{U} \in \mathcal{GL}_n(\mathbb{R})$ such that $\psi(\mathbf{x}_i) = \psi_{\mathbf{U}}(\mathbf{x}_i)$ for all $i = 1, \ldots, n$. Because ψ is an isometry we have for any $\mathbf{y} \in \mathcal{V}(\mathbf{Q})$ the n independent linear equations $\langle \psi(\mathbf{y}), \psi_{\mathbf{U}}(\mathbf{x}_i) \rangle_{\mathbf{Q}'} = \langle \mathbf{y}, \mathbf{x}_i \rangle_{\mathbf{Q}}$ for $i = 1, \ldots, n$, with the unique solution $\psi(\mathbf{y}) = \psi_{\mathbf{U}}(\mathbf{y})$. So $\psi = \psi_{\mathbf{U}}$ is in fact a linear isometry represented by $\mathbf{U}$, and because both characteristic sets generate $\mathbb{Z}^n$ as a $\mathbb{Z}$-module, we have $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ by definition. From the isometry we obtain that $\mathbf{e}_i^\top \mathbf{U}^\top \mathbf{Q} \mathbf{U} \mathbf{e}_j = \mathbf{e}_i^\top \mathbf{Q}' \mathbf{e}_j$ for all pairs of standard unit

vectors $\mathbf{e}_i, \mathbf{e}_j$ with $i, j = 1, \ldots, n$, which in particular gives us that $\mathbf{Q}' = \mathbf{U}^\top \mathbf{Q} \mathbf{U}$. So $\mathbf{U} \in \mathrm{Isom}(\mathbf{Q}, \mathbf{Q}')$, and it has image $\psi = \psi_{\mathbf{U}}$.

If $\mathbf{Q} = \mathbf{Q}'$, then both sides are groups under composition. For $\mathbf{U}, \mathbf{V} \in \mathrm{Isom}(\mathbf{Q}, \mathbf{Q}) = \mathrm{Aut}(\mathbf{Q})$ their composition as automorphisms $\mathbf{V} \circ \mathbf{U}$ is represented by $\mathbf{UV} \in \mathrm{Aut}(\mathbf{Q})$. Then for $\mathbf{x} \in \mathcal{V}(\mathbf{Q})$ we have

$$\psi_{\mathbf{V} \circ \mathbf{U}}(\mathbf{x}) = \psi_{\mathbf{UV}}(\mathbf{x}) = (\mathbf{UV})^{-1}\mathbf{x} = \mathbf{V}^{-1}\mathbf{U}^{-1}\mathbf{x} = (\psi_{\mathbf{V}} \circ \psi_{\mathbf{U}})(\mathbf{x}),$$

and thus we have an isomorphism. □

We will now discuss some examples of characteristic vector set functions.

9.4.1 A set based on short vectors

As mentioned before the map $\mathbf{Q} \mapsto \mathrm{Min}(\mathbf{Q})$ satisfies the desired transformation property of a characteristic vector set function. However, two problems remain for using $\mathrm{Min}(A)$ as a characteristic vector set:

PB1. If $n \geq 2$, then $\mathrm{span}(\mathrm{Min}(A))$ may not have rank n.

PB2. If $n \geq 5$, then $\mathrm{span}(\mathrm{Min}(A))$ may have rank n, but may not generate $\mathbb{Z}^n$ as a $\mathbb{Z}$-module.

Thus we have to consider larger sets of short lattice vectors. For $\lambda > 0$, let

$$\mathrm{Min}(\mathbf{Q}, \lambda) := \left\{ \mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\} : \|\mathbf{x}\|_{\mathbf{Q}} \leq \lambda \right\}, \tag{9.2}$$

be the set of all nonzero integer vectors up to length λ w.r.t. $\mathbf{Q}$. In particular we have $\mathrm{Min}(\mathbf{Q}) = \mathrm{Min}(\mathbf{Q}, \lambda_1(\mathbf{Q}))$.

Now we need to pick $\lambda > 0$ large enough such that $\mathrm{Min}(\mathbf{Q}, \lambda)$ generates $\mathbb{Z}^n$. The implementations `AUTO/ISOM` by Plesken and Souvignier [PS97], pick λ^2 equal to $\mathrm{maxdiag}(\mathbf{Q}) := \max \mathbf{Q}_{ii}$, to compute automorphisms. The set $\mathrm{Min}(\mathbf{Q}, \sqrt{\mathrm{maxdiag}(\mathbf{Q})})$ contains the standard unit vectors, and thus generates $\mathbb{Z}^n$. One could first use lattice reduction techniques to limit the size of $\mathrm{maxdiag}(\mathbf{Q})$, to prevent large sets. Such a set is enough to compute automorphisms, but can be problematic for equivalence, e.g., two forms $\mathbf{Q}, \mathbf{Q}'$ can be equivalent but satisfy $\mathrm{maxdiag}(\mathbf{Q}) \neq \mathrm{maxdiag}(\mathbf{Q}')$. For a particular LIP instance this can be solved by picking the bound $\lambda^2 = \max\{\mathrm{maxdiag}(\mathbf{Q}), \mathrm{maxdiag}(\mathbf{Q}')\}$. However, such a choice is still not canonical (something we will care

about in Section 9.6), and would in particular break the transformation property. Additionally this may lead to much larger sets than necessary.

To prevent these problems we can use a more reliable vector set, namely by picking the smallest bound λ, such that $\mathrm{Min}(\mathbf{Q}, \lambda)$ still spans $\mathbb{Z}^n$.

$$\begin{aligned} \mathcal{V}_{\mathrm{ms}}(\mathbf{Q}) &:= \mathrm{Min}(\mathbf{Q}, \lambda_{\min}), \text{ where} \\ \lambda_{\min} &:= \min\{\lambda > 0 : \mathrm{span}_{\mathbb{Z}}(\mathrm{Min}(\mathbf{Q}, \lambda)) = \mathbb{Z}^n\}. \end{aligned} \tag{9.3}$$

Lemma 158. *The map $\mathbf{Q} \mapsto \mathcal{V}_{\mathrm{ms}}(\mathbf{Q})$ is a characteristic vector set function.*

This characteristic set function often works great in practice[7], but in general we do not have any bound on the size of $|\mathcal{V}_{\mathrm{ms}}(\mathbf{Q})|$, as the following example shows.

Example 159. The quadratic form $\mathbf{Q}_\lambda = \begin{pmatrix} 1 & 0 \\ 0 & \lambda^2 \end{pmatrix}$ for $\lambda \geq 1$ gives

$$\mathcal{V}_{\mathrm{ms}}(\mathbf{Q}_\lambda) = \{\pm e_2\} \cup \{\pm e_1, \pm 2e_1, \ldots, \pm \lfloor \lambda \rfloor e_1\}.$$

One way to solve the particular example above is by also removing all non-primitive vectors from the set, but this is not enough to solve the general problem of obtaining vector sets of unbounded size (under a fixed dimension). To overcome this problem, we introduce a characteristic vector set, although somewhat impractical, that does come with bounds on the number of vectors, only depending on the dimension n.

9.4.2 A set based on Voronoi-relevant vectors

A well-known geometric shape associated to lattices, extensively discussed in Chapter 5, is the *Voronoi cell*. The Voronoi cell is the set of all points closer to 0 with respect to $\mathbf{Q}$ than to any other integer point. For a form $\mathbf{Q}$, the (closed) Voronoi cell is the intersection of half-spaces

$$\mathrm{Vor}(\mathbf{Q}) := \bigcap_{\mathbf{x} \in \mathbb{Z}^n \setminus \{0\}} H_{\mathbf{Q},\mathbf{x}}, \tag{9.4}$$

[7] Practical for computational algebra purposes in low dimensions of say $n \leq 30$. Not for cryptographic lattices of large dimension.

with $H_{\mathbf{Q},\mathbf{x}} := \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{y}\|_{\mathbf{Q}} \leq \|\mathbf{y} - \mathbf{x}\|_{\mathbf{Q}}\}$. However, almost all vectors in this intersection are superfluous, and we only consider the finite set of *Voronoi-relevant vectors* $\mathcal{V}_{\mathrm{vor}}(\mathbf{Q})$, i.e., the (unique) minimal set of vectors such that

$$\mathrm{Vor}(\mathbf{Q}) = \bigcap_{\mathbf{x} \in \mathcal{V}_{\mathrm{vor}}(\mathbf{Q})} H_{\mathbf{Q},\mathbf{x}}. \tag{9.5}$$

Each Voronoi-relevant vector corresponds to exactly one facet of the Voronoi cell. Using, an algorithm by Micciancio and Voulgaris [MV13], we can compute the set of Voronoi-relevant vectors in time $2^{2n+o(n)}$.

Lemma 160. *The following statements hold*

(i) The map $\mathbf{Q} \mapsto \mathcal{V}_{\mathrm{vor}}(\mathbf{Q})$ *is a characteristic vector set function;*

(ii) We have $\#\mathcal{V}_{\mathrm{vor}}(\mathbf{Q}) \leq 2 \cdot (2^n - 1)$.

Proof. Property (ii) of a characteristic vector set for $\mathcal{V}_{\mathrm{vor}}$ follows from the geometric definition, fully independent of the basis. For property (i), note that for any nonzero $\mathbf{x} \in \mathbb{Z}^n$, we have $\mathbf{x} \notin \mathrm{Vor}(\mathbf{Q})$, and thus by definition there is a Voronoi-relevant vector $\mathbf{v} \in \mathcal{V}_{\mathrm{vor}}(\mathbf{Q})$ such that $\mathbf{x} - \mathbf{v}$ lies strictly closer to 0 with respect to $\mathbf{Q}$. Repeating this (a finite number of time by a packing argument) we eventually end up at 0 and thus $\mathbf{x}$ is the sum of Voronoi-relevant vectors. The second statement was already proven by Minkowski in 1897 [Min97]. The idea is that each Voronoi-relevant vector $\mathbf{v}$ has precisely two closests vectors in $2\mathcal{L}$, namely $\mathbf{0}$ and $2\mathbf{v}$. This implies that each $\pm\mathbf{v}$ (up to sign) represents a distinct and nonzero coset modulo $2\mathcal{L}$, so there can be at most $2 \cdot (|\mathcal{L}/(2\mathcal{L})| - 1)$ Voronoi-relevant vectors. □

Although this characteristic vector set has great theoretical bounds, we refrain from using it in practice: most lattices actually attain the worst-case $2 \cdot (2^n - 1)$ Voronoi bound [Vor08; Vor09], whereas constructions based on short vectors often beat the theoretical worst-case bounds and give much smaller vector sets in practice.

9.5 Solving LIP

We discuss several algorithms to solve the Lattice Isomorphism Problem. We start with some practical approaches based on the earlier

introduced characteristic vector sets, and then discuss some theoretical results.

9.5.1 Characteristic set approach

Recall from Lemma 157 that solving LIP between two quadratic forms $\mathbf{Q}, \mathbf{Q}'$ is equivalent to finding an isometry between the finite sets $\mathcal{V}(\mathbf{Q})$ and $\mathcal{V}(\mathbf{Q}')$, for any characteristic set function $\mathbf{Q} \mapsto \mathcal{V}(\mathbf{Q})$.

A generic approach, that we formalize in [DHVW20], of finding such isometries is to construct two complete weighted graphs for $\mathbf{Q}$ and $\mathbf{Q}'$ respectively, where the nodes are the elements from the characteristic vector set, and each edge $(\mathbf{x}, \mathbf{y})$ gets weight $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{Q}}$ or $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{Q}'}$ respectively.

Definition 161. *Let $\mathcal{V}$ be a characteristic vector set function, and let $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$ be a quadratic form. We define the complete weighted graph $G_{\mathbf{Q},\mathcal{V}} = (V, w)$ as the graph with nodes $V := \mathcal{V}(\mathbf{Q})$ and weight function*

$$w(\mathbf{x}, \mathbf{y}) := \langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{Q}} \text{ for all } \mathbf{x}, \mathbf{y} \in \mathcal{V}(\mathbf{Q}).$$

Using these graphs turns the problem of quadratic form equivalence into one of graph isomorphism. Recall that two complete weighted graphs $G = (V, w)$ and $G' = (V', w')$ are isomorphic $G \cong G'$ if there exists a bijection $\psi : V \to V'$, which keeps the edge weights intact, i.e., $w'(\psi(x), \psi(y)) = w(x, y)$ for each edge $(x, y) \in V^2$. We denote the set of all such bijections by $\mathrm{Isom}(G, G')$.

Lemma 162. *Let $\mathcal{V}$ be a characteristic vector set function, and let $\mathbf{Q}, \mathbf{Q}' \in \mathcal{S}_n^{>0}(\mathbb{R})$ be two quadratic forms, then $\mathbf{Q}$ and $\mathbf{Q}'$ are equivalent if and only if $G_{\mathbf{Q},\mathcal{V}}$ and $G_{\mathbf{Q}',\mathcal{V}}$ are isomorphic. In particular there exists a bijection*

$$\mathrm{Isom}(\mathbf{Q}, \mathbf{Q}') \leftrightarrow \mathrm{Isom}(G_{\mathbf{Q},\mathcal{V}}, G_{\mathbf{Q}',\mathcal{V}}).$$

If $\mathbf{Q} = \mathbf{Q}'$, then $\mathrm{Isom}(\mathbf{Q}, \mathbf{Q}') = \mathrm{Aut}(\mathbf{Q})$ and $\mathrm{Isom}(G_{\mathbf{Q},\mathcal{V}}, G_{\mathbf{Q}',\mathcal{V}})$ are groups under function composition and the bijection is an isomorphism.

Proof. By Lemma 157 we only have to show that there exists a bijection between $\mathrm{Isom}(\mathcal{V}(\mathbf{Q}), \mathcal{V}(\mathbf{Q}'))$ and $\mathrm{Isom}(G_{\mathbf{Q},\mathcal{V}}, G_{\mathbf{Q}',\mathcal{V}})$. Because all pairwise inner products (and norms) are encoded in the graph, these

two sets are the same by definition, so the bijection (and isomorphism) is trivial. □

So to solve LIP, we can use heavily optimized graph isomorphism algorithms on the resulting graphs, such as `Nauty`, `Traces` [MP14] and `Bliss` [JK07]. These are backtrack algorithms, that try to efficiently enumerate all possible node mappings that keep the edge weights invariant. They make heavy use of graph invariants to limit the search tree, such as the degree of nodes, or the degree of all neighbours. These algorithms do not directly work on weighted graphs, but in the `Nauty` manual [McK07] there are generic transformations from weighted graphs to (somewhat larger) unweighted graphs that keep the isomorphisms intact. Depending on the size of the characteristic set $\mathcal{V}(\mathbf{Q})$, this approach can be very practical, and a public implementation by Dutour Sikirić can be found at [Dut22]. On the asymptotic side, the graph isomorphism algorithm by Babai [Bab16], gives an algorithm that runs in time quasi-polynomial in the characteristic vector set size $|\mathcal{V}(\mathbf{Q})|$. However, this time may be as large as $\exp(n^{O(1)})$ given that $|\mathcal{V}(\mathbf{Q})|$ itself may be as large as $\exp(\Theta(n))$, for example, when using Voronoi-relevant vectors.

The approach of Plesken and Pohst [PP85], later improved by Plesken and Souvignier [PS97], can be seen as a specialized variant of the above approach, where more (geometric) invariants are considered to improve the search, and possibly without explicitly constructing the full graph. Implementations by Souvignier for automorphisms and equivalence testing are available under the names `AUTO` and `ISOM` respectively[8].

9.5.2 A provable algorithm

The best asymptotic algorithm is one by Haviv and Regev [HR14] that runs in time and space $n^{O(n)}$, and returns all solutions to a LIP instance. As the lattice $\mathbb{Z}^n$ has $2^n \cdot n! = n^{O(n)}$ automorphisms, this result is in some sense optimal. A major open question is, whether there exists a single exponential time algorithm to find only a single solution to LIP.

[8] The `AUTO` and `ISOM` programs are available through MAGMA. The implementation has also been ported to PARI/GP (and thus SageMath).

The algorithm by Haviv and Regev requires short primal as well as short dual vectors. Let $\mathbf{Q}, \mathbf{Q}' \in \mathcal{S}_n^{>0}(\mathbb{R})$ be two equivalent quadratic forms. They first consider the case that $\mathrm{Min}(\mathbf{Q})$ (and $\mathrm{Min}(\mathbf{Q}')$) are full rank, i.e., $\mathrm{span}(\mathrm{Min}(\mathbf{Q})) = \mathbb{R}^n$. Fundamental to their algorithm is the so-called isolation lemma.

Lemma 163 (Isolation Lemma [HR14], informal). *there exists a relatively short dual vector $\mathbf{v} \in \mathbb{Z}^n$ (in the sense of $\|\mathbf{v}\|_{\mathbf{Q}^{-1}}$), that uniquely defines n linearly independent vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathrm{Min}(\mathbf{Q})$. These vectors are defined as follows:*
for every $1 \leq j \leq n$, the minimum standard inner product[9] $\langle \mathbf{v}, \mathbf{x} \rangle$ with vectors $\mathbf{x} \in \mathrm{Min}(\mathbf{Q}) \setminus \mathrm{span}(\mathbf{x}_1, \ldots, \mathbf{x}_{j-1})$ is uniquely achieved by $\mathbf{x}_j$.

After such an isolating vector $\mathbf{v} \in \mathbb{Z}^n$ is found for $\mathrm{Min}(\mathbf{Q})$ by enumerating short dual vectors, we do the same for $\mathrm{Min}(\mathbf{Q}')$. Once the correct image of $\mathbf{v}$ in $\mathrm{Min}(\mathbf{Q}')$ is found by a similar enumeration (which happens because short dual vectors are mapped to short dual vectors), we obtain another unique chain of n independent vectors $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathrm{Min}(\mathbf{Q}')$, and the unimodular transformation $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ can be recovered from the mapping $\mathbf{y}_i \mapsto \mathbf{x}_i$. To provably find such an isolating short dual vector we need to enumerate all dual vectors up to some norm $n^{O(1)} \cdot \lambda_1(\mathbf{Q}^{-1})$, of which there can be at most $n^{O(n)}$. Running such an enumeration can also be done in time $n^{O(n)}$, and space $\exp(O(n))$ (to store $\mathrm{Min}(\mathbf{Q}_1), \mathrm{Min}(\mathbf{Q}_2)$).

For the general case Regev and Haviv note that one can first solve LIP for the sublattices in the span of $\mathrm{Min}(\mathbf{Q})$ and $\mathrm{Min}(\mathbf{Q}')$ respectively, and (recursively) for the lattices projected away from that span. The resulting transformations can then be combined if compatible.

While the algorithm is useful as an asymptotic result, the (proven) constant in the exponent of $n^{O(n)}$ is quite large, which makes the algorithm seemingly impractical. We do not know of any implementation of this algorithm, and therefore if the practical complexity might be better.

[9] The standard inner product $\langle \mathbf{v}, \mathbf{x}_i \rangle$ has the same value as the inner product between the primal lattice vector $\mathbf{B}\mathbf{x}_j$ and the dual lattice vector $(\mathbf{B}^\top)^{-1}\mathbf{v}$ for any full rank basis $\mathbf{B}$.

9.5.3 The case of $\mathbb{Z}^n$, an approach by Szydlo

In 2003 Szydlo [Szy03] showed a search LIP to decision LIP reduction for the special case of the integer lattice $\mathbb{Z}^n$. Note that for $\mathbb{Z}^n$ the standard basis gives the quadratic form $\mathbf{I}_n$, so the search problem becomes: given $\mathbf{U}^\top\mathbf{U}$ for some unimodular matrix $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$, recover $\mathbf{U}$ (up to $\mathrm{Aut}(\mathbf{I}_n)$; i.e., all signed permutations). He shows how to solve this search problem by querying a decisional LIP oracle on lattices that are very similar to $\mathbb{Z}^n$.

Additionally, Szydlo gives a heuristic algorithm to instantiate such a decision oracle for these specific cases, that requires to sample vectors of length $O(\sqrt{n})$. With the current reduction techniques, however, it is easier to find the unusually short vectors of length 1, in any rotation of $\mathbb{Z}^n$, than to enumerate vectors of length $O(\sqrt{n})$. So even if this heuristic algorithm works, it is worse than solving the search problem directly.

An interesting open question is, whether there exists a more general search to decision reduction for LIP.

9.6 A canonical function

Using the algorithm in Section 9.5, we can compute the automorphism group, and check for equivalence between two quadratic forms of low dimension. However, in practice LIP often appears in the following setting: we have many quadratic forms $\mathbf{Q}_1, \ldots, \mathbf{Q}_N \in \mathcal{S}_n^{>0}$, and we wish to identify them up to equivalence. For example, this occurs during the enumeration (up to equivalence) of quadratic forms of low rank with special properties, such as unimodular forms, perfect forms and more. Since a naive application of an equivalence algorithm requires $O(N^2)$ equivalence tests (in the worse case), this can quickly become a bottleneck. The number of tests can be somewhat lowered if many of the invariants presented in Section 9.3 differ, which may or may not be the case.

The solution we present in this section solves the above problem by introducing a canonical form $\mathrm{Can}_{\mathcal{GL}_n(\mathbb{Z})}(\mathbf{Q}) \in \mathcal{S}_n^{>0}$ for $\mathbf{Q} \in \mathcal{S}_n^{>0}$. This canonical form should satisfy the following two requirements.

Definition 164. *We say* $\mathrm{Can}_{\mathcal{GL}_n(\mathbb{Z})} : \mathcal{S}_n^{>0} \to \mathcal{S}_n^{>0}$ *is a canonical form function if it satisfies the following two requirements:*

1. *For every* $\mathbf{Q} \in \mathcal{S}_n^{>0}$, $\mathrm{Can}_{\mathcal{GL}_n(\mathbb{Z})}(\mathbf{Q})$ *is equivalent to* $\mathbf{Q}$*; and*

2. *for every* $\mathbf{Q} \in \mathcal{S}_n^{>0}$ *and* $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$, $\mathrm{Can}_{\mathcal{GL}_n(\mathbb{Z})}(\mathbf{U}^\top \mathbf{Q} \mathbf{U}) = \mathrm{Can}_{\mathcal{GL}_n(\mathbb{Z})}(\mathbf{Q})$.

We call the output $\mathrm{Can}_{\mathcal{GL}_n(\mathbb{Z})}(\mathbf{Q})$ of $\mathbf{Q}$ the canonical form of $\mathbf{Q}$ (assuming that the function is fixed). A canonical form can be seen as a canonical representative of the equivalence class $[\mathbf{Q}]$, and thus two quadratic forms are equivalent if and only if their canonical forms are identical. Combining a canonical form with a hash table, the identification of equivalence classes in a list of N quadratic forms then takes only N canonical form computations (and N hash table lookups), and thus has the potential to be much faster. In Section 9.7 we present an application with $N \geq 10^9$ quadratic forms that became feasible largely due to the introduction of a canonical form function.

Recall from Lemma 162 that we used a characteristic vector set function $\mathcal{V}$ to turn the question of quadratic form equivalence $\mathbf{Q} \cong \mathbf{Q}'$ into one of graph isomorphism $G_{\mathbf{Q},\mathcal{V}} \cong G_{\mathbf{Q}',\mathcal{V}}$. For graphs there is already a long line of research into canonical functions $\mathrm{Can}_{\mathrm{Sym}_m}$ (e.g., [BKL80; McK81; BL83; FSS83; JK07; MP14; Bab19]). The main idea is to apply a canonical function to the graph $G_{\mathbf{Q},\mathcal{V}}$, and then lift the result back.

For simplicity we limit ourself to the set $\mathcal{G}$ of weighted graphs with (ordered) nodes $[m] = \{1, \ldots, m\}$ for some $m \geq 1$, such that automorphisms and isomorphisms can be represented by permutations. For a graph $G = ([m], w) \in \mathcal{G}$ with $m \geq 1$ nodes and weight function w, and a permutation $\pi \in \mathrm{Sym}_m$ we denote by $\pi(G) = ([m], w') \in \mathcal{G}$ the graph with weight function w' given by $w'(i,j) := w(\pi^{-1}(i), \pi^{-1}(j))$ for all $i, j = 1, \ldots, m$. Two graphs $G, G' \in \mathcal{G}$ with $m \geq 1$ nodes are called isomorphic $G \cong G'$ if there exists a $\pi \in \mathrm{Sym}_m$ such that $\pi(G) = G'$.

Definition 165. *For* $m \geq 1$ *we say that* $\mathrm{Can}_{\mathrm{Sym}_m} : \mathcal{G} \to \mathcal{G}$ *is a canonical graph ordering if it satisfies the following requirement:*

1. *For every* $G \in \mathcal{G}$, $\mathrm{Can}_{\mathrm{Sym}_m}(G)$ *is isomorphic to* G*; and*

2. *for every* $G \in \mathcal{G}$ *with* $m \geq 1$ *nodes and* $\pi \in \mathrm{Sym}_m$, $\mathrm{Can}_{\mathrm{Sym}_m}(\pi(G)) = \mathrm{Can}_{\mathrm{Sym}_m}(G)$.

In fact, the graph isomorphism implementations mentioned in Section 9.5, solve the graph isomorphism problem by computing a canonical graph ordering on both graphs. On the theoretical side there also has been some effort by Babai [Bab19] to turn the quasi-polynomial (in m) graph isomorphism algorithm [Bab16] into a canonical graph ordering algorithm.

Remark 166. Given any vector set function $\mathcal{V}$ and canonical graph ordering $\mathrm{Can}_{\mathrm{Sym}_m}$, the map $\mathbf{Q} \mapsto \mathrm{Can}_{\mathrm{Sym}_m}(G_{\mathbf{Q},\mathcal{V}})$ already defines a canonical function, in the sense that the output on two forms is identical if and only if the forms are equivalent. However this graph can be quite large, and the canonical quadratic form allows a more compact representation.

We now construct a canonical form function by lifting the canonical graph ordering back to the quadratic form setting. For this we first need another type of canonization algorithm: the Hermite Normal Form.

Definition 167 (Hermite Normal Form (HNF)). *The Hermite Normal Form of an integer matrix* $\mathbf{M} \in \mathbb{Z}^{n \times m}$ *of row rank* $r \leq n$ *is the unique matrix* $\mathbf{H} \in \mathbb{Z}^{n \times m}$ *for which there exists a* $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$, *such that* $\mathbf{M} = \mathbf{UH}$, *and moreover*

1. *The first* r *rows of* $\mathbf{H}$ *are nonzero and the remaining rows are zero; and*

2. *for* $1 \leq i \leq r$, *if* h_{i,j_i} *is the first nonzero entry in row* i, *then* $j_1 < \ldots < j_r$; *and*

3. *for* $1 \leq i \leq r$, *we have* $h_{i,j_i} > 0$; *and*

4. *for* $1 \leq k < i \leq r$, *we have* $0 \leq h_{k,j_i} < h_{i,j_i}$.

The HNF of a matrix is computable in polynomial time [KB79; MW01]. The HNF can be seen as a canonical function of the left action of $\mathcal{GL}_n(\mathbb{Z})$ given by multiplication on integer matrices $\mathbb{Z}^{n \times m}$.

In particular it turns any (transpose) basis of a lattice into a (transpose) canonical basis. Note however that it is only canonical up to the action of $\mathcal{GL}_n(\mathbb{Z})$, and not with respect to the action of $\mathcal{O}_n(\mathbb{R})$. Therefore it cannot directly be used to solve LIP. Furthermore, given that $\mathcal{V}(\mathbf{U}^\top \mathbf{Q}\mathbf{U}) = \mathbf{U}^{-1}\mathcal{V}(\mathbf{Q})$, one might be temped to apply HNF directly to the characteristic set to make the action of $\mathcal{GL}_n(\mathbb{Z})$ canonical, however, the output of the HNF does depend on the order of the characteristic vectors, i.e., it is not simultaneously canonical with respect to the permutation action of Sym_m. Obtaining a canonical ordering is precisely where we use the canonical graph function.

Algorithm 15: A canonical form function $\mathrm{Can}_{\mathcal{GL}_n(\mathbb{Z})}$.

Given : A characteristic set function $\mathcal{V}$, and a canonical graph function $\mathrm{Can}_{\mathrm{Sym}_m}$.
Input : A quadratic form $\mathbf{Q} \in \mathcal{S}_n^{>0}$.
Output: A canonical representative $\mathbf{Q}' \in [\mathbf{Q}]$.

1 Compute the characteristic vector set $\mathcal{V}(\mathbf{Q})$, $m \leftarrow |\mathcal{V}(\mathbf{Q})|$
2 Construct the weighted graph $G_{\mathbf{Q},\mathcal{V}}$
3 Run $\mathrm{Can}_{\mathrm{Sym}_m}$ on $G_{\mathbf{Q},\mathcal{V}}$ to obtain a canonical ordering $\mathbf{v}_1, \ldots, \mathbf{v}_m$ of $\mathcal{V}(\mathbf{Q})$
4 $\mathbf{M} \leftarrow (\mathbf{v}_1, \ldots, \mathbf{v}_m) \in \mathbb{Z}^{n \times m}$
5 Compute $\mathbf{U}_{\mathcal{V}(\mathbf{Q})} \in \mathcal{GL}_n(\mathbb{Z})$ such that $\mathbf{M} = \mathbf{U}_{\mathcal{V}(\mathbf{Q})}\mathbf{H}$ is the unique HNF decomposition of $\mathbf{M}$
6 **return** $\mathrm{Can}_{\mathcal{GL}_n(\mathbb{Z})}(\mathbf{Q}) = \mathbf{U}_{\mathcal{V}(\mathbf{Q})}^\top \mathbf{Q}\mathbf{U}_{\mathcal{V}(\mathbf{Q})}$

Proposition 168. *Algorithm 15 is a canonical form function.*

Proof. We first show that the algorithm is well defined, i.e., that it does not depend on the initial ordering of the characteristic vector set $\mathcal{V}(\mathbf{Q})$. Note that the output ordering of $\mathrm{Can}_{\mathrm{Sym}_m}$ is unique up to $\mathrm{Aut}(G_{\mathbf{Q},\mathcal{V}})$, and thus by Lemma 162 the ordering of the characteristic vectors is unique up to $\mathrm{Aut}(\mathbf{Q})$. For any such different ordering $\mathbf{S}\mathbf{v}_1, \ldots, \mathbf{S}\mathbf{v}_m$ for $\mathbf{S} \in \mathrm{Aut}(\mathbf{Q})$ would lead to the matrix $\mathbf{S}\mathbf{U}_{\mathcal{V}(\mathbf{Q})}$, and thus $\mathbf{U}_{\mathcal{V}(\mathbf{Q})}$ is well-defined as a coset representative in $\mathrm{Aut}(\mathbf{Q})\backslash\mathcal{GL}_n(\mathbb{Z})$. As $\mathbf{Q}$ is by definition invariant under $\mathrm{Aut}(\mathbf{Q})$ the output form is thus well-defined.

We now verify the two properties in Definition 164. The first property is clear by definition. For the second property note that for any $\mathbf{V} \in \mathcal{GL}_n(\mathbb{Z})$, we have

$$\mathbf{U}_{\mathcal{V}(\mathbf{V}^\top \mathbf{Q}\mathbf{V})} \equiv \mathbf{U}_{\mathbf{V}^{-1}\mathcal{V}(\mathbf{Q})} \equiv \mathbf{V}^{-1}\mathbf{U}_{\mathcal{V}(\mathbf{Q})} \in \mathrm{Aut}(\mathbf{V}^\top \mathbf{Q}\mathbf{V}) \setminus \mathcal{GL}_n(\mathbb{Z}),$$

and thus $\mathrm{Can}_{\mathcal{GL}_n(\mathbb{Z})}(\mathbf{V}^\top \mathbf{Q}\mathbf{V}) = \mathrm{Can}_{\mathcal{GL}_n(\mathbb{Z})}(\mathbf{Q})$ as desired. Here the first equivalence stems from the transformation property of the characteristic set, and the second equivalence stems from the uniqueness of the HNF. □

9.7 Applications to perfect form enumeration

The technique to construct a canonical form from graphs is quite versatile and has many applications. For example, a similar function can be constructed for symplectic groups [DHVW20], or for the equivalence of C-type domains [DMW22]. Beyond testing for equivalence, a canonical function also gives a deterministic naming scheme for lattices (for a fixed function). In this section, we dive in a bit deeper on applying the canonical form function for perfect form enumeration, and we discuss a further improvement that gives a 30-fold speed-up in practice.

9.7.1 Background on perfect lattices and forms

The search for perfect forms is motivated by the Lattice Packing Problem (LPP): how to pack n-dimensional balls in $\mathbb{R}^n$ in a lattice pattern, such that their density, the proportion of $\mathbb{R}^n$ they fill, is maximized? So far LPP has only been solved up to dimension 8 and in dimension 24 (by the remarkable Leech lattice). It is known that every optimal lattice packing is attained by a perfect lattice, and up to similarity there are only a finite number of them in each dimension. Therefore, to solve LPP, one can simply enumerate all perfect lattices.

From a quadratic form perspective the goal of LPP is to find a form $\mathbf{Q} \in \mathcal{S}_n^{>0}$ that maximizes $\lambda_1(\mathbf{Q})^2/\det(\mathbf{Q})^{1/n}$. This is a search problem over the cone of (positive definite) quadratic forms $\mathcal{S}_n^{>0}$. As the objective function is invariant under scaling we can restrict the

space to those forms with $\lambda_1(\mathbf{Q})^2 \geq 1$, which defines the Ryshkov polyhedra

$$\mathcal{P} := \{\mathbf{Q} \in \mathcal{S}_n^{>0} : \lambda_1(\mathbf{Q})^2 \geq 1\}.$$

The inequality $\lambda_1(\mathbf{Q})^2 \geq 1$ is equivalent to the infinite set of (linear) inequalities $\mathbf{x}^\top \mathbf{Q}\mathbf{x} \geq 1$ for all nonzero $\mathbf{x} \in \mathbb{Z}^n$, which implies that the Ryshkov polyhedra is an intersection of infinitely many halfspaces. Over the Ryshkov polyhedra we now only have to minimize the determinant. Minkowski showed that the function $\mathbf{Q} \mapsto \det(\mathbf{Q})^{1/n}$ is concave on $\mathcal{S}_n^{>0}$, which implies that the optima lie at the vertices of the Ryshkov polyhedra, and we call the quadratic forms that correspond to these vertices *perfect*. Up to equivalence (and scaling) there are only a finite number of distinct perfect forms. In fact, we have an upper bound of $\exp(O(n^2))$ on the number of equivalence classes [Woe20]. Voronoi's Algorithm [Vor08] explores the Ryshkov polyhedra up to equivalence, and thereby enumerates all perfect forms in a fixed dimension. It has been successfully used to find all perfect forms in dimensions up to 8, with respectively $1, 1, 1, 2, 3, 7, 33, 10916$ equivalence classes in each dimension.

While in dimension 8 the main bottleneck was due to high degeneracy in the Ryshkov polyhedra, we have an additional problem in dimension 9: the large number of perfect forms. With possibly billions of non-equivalent perfect forms we require a very efficient canonical function to quickly check if an encountered perfect form is already known or not. Previous enumeration attempts in dimension 9, stranded at only 500.000 [SSV07] or 25 million [Woe18] forms respectively, with equivalence computations as the main bottleneck.

9.7.2 An improved canonical function

Ignoring the few highly degenerate vertices (perfect forms), the main cost of enumerating all 9-dimensional perfect forms is the computation of canonical forms. Any speed-up in this computation directly translates to a significantly reduced running time to finish the enumeration. We discuss one trick to achieve this. For perfect forms the most costly step in Algorithm 15 is that of computing the canonical graph function, e.g., using `Nauty`, `Traces` [MP14] or `Bliss` [JK07]. The runtime of these algorithms depends on many factors, but when restricting to our application the runtime mostly seems to depend on

the number of vertices $|\mathcal{V}(\mathbf{Q})|$, and the number of distinct weights, i.e., $|\{\langle \mathbf{x}, \mathbf{y}\rangle_{\mathbf{Q}} : \mathbf{x}, \mathbf{y} \in \mathcal{V}(\mathbf{Q})\}|$.

In order to improve the algorithm we thus have to try to limit the size of this graph. The minimal vectors of a perfect form often (except for a few cases) already span $\mathbb{Z}^n$, and thus we have $\mathcal{V}(\mathbf{Q}) := \mathcal{V}_{\text{ms}}(\mathbf{Q}) = \text{Min}(\mathbf{Q})$. Reducing this set any further seems hard (and is actually impossible if the automorphism group acts transitively). Note, however, that we have the sign symmetry $\mathbf{x} \in \mathcal{V}_{\text{ms}}(\mathbf{Q}) \Leftrightarrow -\mathbf{x} \in \mathcal{V}_{\text{ms}}(\mathbf{Q})$. What happens if we ignore the signs?

Similarly to $G_{\mathbf{Q},\mathcal{V}}$ we can construct the *absolute* graph $G^{\text{abs}}_{\mathbf{Q},\mathcal{V}}$, where each node corresponds to some $\pm\mathbf{x} \in \mathcal{V}(\mathbf{Q})$, and the weight function is (well-)defined as $w(\pm\mathbf{x}, \pm\mathbf{y}) := |\langle \mathbf{x}, \mathbf{y}\rangle_{\mathbf{Q}}|$.

Lemma 169. *For any perfect form $\mathbf{Q} \in \mathcal{S}_n^{>0}$, and the characteristic set function $\mathcal{V}_{\text{ms}}$, we have an injective homomorphism*

$$\begin{aligned} \text{Aut}(G_{\mathbf{Q},\mathcal{V}_{\text{ms}}})/\langle \pm\mathbf{I}_n\rangle &\hookrightarrow \text{Aut}(G^{abs}_{\mathbf{Q},\mathcal{V}_{\text{ms}}}), \\ \pi &\mapsto \pi^{abs}, \end{aligned}$$

where $-\mathbf{I}_n$ sends $\mathbf{x} \mapsto -\mathbf{x}$ for all $\mathbf{x} \in \mathcal{V}_{\text{ms}}$.

Proof. Let $\pi \in \text{Aut}(G_{\mathbf{Q},\mathcal{V}_{\text{ms}}})$. The inner product constraints enforce that if π maps $\mathbf{x} \mapsto \mathbf{y}$, then $-\mathbf{x} \mapsto -\mathbf{y}$ for $\mathbf{x}, \mathbf{y} \in \mathcal{V}_{\text{ms}}$. So π induces a permutation π^{abs} on $G^{\text{abs}}_{\mathbf{Q},\mathcal{V}_{\text{ms}}}$ by mapping pairs to pairs $\pm\mathbf{x} \mapsto \pm\mathbf{y}$. Because π is invariant with respect to the inner products, π^{abs} is clearly invariant with respect to the absolute invariants, and thus $\pi^{\text{abs}} \in \text{Aut}(G^{\text{abs}}_{\mathbf{Q},\mathcal{V}_{\text{ms}}})$.

For the injectivity we have to show that the kernel is $\langle \pm\mathbf{I}_n\rangle$. First suppose (as we will later prove) that the graph $G_{\mathbf{Q},\mathcal{V}_{\text{ms}}}$ is connected if we remove all edges of weight 0 . Let $\pi \in \text{Aut}(G_{\mathbf{Q},\mathcal{V}_{\text{ms}}})$ be such that π^{abs} is trivial. Every $\mathbf{x} \in \mathcal{V}_{\text{ms}}$ is mapped to either $\mathbf{x}$ or $-\mathbf{x}$. However any pair $\mathbf{x}, \mathbf{y}$ is connected by some path with nonzero (signed) inner products in $G_{\mathbf{Q},\mathcal{V}_{\text{ms}}}$, and thus any choice for $\mathbf{x} \mapsto \mathbf{x}$ or $\mathbf{x} \mapsto -\mathbf{x}$ determines the signs for all other elements. So there are only two choices for π, and clearly the identity map $\mathbf{I}_n$ and the negation map $-\mathbf{I}_n$ satisfy this.

What remains is to shoe that the graph $G_{\mathbf{Q},\mathcal{V}_{\text{ms}}}$ is connected. We have $\text{Min}(\mathbf{Q}) \subset \mathcal{V}_{\text{ms}}$, and by definition of perfect, the set $\{\mathbf{x}\mathbf{x}^\top : \mathbf{x} \in \text{Min}(\mathbf{Q})\}$ has full rank $\frac{1}{2}n(n+1)$ inside the space of $n \times n$ symmetric

matrices. Assume for contradiction that the graph $G_{\mathbf{Q},\mathcal{V}_{\text{ms}}}$ is disconnected if we remove the edges of weight 0. This implies that we can decompose $\mathcal{V}_{\text{ms}} = V_1 \sqcup V_2$ into two disjunct non-empty sets such that $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{Q}} = 0$ for all pairs $(\mathbf{x}, \mathbf{y}) \in V_1 \times V_2$. Let $1 \leq r_i := \text{rk}(V_i)$ be the rank of the subspace spanned by V_i inside $\mathbb{R}^n$, and note that the pairwise orthogonality gives $r_1 + r_2 \leq n$. By the (strict) convexity of the map $c \mapsto c(c+1)$ for $c > 0$ we then obtain the following contradiction

$$\frac{1}{2}n(n+1) \leq \sum_{i=1}^{2} \text{rk}\{\mathbf{x}\mathbf{x}^{\top} : \mathbf{x} \in V_i\} \leq \sum_{i=1}^{2} \frac{1}{2} r_i(r_i+1) < \frac{1}{2}n(n+1),$$

and thus the graphs are connected. □

If the above injective morphism is in fact an isomorphism, and we have an efficient way to lift π^{abs} back to π (up to sign), then we could simply use $G^{\text{abs}}_{\mathbf{Q},\mathcal{V}}$ instead of $G_{\mathbf{Q},\mathcal{V}}$, with half as many nodes and almost half as many distinct weights. In fact an algorithm for the inverse map immediately follows from the proof of Lemma 9.7.2: fix a single sign choice, and determine the other signs uniquely via a spanning tree of the graph.

In general however, removing the signs could also introduce more automorphisms, so we have to check if this is the case or not. The strategy is as follows, first we compute the canonical function using $G^{\text{abs}}_{\mathbf{Q},\mathcal{V}}$. As a cheap by-product of the canonical graph algorithm we also obtain generators for $\text{Aut}(G^{\text{abs}}_{\mathbf{Q},\mathcal{V}})$, and we check if each of these (unsigned) generators π^{abs} correctly lifts to a (signed) automorphism π of $G_{\mathbf{Q},\mathcal{V}}$. If this is the case then we are done, if not then we recompute the full canonical function using $G_{\mathbf{Q},\mathcal{V}}$. Our hope is that the latter almost never happens, and thus we can rely on a much smaller, and thus more efficient, canonical graph computation.

9.7.3 Results

We implemented a canonical function, highly optimized for the perfect form use-case. As a characteristic function we used $\mathcal{V}_{\text{ms}}(\mathbf{Q})$, which in the case of perfect forms is often just $\text{Min}(\mathbf{Q})$ and thus quite small. For the canonical graph function we implemented both `BLISS` [JK07] and `Traces` [MP14].

	Graph Variant		
Canonical Library	Signed	Absolute	speed-up
`BLISS` [JK07]	3558 ± 613 µs	398 ± 129 µs	8.94×
`Traces` [MP14]	601 ± 84 µs	223 ± 36 µs	2.70×

Table 9.1: The average number of microseconds and the standard deviation per full canonical form computation on the 524.288 9-dimensional perfect forms found by [SSV07]. The absolute graph timings include the check and fallback to the signed case if needed.

The timings for this highly optimized implementation, and the additional speed-up of the absolute graph improvement, are shown in Table 9.1. The low dimension allowed to implement most steps of the algorithm with primitive types (no extended precision), which gave a large speed-up over the original generic implementation. In addition, depending on the canonical graph library we obtain a speed-up of 2.7 to 8.94 using the absolute graph approach, even including the fallback to the original case if needed. The generic implementation from the original paper (using `Traces`) took an average of 5941µper canonical form on the same dataset and same hardware [DHVW20; Dut22]; in total we obtain a 27-fold speed-up over this implementation.

Exploring all perfect forms in dimension 9 requires roughly $2 \cdot 10^{11}$ canonical form computations. Based on the experiments, the extra improvement thus reduced the running time spend on this part of the algorithm from roughly 330.000 core-hours to less than 12.500 core-hours, making (that part of the) computation feasible on a high-end server or a small cluster.

Finishing the full enumeration is an ongoing joint project with Mathieu Dutour Sikirić. The canonical form function allowed us to find (probably) all perfect forms, with only a few exceptionally hard cases yet to treat.

Lemma 170. *There are at least* 2,237,251,033 *non-similar perfect forms in dimension* 9.

Conjecturally the above inequality is tight and we have found all perfect forms in dimension 9, of which the densest corresponds to the

laminated lattice Λ_9. To be certain about this we still have to enumerate all neighbours of a few perfect forms (with many neighbours).

The same idea, using the absolute graph canonical function, has been used to make the enumeration of all 55,083,358 C-type domains in dimension 6 feasible [DMW22].