The isomorphism problem: from lattices to graphs

Thomas Camus

ID Quantique SA - Chemin de la Marbrerie 3, 1227 Carouge – Switzerland
E-mail: thomas.camus@idquantique.com

Abstract

We study the algorithmic complexity of the *Lattices Isometry Problem* (LIP), the aim of which is to decide whether two given lattices are isometric. We prove that a weakened version of this problem is reducible to the famous Graphs Isomorphism Problem (GIP). Used in combination with the recent quasi-polynomial resolution of GIP due to Babai [6], this reduction allows us to exhibit an algorithm that solves LIP in a time quasi-polynomial in the number of relatively short vectors in the lattices considered.

2010 Mathematics Subject Classification. **11H56**.
Keywords. lattices, graphs, isomorphism problem, automorphism group, computational aspect and explicit method.

1 Introduction

An isometry between two n-dimensional Euclidean lattices Λ and Λ' is an isometry $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ such that $f(\Lambda) = \Lambda'$. The problem of *deciding* the existence of isometries between lattices (known as LIP, for *Lattices Isometry Problem*) as well as *computing* them naturally arises when one deals with effective variants of fundamental number-theoretic results, such as the computation of the cohomology of arithmetic groups and of the K-theory of $\mathbb{Z}$ [13, 30, 14]. This question is also of significant interest in the field of lattice-based cryptography, which is at the very core of the transition towards post-quantum technologies [3, 4]. Despite the fact that the still commonly used LIP solver due to Plesken and Souvignier [28] is more than twenty years old, the study of the asymptotic complexity of this question has regained in interest in the past decade, notably through the results of Dutour Sikirić, Schürmann, and Vallentin [12], and Haviv and Regev [18].

The Lattice Isometry Problem can be understood as the specification in the language of lattices of the isomorphism problem, which aims at deciding if two generic mathematical structures share the same properties. The most widely-known version of this question is perhaps the *Graphs Isomorphism Problem* (GIP). It has become a central question in computational complexity, mainly because it is one of the mathematical problems for which we do not have a polynomial-time solver, but whose **NP**-completeness would imply that the polynomial-time hierarchy collapses, hence that **P** = **NP**. Since more than forty years, effective algorithms for tackling GIP have got closer and closer to the polynomial-time bound, culminating in the quasi-polynomial solver due to Babai [6].

Dutour Sikirić, Schürmann, and Vallentin have proven in [12] that LIP is at least as hard as GIP. To the best of our knowledge, this reduction establishes one of the first relations between lattices and graphs when dealing with the notion of isomorphism. In the present article, we complement this result by establishing that up to the computation of a set of relatively short vectors of the lattices considered, LIP is polynomial-time reducible to GIP. From this novel reduction and from known bounds on GIP solvers, we also exhibit new estimates of the computational complexity of LIP.

Advanced Studies: Euro-Tbilisi Mathematical Journal Special Issue (9 - 2021), pp. 81–104.

Tbilisi Centre for Mathematical Sciences.
Received by the editors: 28 December 2020.
Accepted for publication: 23 May 2021.

The paper is organized as follows. In §2 we recall basic results about graphs and lattices, and we introduce several algorithmic problems dealing with these objects. In §3, §4 and §5 we detail successive reductions between lattices and different families of graphs. In §6 we extract from these reductions our main result, linking the complexity of computing isomorphisms between graphs and the complexity of computing isometries between lattices. Finally, we exhibit in §7 a set of estimates of the amount of relatively short elements of a lattice, which allows to provide more explicit version of the reductions detailed in the previous paragraphs.

Acknowledgment. Most of the results detailed in the present paper arise from the author's Ph.D. thesis [9], carried out at the Institut Fourier[1,2]. The author is especially thankful to his supervisor Ph. Elbaz-Vincent for the wealth of helpful discussions and comments he provided. The author also benefited from the financial support of the LabEx PERSYVAL-Lab (ANR-11-LABX-2005) and of the Marie Skłodowska-Curie European Individual Fellowship (H2020-MSCA-IF-2017, project 796619).

2 Lattices, graphs, and related algorithmic problems

2.1 Graphs

All graphs considered in this paper are *finite* (*i.e.* with only finitely many vertices and edges) and *undirected* (*i.e.* edges are unordered pairs of vertices). Note that graphs we consider may contain loops (but multiple edges between the same pair of vertices are not allowed), hence we do not restrict ourselves to simple graphs. If G is a graph with vertices $v_1, \dots, v_N$, its *incidence matrix* is the symmetric matrix of size N whose (i,j)-th coefficient is 1 if there is an edge between v_i and v_j and 0 otherwise. The graph G is said to be:

- *complete* if there is an edge between every pair of vertices of G, or equivalently if the coefficients of its incidence matrix are all equal to 1,
- *connected* if there is a path of edges linking every pair of vertices of G.

The *density* $\rho(G)$ of G is defined as

$$\rho(G) := \frac{2|E|}{N(N+1)} \in [0,1],$$

where $|E|$ is the number of edges in G (including loops). Note that when restricted to *simple* graphs, the density is usually defined as $2|E|/N(N-1)$. The graphs we consider may contain loops which have to be factored in density computations, thus explaining the slightly unusual definition chosen here. The density of G is also the density of the upper (or lower) triangle (including the diagonal) of its incidence matrix. Let us recall that as for matrices, the density plays an important role in the computations involving graphs: some algorithms are fully optimized for sparse graphs and may be quite slow for dense graphs, and vice-versa (for example, the popular program `Nauty/Traces` [26] has different implementations for dense and sparse graphs).

We are mainly interested in the notion of isomorphism between graphs. An *isomorphism* between two graphs G and G' with sets of vertices V and V' respectively is a bijection $f : V \longrightarrow V'$

[1] Institut Fourier – UMR 5582, Université Grenoble Alpes, France.
[2] https://www-fourier.ujf-grenoble.fr

which preserves the incidence relations: for all $v_1, v_2 \in V$, there is an edge between v_1 and v_2 if and only if there is one between $f(v_1)$ and $f(v_2)$. An isomorphism between G and itself is called an *automorphism* of G, and the set $\mathrm{Aut}(G)$ of all automorphisms of G is a finite group. Specifically, we focus our attention on two famous computational problems on graphs:

- *Graphs Isomorphism Problem* (abbreviated GIP): decide whether two given graphs are isomorphic.
- *Graph Automorphism Problem* (abbreviated GAP): compute a generating set of the automorphism group of a given graph.

Throughout this paper, by *computing a group* H, we mean *computing a generating set of* H. Note that computing a generating set of a group and enumerating all elements of a group are two slightly different problems. It is well known that there is a polynomial-time reduction from GIP to GAP. We recall a simple proof of this fact, which will be modified later on to fit the case of lattices.

Proposition 2.1. There is a polynomial-time reduction from GIP to GAP. More precisely, deciding if two graphs with N vertices are isomorphic is reducible to computing the automorphism group of a graph with $2N$ vertices.

Proof. Let G and G' be two graphs with N vertices. Without any loss of generality, we may assume that they are connected. Let $G \sqcup G'$ be the disjoint union of G and G'. The graphs G and G' are isomorphic if and only if there is an automorphism of $G \sqcup G'$ permuting G and G'. Moreover, if such an automorphism exists, any generating set of $\mathrm{Aut}(G \sqcup G')$ must contain one. Q.E.D. Q.E.D.

The automorphism and isomorphism problems for graphs are heavily studied. It is known that GIP is an **NP** problem which is not **NP**-complete unless the polynomial-time hierarchy collapse (see [5, §8.2.4, p.156–157]). Such a collapse notably implies that $\mathbf{P} = \mathbf{NP}$, an equality widely believed to be not satisfied (see [15]). GIP has also been proven to be solvable in polynomial time for many special classes of graphs (*e.g.* trees [22], planar graphs [20], and graphs of bounded valence [24]). Until 2015, the best theoretical algorithm known for tackling the general GIP was the one of Babai and Luks [7], whose time complexity in the number n of vertices is $2^{O(\sqrt{n \log n})}$, but Babai recently proposed a quasi-polynomial algorithm with running time $\exp(\log(n)^{O(1)})$ (see [6]). It is also worth noting that GIP has been linked to many other computational problems dealing with the notion of isomorphism between various mathematical objects (see for example [8, 32]).

A *vertex-labeled graph* is a graph together with a labeling of the vertices, *i.e.* a function from the set of vertices of the graph to a finite set of labels. Throughout this paper we will generally assume that vertices are labeled by positive integers. Edge-labeled graphs are defined similarly: an *edge-labeled graph* is a graph together with a labeling of the edges. Isomorphisms for vertex-labeled graphs and edge-labeled graphs are required to preserve the labeling (which is stronger than preserving only the equivalence relation given by pairs of vertices or edges with the same label). The versions of GIP and GAP for vertex-labeled graphs (respectively for edge-labeled graphs) will be denoted VLGIP and VLGAP (respectively ELGIP and ELGAP).

2.2 Lattices

Throughout this paper, the $\mathbb{R}$-vector space $\mathbb{R}^n$ is equipped with its standard inner product, defined for all $x, y \in \mathbb{R}^n$ by $\langle x \,|\, y \rangle := \sum_{i=1}^{n} x_i y_i$, with associated Euclidean norm $\|x\| := \sqrt{\langle x \,|\, x \rangle}$.

By *lattice*, we mean *Euclidean lattice of full rank*, that is to say a maximal discrete subgroup of $\mathbb{R}^n$ for some $n \in \mathbb{N}_{\geqslant 1}$, called the *dimension* of the lattice. A lattice $\Lambda \subset \mathbb{R}^n$ can be expressed as the set integral combinations of a basis of $\mathbb{R}^n$: there exists $(b_1, \ldots, b_n)$ a $\mathbb{R}$-basis of $\mathbb{R}^n$ such that

$$\Lambda = \left\{ \sum_{i=1}^{n} x_i b_i \ : \ x_i \in \mathbb{Z} \text{ for all } 1 \leqslant i \leqslant n \right\}.$$

Such a family is called a *basis* of Λ. Note that it also possible to start with an $\mathbb{R}$-basis $\mathcal{B}$ of $\mathbb{R}^n$ and to consider the *lattice generated by* $\mathcal{B}$, denoted $\Lambda(\mathcal{B})$. The *minimum* $m(\Lambda)$ of Λ is defined as

$$m(\Lambda) := \inf_{x \in \Lambda \setminus \{0\}} \|x\|^2.$$

Since Λ is a closed and discrete subgroup of $\mathbb{R}^n$, the set

$$S(\Lambda) := \{x \in \Lambda \ : \ \|x\|^2 = m(\Lambda)\}$$

is finite and non-empty. The elements of $S(\Lambda)$ are called *short vectors* of Λ.

Two lattices Λ and Λ' in $\mathbb{R}^n$ are said to be *isometric* if there exists an orthogonal linear transformation $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ such that $f(\Lambda) = \Lambda'$. An isometry between Λ and itself is called an automorphism of Λ. It is well-known that the set of automorphisms of Λ is a finite group (see [25, thm. 1.4.2, p.12]), denoted $\mathrm{Aut}(\Lambda)$. This paper deals with two fundamental computational problems related to the notion of isometric lattices:

- *Lattices Isometry Problem* (abbreviated LIP): given $\mathcal{B}$ and $\mathcal{B}'$ two bases of $\mathbb{R}^n$, decide whether the lattices $\Lambda(\mathcal{B})$ and $\Lambda(\mathcal{B}')$ are isometric.
- *Lattice Automorphisms Problem* (abbreviated LAP): given $\mathcal{B}$ a basis of $\mathbb{R}^n$, compute (a generating set of) $\mathrm{Aut}(\Lambda(\mathcal{B}))$.

As for graphs, we start by proving that LIP is polynomial-time reducible to LAP. Note that proofs for graphs and lattices a very similar.

Proposition 2.2. There is a polynomial-time reduction from LIP to LAP. More precisely, deciding if two lattices of $\mathbb{R}^n$ are isometric is reducible to computing the automorphism group of a lattice in $\mathbb{R}^{2n}$.

Proof. Let Λ and Λ' be two lattices of $\mathbb{R}^n$ with bases $\mathcal{B}$ and $\mathcal{B}'$ respectively. Let $\Lambda \oplus \Lambda'$ be the lattice of $\mathbb{R}^{2n}$ generated by the $\mathbb{R}$-basis $\begin{bmatrix} \mathcal{B} & 0 \\ 0 & \mathcal{B}' \end{bmatrix}$. The lattices Λ and Λ' are isometric if and only if there is an automorphism of $\Lambda \oplus \Lambda'$ permuting Λ and Λ'. Moreover, if such an automorphism exists, any generating set of $\mathrm{Aut}(\Lambda \oplus \Lambda')$ must contain one. Q.E.D.

Despite the fact that these problems play important roles in numerous domains, in contrast to the case of graphs we do not know much about their complexity. The most practical and widely implemented algorithm for tackling LIP and LAP is the one of Plesken and Souvignier [28], but to the best of our knowledge no complexity analysis has been conducted on this algorithm. Haviv and Regev presented in [18] a theoretical algorithm which enumerates all isometries between two

n-dimensional lattices in time $n^{O(n)}s^{O(1)}$, where s is the input size. Up to constants, this algorithm has optimal running time, but the problem of enumerating all isometries between two lattices is quite different from the one of deciding if one exists.

Note that similarly to GIP, LIP is in the **NP** complexity class but unlikely to be **NP**-complete. Indeed, if LIP is **NP**-complete, then the polynomial-time hierarchy collapses (essentially because LIP lies in the **SZK** complexity class). More details about this result can be found in [18, §1 and §5.2]. Nevertheless, all algorithms currently known for tackling LIP and LAP (including [28] and [18]) require the computation of sets of the form $\{x \in \Lambda \,:\, \|x\| = C\}$ for various $C > 0$, and the problem of computing such sets is very likely to be **NP**-hard. Indeed, Charles proved in [10, §3] that counting the elements of a given norm in a lattice is ♯**P**-hard. Moreover, enumerating elements of $\{x \in \Lambda \,:\, \|x\| = C\}$ is generally done through the computation of $\{x \in \Lambda \,:\, \|x\| \leqslant C\}$, and it is well-known that computing such sets is **NP**-hard under probabilistic reductions [1, 2]. Even if we take these computations into account, not much is known about the complexity of LIP and LAP. This fact motivates the introduction of weakened variants of these problems. If Λ is a lattice in $\mathbb{R}^n$ and $X := (x_1, \ldots, x_k)$ is a family of non-zero elements of $\mathbb{R}^n$, let

$$S(\Lambda, X) := \bigcup_{i=1}^{k} \{x \in \Lambda \,:\, \|x\| = \|x_i\|\}.$$

Let $\mathcal{B} := (b_1, \ldots, b_n)$ be a basis of Λ and Λ' be another lattice in $\mathbb{R}^n$. Clearly, if f is an isometry between Λ and Λ', then $f(b_i)$ is an element of $S(\Lambda', \mathcal{B})$ for all $1 \leqslant i \leqslant n$. Hence, the problem of finding an isometry between Λ and Λ' reduces to the problem of finding an n-tuple $(x_1, \ldots, x_n)$ in $S(\Lambda', \mathcal{B})$ such that $\langle x_i \,|\, x_j \rangle = \langle b_i \,|\, b_j \rangle$ for all $1 \leqslant i, j \leqslant n$. Therefore, we consider weakened versions of LIP and LAP where the sets $S(\Lambda, \mathcal{B})$ are given:

- *Lattices Isometry Problem with sets* $S(\cdot, \cdot)$ (abbreviated LIP+S): given $\mathcal{B}$ and $\mathcal{B}'$ two bases of $\mathbb{R}^n$ and the sets $S(\Lambda(\mathcal{B}), \mathcal{B})$ and $S(\Lambda(\mathcal{B}'), \mathcal{B})$, decide whether Λ and Λ' are isometric lattices.
- *Weakened Lattice Automorphisms Problem with set* $S(\cdot, \cdot)$ (abbreviated LAP+S): given $\mathcal{B}$ a basis of $\mathbb{R}^n$ and the set $S(\Lambda(\mathcal{B}), \mathcal{B})$, compute $\mathrm{Aut}(\Lambda)$.

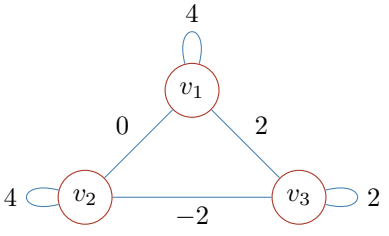
When they are given, we may generally assume that the sets $S(\Lambda, \mathcal{B})$ and $S(\Lambda', \mathcal{B})$ have the same cardinality. Indeed, if it is not the case, Λ is not isometric to Λ'.

The remainder of this article aims at proving that LIP+S and LAP+S are polynomially reducible to GIP and GAP respectively. Besides providing links to well-known and intensively studied problems, these reductions highlight interesting properties of LIP and LAP. Moreover, these results supplement the polynomial-time reduction of GIP to LIP established by Dutour Sikirić, Schürmann, and Vallentin [12].

3 From lattices to edge-labeled graphs

Given a finite subset $S \subset \mathbb{R}^n$, we denote by G_S the complete edge-labeled graph whose vertices are the elements of S and the label of the edge between x and y is $\langle x \,|\, y \rangle$.

Example 3.1. Let $S := \{(2, 0), (0, -2), (1, 1)\} = \{v_1, v_2, v_3\} \subset \mathbb{R}^2$. The associated edge-labeled graph G_S is shown on the Figure 1.



$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 4 & 0 & 2 \\ 0 & 4 & -2 \\ 2 & -2 & 2 \end{bmatrix} \end{array}$$

Figure 1: The edge-labeled graph G_S and its incidence matrix.

Let $\Lambda \subset \mathbb{R}^n$ be a lattice and $\mathcal{B} := (b_1, \ldots, b_n)$ be one of its bases. Since the set $S(\Lambda, \mathcal{B})$ defined in the previous section is finite, we can consider the edge-labeled graph $G_{S(\Lambda,\mathcal{B})}$ associated to the pair $(\Lambda, \mathcal{B})$, which will be simply denoted $G(\Lambda, \mathcal{B})$. Every automorphism of Λ preserves both the set $S(\Lambda, \mathcal{B})$ and inner products of its elements; therefore every such automorphism induces an automorphism of $G(\Lambda, \mathcal{B})$. Moreover, since $S(\Lambda, \mathcal{B})$ contains the base $\mathcal{B}$, distinct automorphisms of Λ induce distinct automorphisms of $G(\Lambda, \mathcal{B})$. In fact, every automorphism of $G(\Lambda, \mathcal{B})$ comes from an automorphism of Λ:

Proposition 3.2. Let σ be an automorphism of $G(\Lambda, \mathcal{B})$. There exists a unique automorphism u of Λ such that $u(x) = \sigma(x)$ for all $x \in S(\Lambda, \mathcal{B})$. In particular, the group $\mathrm{Aut}(\Lambda)$ is explicitly isomorphic to $\mathrm{Aut}(G(\Lambda, \mathcal{B}))$.

Proof. For all $1 \leqslant i \leqslant n$, let $b'_i := \sigma(b_i)$. Let B and B' be the matrices whose columns are $(b_1, \ldots, b_n)$ and $(b'_1, \ldots, b'_n)$ respectively. Since σ preserves the labeling of $G(\Lambda, \mathcal{B})$, the equality $\langle b_i \,|\, b_j \rangle = \langle b'_i \,|\, b'_j \rangle$ is satisfied for all $1 \leqslant i, j \leqslant n$. Therefore, we have $B^\intercal B = B'^\intercal B'$, which shows that the matrix $Q := B'B^{-1}$ is orthogonal. Let us show that $Qx = \sigma(x)$ for all $x \in S(\Lambda, \mathcal{B})$. Let $x \in S(\Lambda, \mathcal{B})$ and $1 \leqslant i \leqslant n$. Since the matrix Q is orthogonal, we have

$$b'^\intercal_i Qx = (Q^{-1} b'_i)^\intercal x,$$

hence

$$b'^\intercal_i Qx = b^\intercal_i x = b'^\intercal_i \sigma(x).$$

Finally, we get that for all $1 \leqslant i \leqslant n$

$$b'^\intercal_i (Qx - \sigma(x)) = 0. \tag{1}$$

Since Q and B are invertible, B' is also invertible, which means that $(b'_1, \ldots, b'_n)$ is a $\mathbb{R}$-basis of $\mathbb{R}^n$. It follows from the equality (1) that $Qx = \sigma(x)$. Since $S(\Lambda, \mathcal{B})$ contains the basis $\mathcal{B}$, the endomorphism u of $\mathbb{R}^n$ of matrix Q in the standard basis of $\mathbb{R}^n$ is an element of $\mathrm{Aut}(\Lambda)$ fully determined by σ. Q.E.D.

It is fairly easy to adapt the previous proof to the case of the isometric lattices problem:

Proposition 3.3. Let Λ and Λ' be lattices of $\mathbb{R}^n$, with bases $\mathcal{B}$ and $\mathcal{B}'$ respectively. The lattices Λ and Λ' are isometric if and only if the edge-labeled graphs $G(\Lambda, \mathcal{B})$ and $G(\Lambda', \mathcal{B})$ are isomorphic.

Example 3.4. Let us consider the Gaussian lattice $\mathbb{Z}^2 \subset \mathbb{R}^2$ with basis $I_2 := ((1,0),(0,1))$. We have

$$S(\mathbb{Z}^2, I_2) := \{(1,0),(-1,0),(0,1),(0,-1)\} = \{v_1, v_2, v_3, v_4\} \subset \mathbb{Z}^2.$$

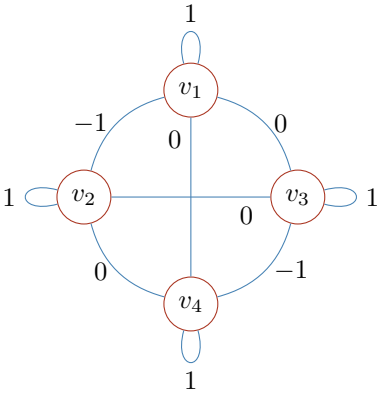
The associated edge-labeled graph $G(\mathbb{Z}^2, I_2)$ is shown on the Figure 2. Since

$$\mathrm{Aut}(\mathbb{Z}^2) = \left\langle \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \right\rangle$$

and

$$\mathrm{Aut}(G(\mathbb{Z}^2, I_2)) = \langle (v_1\ v_2)(v_3\ v_4), (v_1\ v_3)(v_2\ v_4), (v_3\ v_4) \rangle,$$

the groups $\mathrm{Aut}(\mathbb{Z}^2)$ and $\mathrm{Aut}(G(\mathbb{Z}^2, I_2))$ are isomorphic as expected.



	v_1	v_2	v_3	v_4
v_1	1	−1	0	0
v_2	1	−1	0	0
v_3	0	0	1	−1
v_4	0	0	−1	1

Figure 2: The edge-labeled graph $G(\mathbb{Z}^2, I_2)$ and its incidence matrix.

Given a finite set S, the edge-labeled graph G_S is computable in a time polynomial in the cardinality of S. Thus, the previous propositions lead us to our first reduction result. In the rest of this paper, by *elementary arithmetic operations*, we mean additions and multiplications over $\mathbb{Z}$, $\mathbb{Q}$ or $\mathbb{R}$ (note that we ignore precision issues[3]).

Theorem 3.5. LIP+S and LAP+S are polynomial-time reducible to ELGIP and ELGAP respectively. More precisely:

- Let Λ and Λ' be two lattices in $\mathbb{R}^n$ with bases $\mathcal{B}$ and $\mathcal{B}'$ respectively. Let us assume that the sets $S(\Lambda, \mathcal{B})$ and $S(\Lambda', \mathcal{B})$ are given and have the same cardinality. Then, using $O(n|S(\Lambda, \mathcal{B})|^2)$ elementary arithmetic operations, deciding whether Λ and Λ' are isometric is reducible to the problem of deciding whether two edge-labeled graphs with $|S(\Lambda, \mathcal{B})|$ vertices are isomorphic.
- If $\Lambda \subset \mathbb{R}^n$ is a lattice with basis $\mathcal{B}$, computing $\mathrm{Aut}(\Lambda)$ is reducible to the problem of computing the automorphism group of an edge-labeled graph with $|S(\Lambda, \mathcal{B})|$ vertices, and this using $O(n|S(\Lambda, \mathcal{B})|^2)$ elementary arithmetic operations.

[3]One may even limit oneself to the case of *integral lattices*, *i.e.* lattices on which the inner product takes integral values, which allows one to consider only operations in $\mathbb{Z}$.

Proof. According to the Theorem 3.3, we only need to prove that given S a finite subset of $\mathbb{R}^n$, the edge-labeled graph G_S can be computed using $O(n|S|^2)$ elementary arithmetic operations. It is clear since it naively requires to compute $\frac{|S|(|S|+1)}{2}$ inner products, each necessitating $2n-1$ elementary arithmetic operations. Q.E.D.

4 From edge-labeled graphs to vertex-labeled graphs

In this paragraph we provide details on a known method (mentioned for example in [26, §14, p. 60]) for converting an edge-labeled graph into a vertex-labeled graph while preserving its automorphism group and isometry class. Let G be an edge-labeled graph with vertices $v_1, \dots, v_N$. Let us assume that the edges are labeled by $1, 2, 3, \dots, 2^d - 1$. Let $G_\bullet$ be the vertex-labeled graph such that:

- $G_\bullet$ has Nd vertices, denoted v_i^j for $1 \leqslant i \leqslant N$ and $1 \leqslant j \leqslant d$.
- The vertex v_i^j of $G_\bullet$ has label j. Hence, $G_\bullet$ has d different labels. We may sometimes use the term *level* instead of label.
- There is an edge between v_i^j and v_k^l for $j \neq l$ if and only if $i = k$ and $j = l + 1$. These edges are said to be *vertical* (because they are between vertices on different levels).
- There is an edge between v_i^j and v_k^j if and only if there is an edge in G between v_i and v_k of label α such that the j-th bit in the binary decomposition of α is equal to 1. These edges are said be *horizontal* (because they are between vertices on the same level).

Before proving that this construction preserves automorphisms and isomorphisms, let us illustrate it by a simple example.

Example 4.1. Consider the edge-labeled graph G and the corresponding vertex-labeled graph shown on the Figure 3. The graph G has $N = 3$ vertices and we need $d = 3$ bits to encode the labels. Hence, $G_\bullet$ has 9 vertices spread among 3 levels. The label 1 is encoded as 001, thus determining edges between vertices on the level 1. For example, the edge of label 1 between v_2 and v_3 in G is converted to an edge between v_2^1 and v_3^1 in $G_\bullet$. Similarly, the label 3 is encoded as 011, thus determining edges between vertices on the levels 1 and 2. The loop on v_3 of label 3 is therefore converted to loops on v_3^1 and v_3^2.

We now prove that the passage from G to $G_\bullet$ does preserve isomorphisms and automorphisms.

Lemma 4.2. An automorphism of $G_\bullet$ is fully determined by its action on $v_1^1, \dots, v_N^1$.

Proof. Let $\sigma \in \mathrm{Aut}(G_\bullet)$. Since σ preserves the labeling of $G_\bullet$, there exists a permutation φ of $\{1, \dots, N\}$ such that $\sigma(v_i^1) = v_{\varphi(i)}^1$ for all $1 \leqslant i \leqslant N$. We show by induction on $1 \leqslant j \leqslant d$ that for all $1 \leqslant i \leqslant N$, $\sigma(v_i^j) = v_{\varphi(i)}^j$. Let $1 \leqslant i \leqslant N$ and $2 \leqslant j \leqslant d$. Let us assume that the previous equality is verified for $j-1$. There is an index $1 \leqslant k \leqslant N$ such that $\sigma(v_i^j) = v_k^j$. Since there is a vertical edge between the vertices v_i^{j-1} and v_i^j, there is also a vertical edge between $\sigma(v_i^{j-1}) = v_{\varphi(i)}^{j-1}$ and $\sigma(v_i^j) = v_k^j$. By construction of $G_\bullet$, it is possible if and only if $k = \varphi(i)$. Hence, $\sigma(v_i^j) = v_{\varphi(i)}^j$ for all $1 \leqslant i \leqslant N$ and $1 \leqslant j \leqslant d$, which proves the lemma. Q.E.D.

Proposition 4.3. The groups $\mathrm{Aut}(G)$ and $\mathrm{Aut}(G_\bullet)$ are (explicitly) isomorphic.

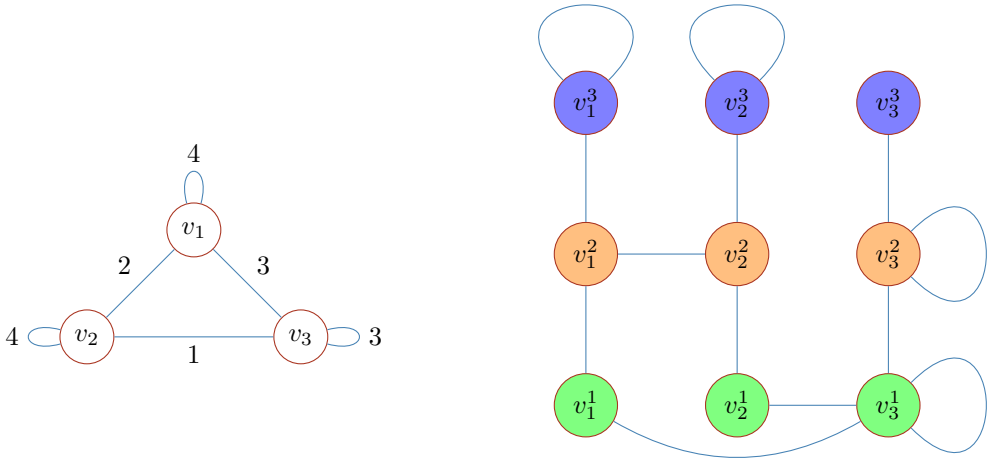


Figure 3: The edge-labeled graph G and its associated vertex-labeled graph.

Proof. Let $\sigma \in \mathrm{Aut}(G)$ and

$$\begin{array}{rccc} \widetilde{\sigma}: & G_\bullet & \longrightarrow & G_\bullet \\ & v_i^j & \longmapsto & \sigma(v_i)^j \end{array} .$$

Let us show that $\widetilde{\sigma}$ is an automorphism of $G_\bullet$. By definition, $\widetilde{\sigma}$ preserves the labeling of $G_\bullet$. Moreover, it is clear that $\widetilde{\sigma}$ preserves the vertical edges of $G_\bullet$. Let v_i^k and v_j^k be vertices of $G_\bullet$ linked by an horizontal edge. There is an edge of label α linking v_i and v_j in G such that the k-th bit in the binary decomposition of α is equal to 1. Since σ is an automorphism G, there is an edge of label α between $\sigma(v_i)$ and $\sigma(v_j)$ in G, which shows that there is an edge between $\widetilde{\sigma}(v_i^k) = \sigma(v_i)^k$ and $\widetilde{\sigma}(v_j^k) = \sigma(v_j)^k$ in $G_\bullet$. Therefore, $\widetilde{\sigma}$ is an element of $\mathrm{Aut}(G_\bullet)$.

Two distinct automorphisms of G induce distinct automorphism of $G_\bullet$ by such construction. We still need to show that every automorphism of $G_\bullet$ is of the form $\widetilde{\sigma}$ for some $\sigma \in \mathrm{Aut}(G)$. Let $\tau \in \mathrm{Aut}(G_\bullet)$ and φ be the permutation of $\{1, \ldots, N\}$ induced by the action of τ on $v_1^1, \ldots, v_N^1$. The map

$$\begin{array}{rccc} \sigma: & G & \longrightarrow & G \\ & v_i & \longmapsto & v_{\varphi(i)} \end{array}$$

is an automorphism of G (which is proved as previously) such that $\widetilde{\sigma} = \tau$. Indeed, $\widetilde{\sigma}$ and τ are equal on $v_1^1, \ldots, v_N^1$, which is enough to prove the required equality according to the Theorem 4.2. Q.E.D.

As before, it is easy to adapt the previous proof for the graphs isomorphism problem.

Proposition 4.4. Two edge-labeled graphs G and G' with the same number of vertices are isomorphic if and only if the corresponding vertex-labeled graphs $G_\bullet$ and $G'_\bullet$ are.

Note that if the edges of G are not labeled by consecutive integers, one first needs to arrange the said labels before converting it to a vertex-labeled graph using the method previously described. The vertex-labeled graph $G_\bullet$ produced depends on the ordering chosen, but its automorphism group does not. Moreover, if we want to decide whether two edge-labeled graphs are isomorphic using the associated vertex-colored graphs, we can simply chose the same ordering of the labels for the two graphs (since two isomorphic edge-labeled graphs must have the same set of labels by definition). Furthermore, the ordering chosen has an influence on the graph density. Thus, a clever ordering may lead to significant performance improvements if one aims at effective computations. This will be discussed in the next paragraph for edge-labeled graphs arising from lattices.

Example 4.5. The vertex-labeled graph $G(\mathbb{Z}^2, I_2)$ from the Theorem 3.4 has $N = 4$ vertices, and only 2 bits are necessary for encoding the labels $\{-1, 0, 1\}$. For the ordering $(-1, 0, 1)$, the binary labeling of $G(\mathbb{Z}^2, I_2)$ and the associated vertex-labeled graph $G(\mathbb{Z}^2, I_2)^1_\bullet$ are presented on the Figure 4. The Figure 5 shows the vertex-labeled graph $G(\mathbb{Z}^2, I_2)^2_\bullet$ obtained from $G(\mathbb{Z}^2, I_2)$ for the ordering $(1, -1, 0)$. The graphs $G(\mathbb{Z}^2, I_2)$ and $G(\mathbb{Z}^2, I_2)^2_\bullet$ are obviously not isomorphic (they do not have the same number of edges), but one may easily check that

$$\mathrm{Aut}(G(\mathbb{Z}^2, I_2)^1_\bullet) = \mathrm{Aut}(G(\mathbb{Z}^2, I_2)^2_\bullet) = \left\langle \begin{array}{c} (v_3\ v_4)(v_7\ v_8) \\ (v_1\ v_2)(v_5\ v_6) \\ (v_1\ v_3)(v_2\ v_4)(v_5\ v_7)(v_6\ v_8) \end{array} \right\rangle.$$

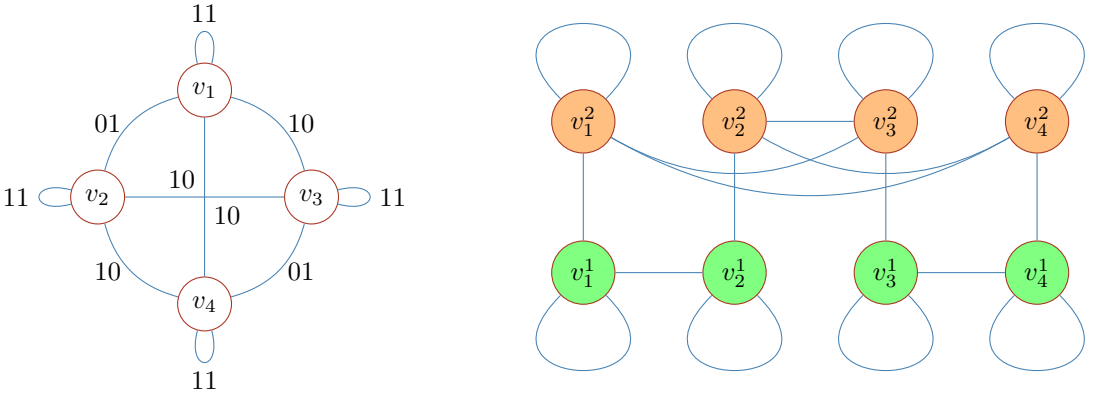


Figure 4: The graph $G(\mathbb{Z}^2, I_2)$ after the binary relabeling according to the order $(-1, 0, 1)$ and the associated vertex-labeled graph $G(\mathbb{Z}^2, I_2)^1_\bullet$.

We deduce from the previous construction that the case of edge-labeled graphs is polynomial time reducible to the case of vertex-labeled graphs when one deals with the notion of isomorphism:

Theorem 4.6. ELGIP and ELGAP are polynomial-time reducible to VLGIP and VLGAP respectively. More precisely:

- Let G and G' be two edge-labeled graphs with N vertices and labels $1, 2, \ldots, 2^d - 1$. Deciding whether G and G' are isomorphic is reducible in time $O(dN^2)$ to the problem of deciding whether two vertex-labeled graphs with Nd vertices and d labels are isomorphic.

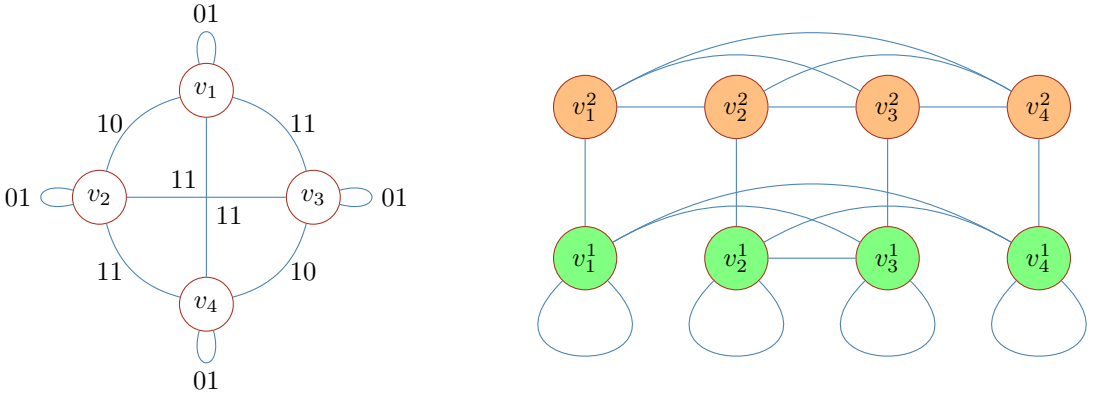


Figure 5: The graph $G(\mathbb{Z}^2, I_2)$ after the binary relabeling according to the order $(1, -1, 0)$ and the associated vertex-labeled graph $G(\mathbb{Z}^2, I_2)^2_\bullet$.

- If G is an edge-labeled graph with N vertices and labels $1, 2, \ldots, 2^d - 1$, computing $\mathrm{Aut}(G)$ is reducible in time $O(dN^2)$ to the problem of computing the automorphism group of a vertex-labeled graph with Nd vertices and d labels.

Proof. Given an edge-labeled graph with N vertices and labels $1, 2, \ldots, 2^d - 1$, the corresponding vertex-labeled graph can be naively computed in time $O(dN^2)$. Q.E.D.

Digression on the density of $G(\Lambda, \mathcal{B})_\bullet$.

As noted above, if G is an edge-labeled graph, the density of $G_\bullet$ depends on the ordering chosen on the labels of G. In fact, it is easy to define a layout of the labels that minimize the density of $G_\bullet$ by associating to the most occurring labels the binary words with minimal Hamming weight. Note that it requires to order the labels by occurrences, which may become costly when the size of the graph increases, especially when one wants to compute the vertex-labeled graph associated to a lattice.

Nevertheless, we prove in this paragraph that if $\Lambda \subset \mathbb{R}^n$ is a lattice with basis $\mathcal{B}$, the vertex-labeled graph $G(\Lambda, \mathcal{B})_\bullet$ remains quite sparse regardless of the ordering chosen. We start with the case of a pair $(\Lambda, \mathcal{B})$ associated to an elementary quadratic form.

Lemma 4.7. Let $\mathcal{B} := (b_1, \ldots, b_n)$ be a $\mathbb{R}$-basis of $\mathbb{R}^n$ whose Gram matrix is αI_n for some $\alpha \in \mathbb{R}_{>0}$. Let Λ be the lattice generated by $\mathcal{B}$. The only labels of $G(\Lambda, \mathcal{B})$ are 0 and $\pm\alpha$, and the density of $G(\Lambda, \mathcal{B})_\bullet$ is independent of Λ and $\mathcal{B}$:

$$\rho(G(\Lambda, \mathcal{B})_\bullet) = \begin{cases} \dfrac{2n+3}{8n+2} & \text{if the labels ordering is } (\alpha, -\alpha, 0) \text{ or } (-\alpha, \alpha, 0), \\ \dfrac{n+6}{8n+2} & \text{if the labels ordering is } (-\alpha, 0, \alpha) \text{ or } (0, -\alpha, \alpha), \\ \dfrac{n+5}{8n+2} & \text{if the labels ordering is } (\alpha, 0, -\alpha) \text{ or } (0, \alpha, -\alpha). \end{cases} \tag{2}$$

Proof. Since the Gram matrix of $\mathcal{B}$ is αI_n, we have that $S(\Lambda, \mathcal{B}) = \{\pm b_1, \ldots, \pm b_n\}$ (hence $G(\Lambda, \mathcal{B})$ has $2n$ vertices), and it follows that the labels of $G(\Lambda, \mathcal{B})$ are 0 and $\pm\alpha$. Let us enumerate the edges of $G(\Lambda, \mathcal{B})$:

- There is a loop with label α on every vertex of $G(\Lambda, \mathcal{B})$, for a total of $2n$ loops with label α.
- There is an edge with label $-\alpha$ between b_i and $-b_i$ for $1 \leqslant i \leqslant n$, for a total of n edges with label $-\alpha$.
- There is an edge with label 0 between b_i and $\pm b_j$ for $1 \leqslant i < j \leqslant n$, for a total of $n(n-1)$ edges with label 0.

Since $G(\Lambda, \mathcal{B})$ has $2n$ vertices and $3 = 2^2 - 1$ labels, $G(\Lambda, \mathcal{B})_\bullet$ has $4n$ vertices. If the labels are ordered as $\pm(\alpha, -\alpha, 0)$, $G(\Lambda, \mathcal{B})_\bullet$ has $2n$ loops, $2n$ vertical edges, n edges induced by the label $-\alpha$ and $2n(n-1)$ edges induced by the label 0. Therefore, its density is

$$\rho(G(\Lambda, \mathcal{B})_\bullet) = \frac{2(2n + 2n + n + 2n(n-1))}{4n(4n+1)} = \frac{2n+3}{8n+2}.$$

Similar computations lead to the announced result for other orderings. Q.E.D.

Proposition 4.8. Let $n \geqslant 2$ and $\Lambda \subset \mathbb{R}^n$ be a lattice with basis $\mathcal{B} := (b_1, \ldots, b_n)$. The density of $G(\Lambda, \mathcal{B})_\bullet$ is bounded independently of the ordering the labels of $G(\Lambda, \mathcal{B})$ and of the pair $(\Lambda, \mathcal{B})$:

$$\rho(G(\Lambda, \mathcal{B})_\bullet) \leqslant \frac{2}{3} + \frac{1}{3n}. \tag{3}$$

Proof. Since the equality (2) induces the inequality (3) (one may easily reduce the comparison between these bounds to the study of a pair of quadratic inequalities), we can assume that there does not exist $\alpha \in \mathbb{R}$ such that the Gram matrix of $\mathcal{B}$ is equal to αI_n.

Let d be the number of bits required to encode the labels of $G(\Lambda, \mathcal{B})$ and $s := |S(\Lambda, \mathcal{B})|$. By definition, the graph $G(\Lambda, \mathcal{B})_\bullet$ has sd vertices and $s(d-1)$ vertical edges. A vertex in $G(\Lambda, \mathcal{B})_\bullet$ is concerned by at most s horizontal edges (including a loop), so we can roughly bound the density of $G(\Lambda, \mathcal{B})_\bullet$ by

$$\frac{2(s^2 d + s(d-1))}{sd(sd+1)} \leqslant \frac{2}{d} + \frac{2}{sd}.$$

Since $S(\Lambda, \mathcal{B})$ contains $\pm b_1, \ldots, \pm b_n$, we have $s \geqslant 2n$. To conclude, it remains to show that $d \geqslant 3$. Let us assume that $d = 2$ (note that $d \geqslant 2$ obviously), that is to say that $G(\Lambda, \mathcal{B})$ contains only 2 or 3 distinct labels. Let $\pm\alpha$ with $\alpha \in \mathbb{R}_{>0}$ be the two non-zero labels of $G(\Lambda, \mathcal{B})$. Note that the hypothetical third label of $G(\Lambda, \mathcal{B})$ is necessarily 0. We have $\|b_i\|^2 = \alpha$ for all $1 \leqslant i \leqslant n$. By the assumption made on the Gram matrix of $\mathcal{B}$, there exists $1 \leqslant i < j \leqslant n$ such that $\langle b_i \,|\, b_j \rangle \neq 0$. Since there are only 2 or 3 labels in $G(\Lambda, \mathcal{B})$, we have $\langle b_i \,|\, b_j \rangle = \alpha$ (up to replacing b_j by $-b_j$). Thus

$$\|b_i - b_j\|^2 = -2\langle b_i \,|\, b_j \rangle + \|b_i\|^2 + \|b_j\|^2 = -2\alpha + \alpha + \alpha = 0,$$

which is a contradiction. Therefore, we have $d \geqslant 3$ and the announced inequality follows. Q.E.D.

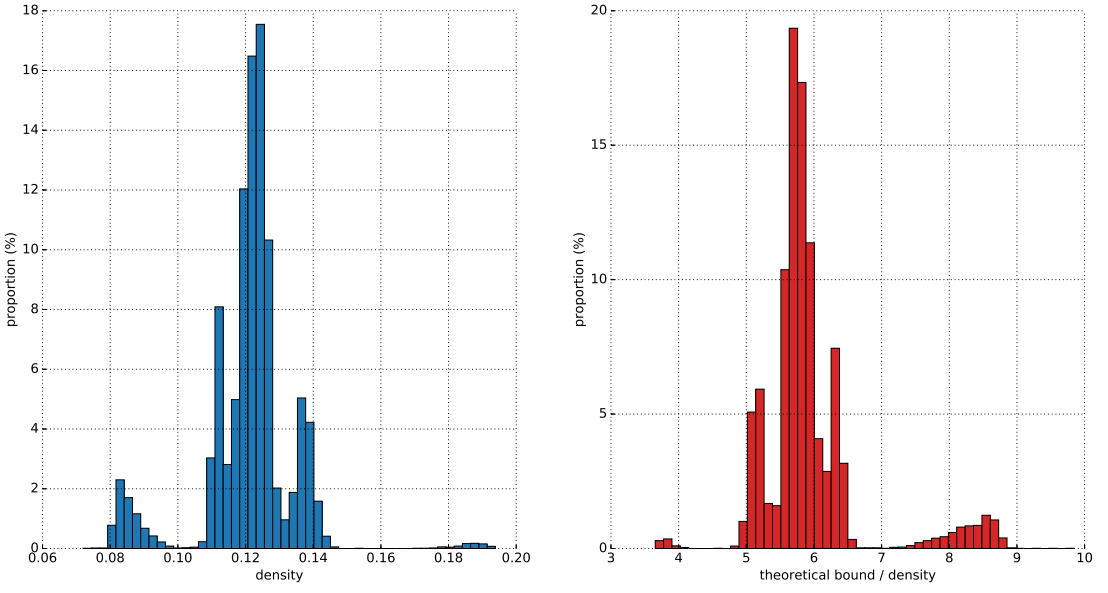


Figure 6: Density of the vertex-labeled graphs associated to the 10916 perfect lattices of dimension 8 and comparison with the theoretical bound (3).

Note that the bound (3) is usually quite far from the real density of the graphs considered, and it is very unlikely to be an optimal bound. For example, let us consider the bases of the 10916 perfect lattices of dimension 8 given in the catalogue of lattices maintained by Nebe and Sloane [27]. The left-hand panel of the Figure 6 shows the distribution of the density of the vertex-labeled graphs associated to these lattices. The graphs have been computed using the labels ordering associated to the usual order on $\mathbb{R}$, so the densities presented are not optimal in general. Nevertheless, the maximum density obtained for these graphs does not exceed 0.20. The right-hand panel of the Figure 6 shows the distribution of the ratio between the theoretical bound (3) and the actual density obtained, which never falls behind 3.5.

5 From vertex-labeled graphs to graphs

Most of the results concerning algorithmic complexity of graphs are formalized for graphs that are not vertex-labeled. Hence, we give in this paragraph some details on a known method (quite similar to the one presented in the previous section) which allows to convert a vertex-labeled graph to a graph with no vertex labelling while preserving isomorphisms. Since vertex-labeled graphs are sometimes called *coloured graphs*, we will refer to this procedure as the *decolourisation* of a (vertex-labeled) graph.

Given $n \in \mathbb{N}$ whose binary decomposition is $n = \sum_{i=0}^{d} b_i 2^i$ with $b_i \in \{0, 1\}$, let T_n be the binary tree of height $d + 1$ such that for all $0 \leqslant i \leqslant d$, a vertex of height i has 1 children if $b_i = 0$ and 2 children if $b_i = 1$. Several examples of such trees are presented on the Figure 7.

Let G be a vertex-labeled graph. Similarly to the case of edge-labeled graphs, we assume that

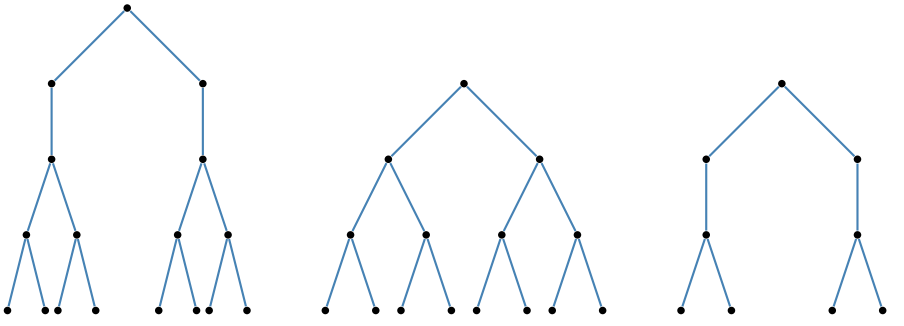


Figure 7: The binary trees T_{13}, T_7, and T_6.

labels of G are $1, 2, \ldots, 2^d - 1$. Let $G_\circ$ be the graph obtained from G by rooting at each vertex of G of label i the binary tree T_i. The Figure 8 presents such a graph $G_\circ$ obtained from a graph G with 4 vertices labeled by $\{1, 2, 3\}$.

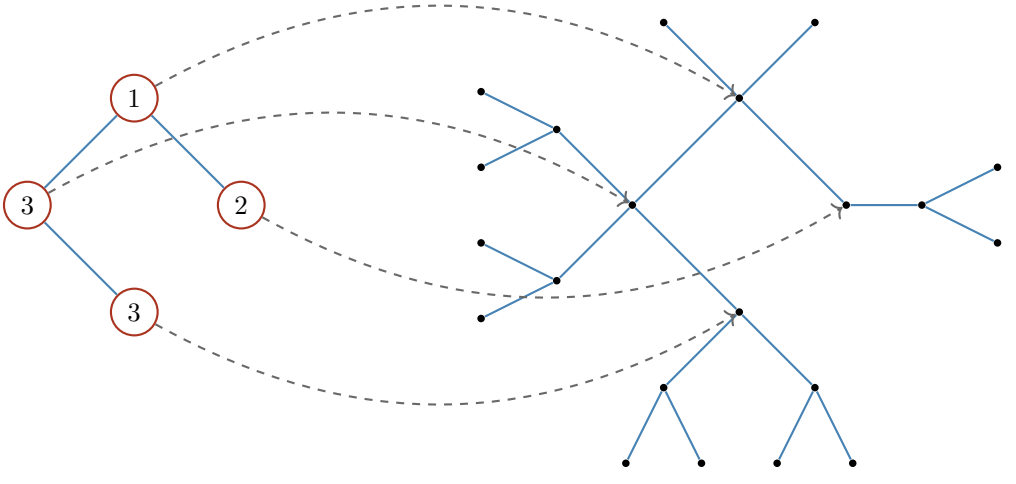


Figure 8: Example of decolourisation of a graph.

By induction on the height, one may easily prove that the decolourisation process does preserve isomorphisms. See [29, thm.1, p.21] for a detailed proof of this result.

Proposition 5.1. Let G and H be two vertex-labeled graphs with the same number of vertices and the same set of labels.

- The groups $\mathrm{Aut}(G)$ and $\mathrm{Aut}(G_\circ)$ are (explicitly) isomorphic.
- G and H are isomorphic (as vertex-labeled graphs) if and only if $G_\circ$ et $H_\circ$ are isomorphic (as graphs).

In order to establish a precise polynomial-time reduction from VLGIP to GIP, it remains to prove that the graph $G_\circ$ is easily computable from G and that it is not much larger than $G_\circ$.

Theorem 5.2. VLGIP and VLGAP are polynomial-time reducible to GIP and GAP respectively. More precisely:

- Let G and H be two vertex-labeled graphs with N vertices and with the same m labels. Let $d := \lfloor \log_2(m) \rfloor + 1$. Deciding if G and H are isomorphic is reducible in time $O\left(dN^2\right)$ to deciding if two graphs with $O\left(N(m-1)\right)$ vertices are isomorphic.
- Let G be a vertex-labeled graph with N vertices and m labels. Let $d := \lfloor \log_2(m) \rfloor + 1$. Computing $\mathrm{Aut}(G)$ is reducible in time $O(dN^2)$ to computing the automorphism group of a graph with $O\left(N(m-1)\right)$ vertices.

Proof. The naive computation of $G_\circ$ from G has the announced complexity. Hence, it remains to prove that if G is a vertex-labeled graph with N vertices and m labels, the graph $G_\circ$ has $O\left(N(m-1)\right)$ vertices. The graph $G_\circ$ is built from G by replacing each vertex of G by a binary tree of height at most $d+1$. Such a tree has at most $2^{d+1}-1$ vertices (taking into account its root, which is a vertex of the initial graph). The number of vertices of $G_\circ$ is therefore bounded by

$$N(2^{d+1}-1) \leqslant N(2^{\log_2(m)+2}-1) = N(4m-1),$$

which concludes the proof. Q.E.D.

It is worth noting that this reduction is also polynomial *in the number N of vertices of G*, since there can be at most N distinct labels in G.

Digression on the density of $G(\Lambda, \mathcal{B})_{\bullet\circ}$.
In the previous section, we proved that for any lattice Λ and any of its bases $\mathcal{B}$, the density of the vertex-labeled graph $G(\Lambda, \mathcal{B})_\bullet$ can be bounded from above by a quantity depending only on the dimension of Λ, thus establishing that these graphs remain quite sparse. In this section, we prove that this property in preserved by the decolourisation process.

Lemma 5.3. Let $n \in \mathbb{N}$ whose binary decomposition is $n = \sum_{i=0}^{d} b_i 2^i$ with $b_i \in \{0,1\}$ for all $0 \leqslant i \leqslant d$. Let

$$t_n := \sum_{i=0}^{d} \prod_{j=0}^{i-1} (b_j + 1).$$

The binary tree T_n has t_n vertices and $t_n - 1$ edges.

Proof. The result on the number of edges is a direct consequence from the one on vertices. For all $0 \leqslant i \leqslant d$, let v_i be the number of vertices at the height i in T_n. By construction, $v_0 = 1$ and $v_{i+1} = (b_i + 1)v_i$ for all $0 \leqslant i < d$. An easy induction shows that

$$v_i = \prod_{j=0}^{i-1} (b_j + 1)$$

for all $0 \leqslant i \leqslant d$. Noting that the number of vertices of T_n is $\sum_{i=0}^{d} v_i$ concludes the proof. Q.E.D.

First, let us prove that $G_\circ$ is always sparser than G.

Proposition 5.4. Let $G_\circ$ be the graph obtained by the decolourisation of a vertex-labeled graph G. Then $\rho(G_\circ) < \rho(G)$. More precisely, let V and E be respectively the sets of vertices and edges of G. Without loss of generality, let us assume that G is labeled by a function $e_G : V \longrightarrow \mathbb{N}$. Then

$$\rho(G_\circ) = \frac{2(|E| + t)}{(|V| + t)(|V| + t + 1)},$$

with

$$t := \sum_{v \in V} \left(t_{e_G(v)} - 1 \right),$$

where $t_{e_G(v)}$ is the constant defined in the Theorem 5.3.

Proof. Let $V_\circ$ and $E_\circ$ be respectively the sets of vertices and edges of $G_\circ$. The graph $G_\circ$ is built from G by replacing each vertex $v \in V$ of G by the binary tree $T_{e_G(v)}$. According to the Theorem 5.3, this tree has $t_{e_G(v)}$ vertices and $t_{e_G(v)} - 1$ edges. Thus

$$|V_\circ| = \sum_{v \in V} t_{e_G(v)} = |V| + \sum_{v \in V} \left(t_{e_G(v)} - 1 \right)$$

and, since the edges of G are preserved by the decolourisation process:

$$|E_\circ| = |E| + \sum_{v \in V} \left(t_{e_G(v)} - 1 \right).$$

Hence, the density of $G_\circ$ is given by

$$\rho(G_\circ) = \frac{2|E_\circ|}{|V_\circ|(|V_\circ| + 1)} = \frac{2(|E| + t)}{(|V| + t)(|V| + t + 1)}.$$

Let us recall that the density of G is defined as

$$\rho(G) = \frac{2|E|}{|V|(|V| + 1)}.$$

The function $x \mapsto \frac{2(|E|+x)}{(|V|+x)(|V|+x+1)}$ being strictly decreasing on $\mathbb{R}_{\geqslant 0}$, the inequality $\rho(G_\circ) < \rho(G)$ is proved. Q.E.D.

It now possible to establish the sparsity of the graphs $G(\Lambda, \mathcal{B})_{\bullet\circ}$ obtained by conversion to vertex-labeled graph and decolourisation from the edge-labeled graphs $G(\Lambda, \mathcal{B})$.

Corollary 5.5. Let Λ be a n-dimensional lattice with basis $\mathcal{B} := (b_1, \ldots, b_n)$. The density of $G(\Lambda, \mathcal{B})_{\bullet\circ}$ is bounded from above, and this bound depends only on the dimension of Λ:

$$\rho\left(G(\Lambda, \mathcal{B})_{\bullet\circ}\right) < \frac{2n + 3}{6n + 1} = \frac{4}{13} + o\left(\frac{1}{n}\right).$$

Proof. Follows from the Theorem 4.8 and the Theorem 5.4. Q.E.D.

6 Polynomial reduction from LIP+S to GIP

Using the well-known reduction from vertex-labeled graphs to graphs (regarding the isomorphism problem), we finally prove in this paragraph that the weakened versions of LIP and LAP are reducible to the equivalent problems for graphs. Nevertheless, note that popular programs like `Nauty/Traces` [26] handle vertex-labeled graphs: for practical computations of automorphisms and isometries of a lattice Λ with basis $\mathcal{B}$, one does not need to convert the vertex-labeled graph $G(\Lambda, \mathcal{B})$ to its non-labeled counterpart. Hence, we present as a first step the reduction from LIP to VLGIP. In the following, we assume that elementary arithmetic operations (*i.e.* additions and multiplications over $\mathbb{Z}$, $\mathbb{Q}$, or $\mathbb{R}$) are made in time $O(1)$. If $X := (x_1, \ldots, x_k)$ is a family of non-zero elements of $\mathbb{R}^n$, let

$$\|X\|_\infty := \max_{1 \leqslant i \leqslant k} \|x_i\|,$$

$$h(X) := \lfloor \log_2(2\|X\|_\infty^2 + 1) \rfloor + 1,$$

and

$$h_2(X) := \lfloor \log_2(h(X)) \rfloor + 1.$$

Theorem 6.1. LIP+S and LAP+S are polynomial-time reducible to VLGIP and VLGAP respectively. More precisely:

- Let Λ and Λ' be two lattices in $\mathbb{R}^n$ with basis $\mathcal{B}$ and $\mathcal{B}'$ respectively. Let us assume that the sets $S(\Lambda, \mathcal{B})$ and $S(\Lambda', \mathcal{B})$ are given and have the same cardinality. Deciding whether Λ and Λ' are isometric is reducible in time

  $$O\left((n + h(\mathcal{B}))|S(\Lambda, \mathcal{B})|^2\right)$$

  to the problem of deciding whether two vertex-labeled graphs with at most $h(\mathcal{B})|S(\Lambda, \mathcal{B})|$ vertices and at most $h(\mathcal{B})$ labels are isomorphic.

- If $\Lambda \subset \mathbb{R}^n$ is a lattice with basis $\mathcal{B}$, computing $\mathrm{Aut}(\Lambda)$ is reducible in time

  $$O\left((n + h(\mathcal{B}))|S(\Lambda, \mathcal{B})|^2\right)$$

  to the problem of computing the automorphism group of a vertex-labeled graph with at most $h(\mathcal{B})|S(\Lambda, \mathcal{B})|$ vertices and at most $h(\mathcal{B})$ labels.

Proof. We detail the proof for the reduction from LAP+S to VLGAP. The same argument used with the first part of the Theorem 3.5 allows to prove the reduction from LIP+S to VLGIP.

We know from the Theorem 3.5 that computing $\mathrm{Aut}(\Lambda)$ is reducible in time $O(n|S(\Lambda, \mathcal{B})|^2)$ to computing $\mathrm{Aut}(G(\Lambda, \mathcal{B}))$, where $G(\Lambda, \mathcal{B})$ is an edge-labeled graph with $|S(\Lambda, \mathcal{B})|$ vertices. Let d be the number of bits required to binary encode the labels of $G(\Lambda, \mathcal{B})$. Using the Cauchy-Schwarz inequality, one can prove that there are at most $2\|\mathcal{B}\|_\infty^2 + 1$ different labels in $G(\Lambda, \mathcal{B})$, which leads to

$$d \leqslant \lfloor \log_2(2\|\mathcal{B}\|_\infty^2 + 1) \rfloor + 1 = h(\mathcal{B}).$$

Finally, Theorem 4.6 states that determining $\mathrm{Aut}(G(\Lambda, \mathcal{B}))$ is reducible in time $O(h(\mathcal{B})|S(\Lambda, \mathcal{B})|^2)$ to determining the automorphism group of $G(\Lambda, \mathcal{B})_\bullet$, a vertex-labeled graph with at most $h(\mathcal{B})|S(\Lambda, \mathcal{B})|$ vertices spread on at most $h(\mathcal{B})$ levels. Q.E.D.

The reduction from LIP+S to GIP follows easily from the previous theorem and the reduction from VLGIP to GIP established in the previous section.

Corollary 6.2. LIP+S and LAP+S are polynomial-time reducible to GIP and GAP respectively. More precisely:

- Let Λ and Λ' be two n-dimensional lattices with basis $\mathcal{B}$ and $\mathcal{B}'$ respectively. Let us assume that the sets $S(\Lambda, \mathcal{B})$ and $S(\Lambda', \mathcal{B})$ are given and have the same cardinality. Then, using

$$O\left((n + h(\mathcal{B})h_2(\mathcal{B}))|S(\Lambda, \mathcal{B})|^2\right)$$

  elementary arithmetic operations, deciding whether Λ and Λ' are isometric is reducible to the problem of deciding whether two graphs with

$$O\left(h(\mathcal{B})^2|S(\Lambda, \mathcal{B})|\right)$$

  vertices are isomorphic.

- If Λ is a n-dimensional lattice with basis $\mathcal{B}$, computing $\mathrm{Aut}(\Lambda)$ is reducible to the problem of computing the automorphism group of an edge-labeled graph with

$$O\left(h(\mathcal{B})^2|S(\Lambda, \mathcal{B})|\right)$$

  vertices, and this using

$$O\left((n + h(\mathcal{B})h_2(\mathcal{B}))|S(\Lambda, \mathcal{B})|^2\right)$$

  elementary arithmetic operations.

Proof. As before, we focus our attention on LAP+S and GAP, leaving the details to the reader for LIP+S and GIP. The previous theorem states that computing $\mathrm{Aut}(\Lambda)$ is reducible in time

$$O\left((n + h(\mathcal{B}))|S(\Lambda, \mathcal{B})|^2\right)$$

to computing the automorphism group of $G(\Lambda, \mathcal{B})_\bullet$, a vertex-labeled graph with at most $h(\mathcal{B})|S(\Lambda, \mathcal{B})|$ vertices spread on at most $h(\mathcal{B})$ levels. But we know from the Theorem 5.2 that computing $\mathrm{Aut}(G(\Lambda, \mathcal{B})_\bullet)$ is reducible in time

$$O\left(dh(\mathcal{B})^2|S(\Lambda, \mathcal{B})|^2\right)$$

with $d = \lfloor \log_2(h(\mathcal{B})) + 1 \rfloor + 1 = h_2(\mathcal{B})$ to computing the automorphism group of $G(\Lambda, \mathcal{B})_{\bullet\circ}$, a graph with

$$O\left(h(\mathcal{B})|S(\Lambda, \mathcal{B})|(h(\mathcal{B}) - 1)\right) = O\left(h(\mathcal{B})^2|S(\Lambda, \mathcal{B})|\right)$$

vertices. Combining these two complexity bounds proves the announced result. Q.E.D.

7 Estimating $|S(\Lambda, \mathcal{B})|$

Let Λ be a n-dimensional lattice with basis $\mathcal{B}$. By $m(\Lambda)$ we denote the minimum of Λ, and let $s(\Lambda) := \frac{m(\Lambda)}{2}$. For all $x \in \mathbb{R}^n$ and all $R > 0$, let $B_n(x, R)$ be the n-dimensional closed ball of center x and radius R, $\mathring{B}_n(x, R)$ be its interior and $\partial B_n(x, R)$ be its boundary (*i.e.* the n-dimensional sphere of center x and radius R). If μ_n denotes the n-dimensional Lebesgue measure, let us recall (see [11, p. 9] for instance) that

$$\mu_n(B_n(x, R)) = \mu_n(\mathring{B}_n(x, R)) = \frac{\pi^{n/2} R^n}{\Gamma\left(\frac{n}{2} + 1\right)} \tag{4}$$

and

$$\mu_{n-1}(\partial B_n(x, R)) = \frac{2\pi^{n/2} R^{n-1}}{\Gamma\left(\frac{n}{2}\right)}, \tag{5}$$

where Γ is the *Euler gamma function*, defined for all $z \in \mathbb{C}$ such that $\Re(z) > 0$ by

$$\Gamma(z) := \int_0^{+\infty} t^{z-1} e^{-t} dt.$$

In order to estimate the quantity $|S(\Lambda, \mathcal{B})|$, two approaches can be highlighted:

- Using the inequality
$$|S(\Lambda, \mathcal{B})| \leqslant \sum_{i=1}^{n} |\Lambda \cap \partial B_n(0, \|b_i\|)|,$$
one may reduce the problem to estimating the size of $\Lambda \cap \partial B_n(0, R)$ for various $R \geqslant 0$. Achieving accurate and explicit estimations for such quantities is not an easy task. As an example, for the lattice $\mathbb{Z}^n$ it is famously known as the problem of *representing integers as sum of squares*. See [16] for an extensive survey on this topic.

- On the other hand, with the help of the inclusion $S(\Lambda, \mathcal{B}) \subset \Lambda \cap B_n(0, \|\mathcal{B}\|_\infty)$, it can be reduced to estimating $|\Lambda \cap B_n(0, R)|$ for various $R > 0$. As before, it is a clunky task even for lattices as simple as $\mathbb{Z}^n$: for $n = 3$, it is the *sphere problem*, which is still a challenging question in modern number theory (for instance see [21] or [19]).

In this paragraph, we propose several estimations of $|\Lambda \cap B_n(0, R)|$ and $|\Lambda \cap \partial B_n(0, R)|$ for Λ an arbitrary n-dimensional lattice and any $R > 0$. Note that different bounds can be obtained if one restricts to particular families of lattices and/or to some values of R (*e.g.* using the Gaussian heuristic [17, §5]). First, let us evaluate $|\Lambda \cap B_n(0, R)|$ using a naive geometric method

Proposition 7.1. For all $R \geqslant 0$ we have

$$|\Lambda \cap B_n(0, R)| \leqslant \left(\frac{R}{s(\Lambda)} + 1\right)^n. \tag{6}$$

In particular, if $\mathcal{B}$ is a basis of Λ then

$$|S(\Lambda, \mathcal{B})| \leqslant \left(\frac{\|\mathcal{B}\|_\infty}{s(\Lambda)} + 1\right)^n. \tag{7}$$

Proof. For all $x \in \Lambda \cap B_n(0,R)$, the ball $\mathring{B}_n(x, s(\Lambda))$ is contained in $B_n(0, R + s(\lambda))$. Moreover, for all $x, y \in \Lambda \cap B_n(0,R)$ distinct, we have

$$\mathring{B}_n(x, s(\Lambda)) \cap \mathring{B}_n(y, s(\Lambda)) = \varnothing$$

by definition of $s(\Lambda)$. Hence:

$$\begin{aligned}\mu_n(B_n(0, R + s(\Lambda))) &\geqslant \sum_{x \in \Lambda \cap B_n(0,R)} \mu_n(\mathring{B}_n(x, s(\Lambda))) \\ &= |\Lambda \cap B_n(0,R)| \cdot \mu_n(B_n(0, s(\Lambda))).\end{aligned}$$

Using (4), the announced inequality is proven. Q.E.D.

For $R = m(\Lambda)$, the bound (6) gives the estimation $|S(\Lambda)| \leqslant 3^n - 1$ of the number of shortest vectors of Λ. Following the algebraic proof of [31, p.107–108], we can establish a better bound for $|S(\Lambda)|$.

Proposition 7.2. The number of shortest vectors of Λ is bounded by

$$|S(\Lambda)| \leqslant 2^{n+1} - 2. \tag{8}$$

Proof. Let $x, y \in S(\Lambda)$, and let us assume that there exists $t \in \Lambda$ such that $y = x + 2t$. Then

$$\|y\|^2 = \|x + 2t\|^2 = \|x\|^2 + 4\|t\|^2 + 4\langle x \,|\, t\rangle,$$

which leads to

$$\|x + t\|^2 + \|t\|^2 = \|x\|^2,$$

and therefore $\|x + t\|^2 \leqslant \|x\|^2$. By definition of $m(\Lambda)$, it is possible if and only if $x = -t$, that is to say if $y = -x$. Hence, two elements of $S(\Lambda)$ equal in $\Lambda/2\Lambda$ differ only by sign. Since $S(\Lambda) \cap 2\Lambda = \varnothing$ and $|\Lambda/2\Lambda| = 2^n$, we have $|S(\Lambda)| \leqslant 2(2^n - 1)$. Q.E.D.

We now turn our attention to estimating $|\Lambda \cap \partial B_n(0,R)|$.

Proposition 7.3. Let $R \geqslant 0$. We have

$$|\Lambda \cap \partial B_n(0,R)| \leqslant \frac{2}{I\left(\eta_R; \frac{n-1}{2}, \frac{1}{2}\right)}, \tag{9}$$

where:

- $I(z; a, b)$ is the regularized incomplete Euler beta function, defined for all $z \in \mathbb{R}$ and all $a, b \in \mathbb{C}$ such that $\Re(a) > 0$ and $\Re(b) > 0$ as

$$I(z; a, b) := \frac{\int_0^z t^{a-1}(1-t)^{b-1}dt}{\int_0^1 t^{a-1}(1-t)^{b-1}dt}.$$

- $\eta_R = \sin(\vartheta_R)^2$, where ϑ_R is the colatitude angle of the hyperspherical cap $B_n(x, s(\Lambda)) \cap \partial B_n(0,R)$ with $x \in \partial B_n(0,R)$, given by

$$\vartheta_R = \arccos\left(1 - \frac{s(\Lambda)^2}{2R^2}\right).$$

Proof. Using an argument similar to the one used to prove the inequality (6), we have

$$|\Lambda \cap \partial B_n(0,R)| \leqslant \frac{\mu_{n-1}(\partial B_n(0,R))}{\mu_{n-1}(B_n(x,s(\Lambda)) \cap \partial B_n(0,R))}, \tag{10}$$

where x is any point in $\partial B_n(0,R)$. Since $y \in \partial B_n(0,R)$ is an element of $B_n(x,s(\Lambda))$ if and only if $s(\Lambda)^2 \geqslant \|x-y\|^2 = \|x\|^2 + \|y\|^2 + 2\langle x \,|\, y\rangle = 2R^2 + 2\langle x \,|\, y\rangle$, we have

$$B_n(x,s(\Lambda)) \cap \partial B_n(0,R) = \left\{ y \in \partial B_n(0,R) \;:\; \frac{\langle x \,|\, y\rangle}{\|x\|\|y\|} \geqslant \cos(\vartheta_R) \right\}.$$

Hence $B_n(x,s(\Lambda)) \cap \partial B_n(0,R)$ is an hyperspherical cap of colatitude ϑ_R. It has been proven in [23, p.68] that the measure of such a cap is

$$\mu_{n-1}(B_n(x,s(\Lambda)) \cap \partial B_n(0,R)) = \frac{1}{2}\mu_{n-1}(\partial B_n(0,R)) I\left(\eta_R; \frac{n-1}{2}, \frac{1}{2}\right).$$

By injecting this equality in (10), the result is established. Q.E.D.

Corollary 7.4. If $\mathcal{B}$ is a basis of Λ, we have

$$|S(\Lambda,\mathcal{B})| \leqslant \frac{2n}{I\left(\eta_{\|\mathcal{B}\|_\infty}; \frac{n-1}{2}, \frac{1}{2}\right)}. \tag{11}$$

Proof. The function $R \mapsto I(\eta_R; \frac{n-1}{2}, \frac{1}{2})$ being decreasing on $\mathbb{R}_{\geqslant m(\Lambda)}$ for all $n \in \mathbb{N}_{>0}$, according to (9) we have

$$\begin{aligned} |S(\Lambda,\mathcal{B})| &\leqslant \sum_{i=1}^{n} |\Lambda \cap \partial B_n(0,\|b_i\|)| \\ &\leqslant \sum_{i=1}^{n} \frac{2}{I\left(\eta_{\|b_i\|}; \frac{n-1}{2}, \frac{1}{2}\right)} \\ &\leqslant \frac{2n}{I\left(\eta_{\|\mathcal{B}\|_\infty}; \frac{n-1}{2}, \frac{1}{2}\right)}, \end{aligned}$$

which proves the result. Q.E.D.

The fact that the bound (11) is not fully explicit (mainly because it includes an Euler integral) makes it tricky to compare to the bound (7). Nevertheless, the experimental results presented on the Figure 9 allow to draw some conclusion:

- The estimation (9) of $|S(\Lambda)|$ (*i.e.* for $R = m(\Lambda)$) is close to one provided by (8), while having the advantage to be valid for all $R \geqslant 0$. However, the bound (6) for $R = m(\Lambda)$ is far less accurate: the ratio between (6) and (9) for $n = 100$ and $R = m(\Lambda)$ is greater than 10^{20}. Nonetheless, note that the all the bounds obtained are most likely not optimal.

- The experiments conducted tend to show that the bound (11) is more accurate than the (7) one, and this independently of the dimension of the lattice considered and of the ratio $\|\mathcal{B}\|_\infty / m(\Lambda)$.

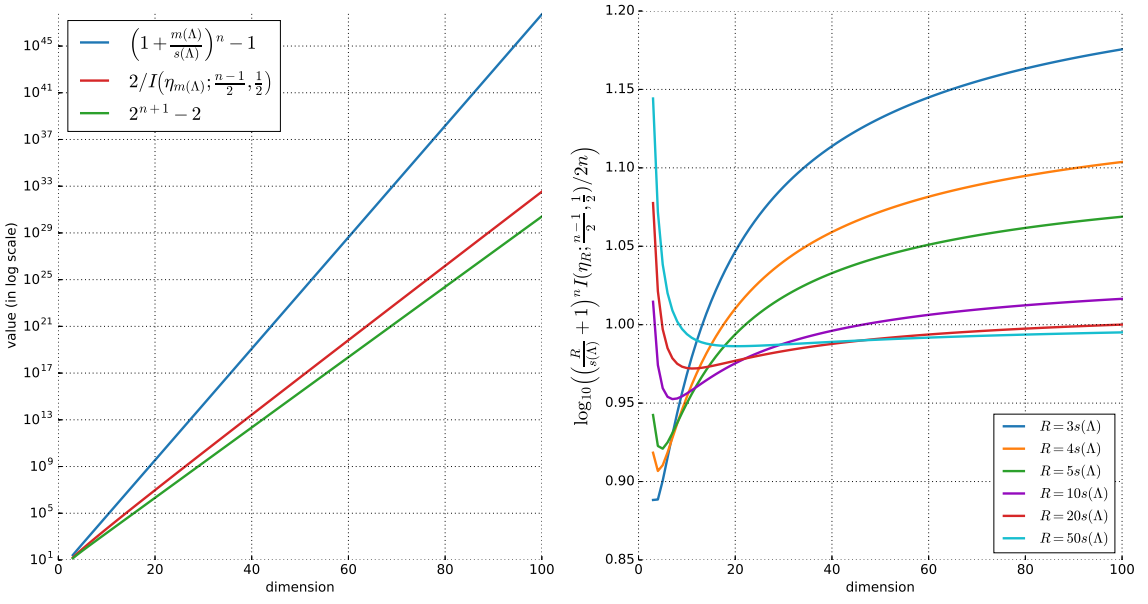


Figure 9: Comparison of the bounds established on $|S(\Lambda, \mathcal{B})|$. Left: the bounds (6) and (9) (for $R = m(\Lambda)$), together with the bound (8). Right: ratio of (7) over (11) for several values of $R = \|\mathcal{B}\|_\infty$.

Bibliography

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

[2] M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 10–19. ACM, 1998.

[3] G. Alagic, G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, et al. *Status report on the first round of the NIST post-quantum cryptography standardization process.* US Department of Commerce, National Institute of Standards and Technology, 2019.

[4] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, J. Kelsey, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, et al. Status report on the second round of the NIST post-quantum cryptography standardization process. *US Department of Commerce, National Institute of Standards and Technology*, 2020.

[5] S. Arora and B. Barak. *Computational complexity: a modern approach.* Cambridge University Press, 2009.

[6] L. Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697, 2016.

[7] L. Babai and E. M. Luks. Canonical labeling of graphs. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 171–183. ACM, 1983.

[8] S. Buss, Y. Chen, J. Flum, S.-D. Friedman, and M. Müller. Strong isomorphism reductions in complexity theory. *The Journal of Symbolic Logic*, 76(04):1381–1402, 2011.

[9] T. Camus. *Méthodes algorithmiques pour les réseaux algébriques.(Algorithmic methods for algebraic lattices)*. PhD thesis, Grenoble Alpes University, France, 2017. Available at https://tel.archives-ouvertes.fr/tel-01563081.

[10] D. X. Charles. Counting lattice vectors. *Journal of Computer and System Sciences*, 73(6):962–972, 2007.

[11] J. H. Conway and N. J. A. Sloane. *Sphere packings, lattices and groups*, volume 290. Springer Science & Business Media, 2013.

[12] M. Dutour Sikirić, A. Schürmann, and F. Vallentin. Complexity and algorithms for computing voronoi cells of lattices. *Mathematics of computation*, 78(267):1713–1731, 2009.

[13] P. Elbaz-Vincent, H. Gangl, and C. Soulé. Quelques calculs de la cohomologie de $\mathrm{GL}_N(\mathbb{Z})$ et de la K-théorie de $\mathbb{Z}$. *Comptes Rendus Mathematique*, 335(4):321–324, 2002.

[14] P. Elbaz-Vincent, H. Gangl, and C. Soulé. Perfect forms, K-theory and the cohomology of modular groups. *Advances in Mathematics*, 245:587–624, 2013.

[15] W. I. Gasarch. The P=?NP poll. *Sigact News*, 33(2):34–47, 2002.

[16] E. Grosswald. *Representations of integers as sums of squares*. Springer Science & Business Media, 2012.

[17] G. Hanrot, X. Pujol, and D. Stehlé. Algorithms for the shortest and closest lattice vector problems. In *International Conference on Coding and Cryptology*, pages 159–190. Springer, 2011.

[18] I. Haviv and O. Regev. On the lattice isomorphism problem. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 391–404. Society for Industrial and Applied Mathematics, 2014.

[19] D. Heath-Brown. Lattice points in the sphere. *Number theory in progress*, 2:883–892, 1997.

[20] J. E. Hopcroft and J.-K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 172–184. ACM, 1974.

[21] H. Iwaniec and F. C. Lorente. On the sphere problem. *Revista matemática iberoamericana*, 11(2):417–430, 1995.

[22] P. J. Kelly et al. A congruence theorem for trees. *Pacific J. Math*, 7(1):961–968, 1957.

[23] S. Li. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics and Statistics*, 4(1):66–70, 2011.

[24] E. M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of computer and system sciences*, 25(1):42–65, 1982.

[25] J. Martinet. *Perfect lattices in Euclidean spaces*, volume 327. Springer, 2003.

[26] B. D. McKay and A. Piperno. Nauty and Traces user's guide (Version 2.5), 2013.

[27] G. Nebe and N. Sloane. LATTICES - A Catalogue of Lattices. http://www.math.rwth-aachen.de/~Gabriele.Nebe/LATTICES.

[28] W. Plesken and B. Souvignier. Computing isometries of lattices. *Journal of Symbolic Computation*, 24(3):327–334, 1997.

[29] P. Schweitzer. *Problems of unknown complexity: graph isomorphism and Ramsey theoretic numbers*. PhD thesis, Saarbrücken, Univ., Diss., 2009, 2009.

[30] M. D. Sikirić, G. Ellis, and A. Schürmann. On the integral homology of $\mathrm{PSL}_4(\mathbb{Z})$ and other arithmetic groups. *Journal of Number Theory*, 131(12):2368–2375, 2011.

[31] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. premier mémoire. sur quelques propriétés des formes quadratiques positives parfaites. *Journal für die reine und angewandte Mathematik*, 133:97–178, 1908.

[32] V. Zemlyachenko, N. Korneenko, and R. Tyshkevich. Graph isomorphism problem. *Journal of Mathematical Sciences*, 29(4):1426–1481, 1985.