Designs, Codes and Cryptography (2023) 91:609–643
https://doi.org/10.1007/s10623-022-01119-y

The irreducible vectors of a lattice:

Some theory and applications

Emmanouil Doulgerakis[1] · **Thijs Laarhoven**[2] · **Benne de Weger**[1]

Received: 13 September 2021 / Revised: 30 June 2022 / Accepted: 17 August 2022 /
Published online: 18 October 2022

Abstract
The main idea behind lattice sieving algorithms is to reduce a sufficiently large number of lattice vectors with each other so that a set of short enough vectors is obtained. It is therefore natural to study vectors which cannot be reduced. In this work we give a concrete definition of an irreducible vector and study the properties of the set of all such vectors. We show that the set of irreducible vectors is a subset of the set of Voronoi relevant vectors and study its properties. For extremal lattices this set may contain as many as 2^n vectors, which leads us to define the notion of a complete system of irreducible vectors, whose size can be upper-bounded by the kissing number. One of our main results shows that modified heuristic sieving algorithms heuristically approximate such a set (modulo sign). We provide experiments in low dimensions which support this theory. Finally we give some applications of this set in the study of lattice problems such as SVP, SIVP and CVPP. The introduced notions, as well as various results derived along the way, may provide further insights into lattice algorithms and motivate new research into understanding these algorithms better.

Keywords Lattices · Relevant vectors · Irreducible vectors · Sieving algorithms

Communicated by D. Stehle.

Emmanouil Doulgerakis is supported by the NWO under Grant 628.001.028 (FASOR). Thijs Laarhoven is supported by a Veni Grant from NWO under Project Number 016.Veni.192.005.

✉ Emmanouil Doulgerakis
emmanouhld@gmail.com

Thijs Laarhoven
mail@thijs.com

Benne de Weger
b.m.m.d.weger@tue.nl

[1] Eindhoven University of Technology, Eindhoven, The Netherlands

[2] TNO, Cyber Security and Robustness, The Hague, The Netherlands

1 Introduction

The need for quantum-resistant cryptography has led to rapid developments in the area of lattice-based cryptography, mainly spurred by the NIST PQ-Crypto competition. Large scale deployment of lattice-based cryptosystems in the near future becomes realistic. This continues to make the deeper understanding of lattice problems an urgent research topic.

In 2010 Micciancio and Voulgaris, based also on previous work [1], described deterministic $\tilde{O}(2^{2n})$—time and $\tilde{O}(2^n)$—space algorithms to solve some of the most important lattice problems (such as SVP, SIVP and CVP) [23] in dimension n. This result mainly relies on an algorithm to compute the set of relevant vectors of (the Voronoi cell of) a lattice. Even though this is a very interesting result, the constants in the exponents of time and space complexities of the Micciancio–Voulgaris algorithm make it impractical, even for moderate dimensions.

The set of relevant vectors was first introduced in 1908 by Voronoi [35]. It provides a useful representation of the Voronoi cell of a lattice. Even though the set of relevant vectors seems to hold the key for solving many lattice problems, its expected size makes it impractical. This becomes even more clear when that size is compared to the (time and) space complexity of algorithms used in practice for solving lattice problems such as [2, 6, 14].

In this work, we introduce a different set of lattice vectors, which appears to serve as a bridge between the provable results relying on the set of relevant vectors and heuristic sieving algorithms [3, 24, 28].

Notions of irreducibility are considered to be fundamental in many areas. Often irreducibility is defined with respect to multiplication. Since a lattice is an additive object, we will however use an additive notion of irreducibility. Clearly the notion of *lattice basis* could be seen as such a construct, but it has been observed to be a too weak notion to provide, on its own, interesting results for lattice problems. Our new notion of *irreducible vectors* provides us with a set of lattice vectors, larger than a basis but smaller than the set of relevant vectors, and possessing interesting properties. To the best of our knowledge this definition is new in the area of lattices.

1.1 Contributions

In this paper we define a notion of irreducibility for a lattice vector. As a first result we show that every irreducible vector of a lattice belongs to the lattice's set of relevant vectors. Hence, the set of irreducible vectors which we denote by $\mathrm{Irr}(\mathscr{L})$ is finite. Additionally, it is shown that the set of irreducible vectors generates the lattice and also contains vectors achieving all the successive minima of the lattice. Finally, the sets of irreducible vectors of the root lattices A_n, D_n and their duals A_n^*, D_n^* are examined as they prove to be interesting extreme cases.

As it turns out, the set $\mathrm{Irr}(\mathscr{L})$ can be as big as the set of relevant vectors. In order to get a set of cardinality provably smaller than 2^n, a *complete system of irreducible vectors* is defined, which is denoted by $\mathrm{P}(\mathscr{L})$. This set inherits the aforementioned properties of the set $\mathrm{Irr}(\mathscr{L})$ and also it is proved that $|\mathrm{P}(\mathscr{L})| < 2^{0.402n}$ where n is the rank of the lattice. Heuristically it is expected that $\mathrm{P}(\mathscr{L})$ will have a cardinality of $2^{0.21n}$. From a computational point of view, it is shown that slightly modified versions of already existing sieving algorithms asymptotically output such a set (modulo sign). This statement is further supported by experimental results. Finally, we discuss the applicability of $\mathrm{P}(\mathscr{L})$ in showing that sieving algorithms like the ones described in [3, 24] can be used for tackling SVP, SIVP and computing the kissing number of a lattice. Additionally we discuss the applicability of $\mathrm{P}(\mathscr{L})$ as preprocessing data in a CVPP

algorithm which we call “the tuple slicer". The tuple slicer can provide a time–memory trade-off without the use of rerandomisations.

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

1.2 Motivation–future work

We believe that the notion of irreducibility will motivate further research on the field of lattices. In this work we focus only on pairwise irreducibility of vectors, even though a definition of higher order irreducibility is also given. In particular, pairwise irreducibility appears to have a close relation to lattice sieving algorithms. Thus, it could be that the set $\mathrm{P}(\mathscr{L})$ can provide further insight on this area. An interesting question would be if the usage of the set $\mathrm{P}(\mathscr{L})$ (under some heuristic assumptions on its size) enables the proof of an upper bound on the time complexity of the GaussSieve [24]. Examining the properties and the utility of higher order irreducibility is left for future research.

The implications of $\mathrm{P}(\mathscr{L})$ in cryptanalytic attacks could be an interesting topic to investigate. The set $\mathrm{P}(\mathscr{L})$ is expected to be affected by an underlying structure in the lattice $\mathscr{L}$. It can thus be expected that structured lattices end up with a smaller set $\mathrm{P}(\mathscr{L})$ than “average-case" lattices. Many of the modern lattice-based cryptosystems possess such underlying structures and hence they could serve as interesting cases to examine from this point of view.

In Sect. 5.1 we argue that computing $\mathrm{P}(\mathscr{L})$ by “brute force" can take up to $\tilde{O}(2^{2n})$ time. Therefore, this can serve as an upper bound. However, this bound may not be tight as discussed in Sect. 5.2. In Sect. 5.2 modified sieving algorithms were utilised in order to show how to compute $\mathrm{P}(\mathscr{L})$ asymptotically. But the question of how to compute it exactly or approximately in practice remains open. Such a result would also imply the ability to compute a subset of $\mathrm{R}(\mathscr{L})$ (of heuristically exponential size) without requiring the set $\mathrm{R}(\mathscr{L})$.

The set $\mathrm{P}(\mathscr{L})$ can be used as a tool in proving a behaviour of a lattice algorithm but could also be used itself (e.g. as preprocessing data of a CVPP algorithm). In Sect. 6 we propose the use of the “tuple slicer" in order to utilise the set $\mathrm{P}(\mathscr{L})$ in the CVPP framework. However this algorithm introduces a new question, namely what size of tuples should be considered during this algorithm. Figure 4 attempts to give some preliminary experimental evidence on this problem. However, a theoretical analysis of this question is left for future work.

Appendix B provides some experimental evidence showing that the size of a set $\mathrm{P}(\mathscr{L})$ could vary a lot in some cases. An “average-case" result implying that if the underlying lattice is not “special" then the size of $\mathrm{P}(\mathscr{L})$ cannot vary a lot would be of interest. A potential tool to reaching such a result could be lattice theta functions [12]. This is due to the fact that the coefficients in a lattice’s theta function actually represent the number of lattice vectors of a specific length. Therefore this property reveals the connection to the definition of $\mathrm{P}(\mathscr{L})$.

Outline. The rest of the paper is organised as follows. In Sect. 2 we introduce notation and give some background about lattices. Section 3 includes some prior work on the set of relevant vectors. The definition of irreducible vectors is given in Sect. 4 along with the first results regarding this new notion. In Sect. 5 we mention theoretical as well as experimental results on computing a complete system of irreducible vectors. Section 6 provides some initial arguments about the link between the new notions defined and the study of lattice algorithms and problems.

2 Preliminaries

To fix notation, let $\vec{\mathrm{B}} = \{\vec{b}_1, \ldots, \vec{b}_n\} \subset \mathbb{R}^n$ be a set of linearly independent vectors, which we may also interpret as a matrix with columns $\vec{b}_i$. The lattice generated by $\vec{\mathrm{B}}$ is defined as $\mathscr{L} = \mathscr{L}(\vec{\mathrm{B}}) := \{\vec{\mathrm{B}}\vec{x} : \vec{x} \in \mathbb{Z}^n\}$. In this paper we deal with full rank lattices unless indicated otherwise. We assume that the reader is familiar with notions such as the *volume* $\mathrm{Vol}(\mathscr{L}) := |\det(\vec{\mathrm{B}})|$, the *successive minima* $\lambda_i(\mathscr{L}) := \min\{\max_i \|\vec{x}_i\| \mid \vec{x}_1, \ldots, \vec{x}_i \in \mathscr{L} \text{ are linearly independent}\}$, in particular the *first successive minimum* $\lambda_1(\mathscr{L}) = \min_{\vec{v} \in \mathscr{L} \setminus \{\vec{0}\}} \|\vec{v}\|$. We refer to [22] for further details on these basic notions.

Definition 1 (First shell) Let $\mathscr{L}$ be a lattice. We define

$$S_1(\mathscr{L}) := \{\vec{v} \in \mathscr{L} \mid \|\vec{v}\| = \lambda_1(\mathscr{L})\}. \tag{1}$$

We call $S_1(\mathscr{L})$ the first shell of $\mathscr{L}$.

The following two well known concepts will be of major importance for our work, so we define them explicitly.

Definition 2 (Voronoi cell) The Voronoi cell $\mathscr{V}(\mathscr{L})$ of a full rank lattice $\mathscr{L}$ is the set of points in $\mathbb{R}^n$ which are closer to the origin than to any other lattice point, i.e.

$$\mathscr{V}(\mathscr{L}) := \{\vec{x} \in \mathbb{R}^n \mid \|\vec{x}\| \leq \|\vec{x} - \vec{v}\| \; \forall \vec{v} \in \mathscr{L}\}. \tag{2}$$

If it's clear what $\mathscr{L}$ is, we may use $\mathscr{V}$ instead of $\mathscr{V}(\mathscr{L})$. Closely related to the Voronoi cell of the lattice is the set of relevant vectors.

Definition 3 (Relevant vectors) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. The set of relevant vectors $\mathrm{R}(\mathscr{L})$ is

$$\mathrm{R}(\mathscr{L}) := \{\vec{r} \in \mathscr{L} \setminus \{\vec{0}\} \mid (\vec{r} + \mathscr{V}) \cap \mathscr{V} = \text{ an } (n-1) - \text{dimensional facet of } \mathscr{V}\}. \tag{3}$$

Let $\mathscr{B}(\vec{x}, r) := \{\vec{y} \in \mathbb{R}^n \mid \|\vec{y} - \vec{x}\| \leq r\}$ denote the closed n-dimensional ball with center $\vec{x}$ and radius r. Finally we have the *kissing number* τ_n, defined as the maximum number of equal n-dimensional spheres that can be made to touch another central sphere of the same size without intersecting.

See [22] for an overview of the main hard lattice problems that we will consider in this paper, namely the Shortest Vector Problem (SVP), determining the kissing number, the Shortest Independent Vector Problem (SIVP), and the Closest Vector Problem (CVP) and its Preprocessing variant (CVPP).

3 Previous work

In this section we give an overview of known results on the set of relevant vectors. This is done for a matter of completeness but also in order to indicate what kind of results we would like to obtain for the set of irreducible vectors which we will define later.

For $\vec{v} \in \mathscr{L}$ we define $H(\vec{v}) := \{\vec{x} \in \mathbb{R}^n \mid \|\vec{x}\| \leq \|\vec{x} - \vec{v}\|\}$, to relate the Voronoi cell of a lattice to its relevant vectors.

Proposition 1 (Relevant vectors) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. The set of relevant vectors $\mathrm{R}(\mathscr{L})$ is the minimal set $L \subset \mathscr{L}$ such that*

$$\mathscr{V}(\mathscr{L}) = \bigcap_{\vec{v} \in L} H(\vec{v}). \tag{4}$$

In order to get a more practical description of the relevant vectors the following theorem is used.

Theorem 1 (Identifying relevant vectors [35]) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$ and $\vec{v} \in \mathscr{L} \setminus \{\vec{0}\}$. Then $\vec{v} \in \mathrm{R}(\mathscr{L})$ if and only if $\vec{0}$ and $\vec{v}$ are the only closest vectors of $\mathscr{L}$ to $\frac{1}{2}\vec{v}$.*

This implies that

$$\mathrm{R}(\mathscr{L}) = \{\vec{v} \in \mathscr{L} \setminus \{\vec{0}\} \mid \|\frac{1}{2}\vec{v} - \vec{x}\| > \|\frac{1}{2}\vec{v}\| \ \forall \vec{x} \in \mathscr{L} \setminus \{\vec{0}, \vec{v}\}\} \tag{5}$$

$$= \{\vec{v} \in \mathscr{L} \setminus \{\vec{0}\} \mid \langle \vec{v}, \vec{x} \rangle < \|\vec{x}\|^2 \ \forall \vec{x} \in \mathscr{L} \setminus \{\vec{0}, \vec{v}\}\} \tag{6}$$

Remark 1 It holds that $\vec{0} \notin \mathrm{R}(\mathscr{L})$. Also note that if $\vec{v} \in \mathrm{R}(\mathscr{L})$ then $-\vec{v} \in \mathrm{R}(\mathscr{L})$.

Remark 2 The condition $\langle \vec{v}, \vec{x} \rangle < \|\vec{x}\|^2$ needs to be checked only for $\vec{x} \in \mathscr{L} \setminus \{\vec{0}\}$ such that $\|\vec{x}\| < \|\vec{v}\|$, because otherwise it is trivially true.

For checking if a vector is relevant, the following lemma is useful.

Lemma 1 (Identifying non-relevant vectors [23]) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$, and $\vec{v} \in \mathscr{L}$. If $\vec{v} \notin \mathrm{R}(\mathscr{L})$ then there exists $\vec{r} \in \mathrm{R}(\mathscr{L})$ such that $\langle \vec{v}, \vec{r} \rangle \geq \|\vec{r}\|^2$.*

Also a lower bound for the set $\mathrm{R}(\mathscr{L})$ can be obtained by the following trivial lemma.

Lemma 2 (All shortest vectors are relevant) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. It holds that $S_1(\mathscr{L}) \subseteq \mathrm{R}(\mathscr{L})$.*

Equality in the above lemma holds for a very special type of lattices called root lattices (see [8, Chapter 4]).

Theorem 2 (Root lattices [29]) *$S_1(\mathscr{L}) = \mathrm{R}(\mathscr{L})$ iff $\mathscr{L}$ is a root lattice.*

The following theorem by Minkowski gives an upper bound on the size of $\mathrm{R}(\mathscr{L})$.

Theorem 3 (Upper bound on $|\mathrm{R}(\mathscr{L})|$ [25]) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. It holds that $|\mathrm{R}(\mathscr{L})| \leq 2(2^n - 1)$.*

Apart from an upper bound we can also obtain a lower bound on $|\mathrm{R}(\mathscr{L})|$.

Remark 3 For the lattice $\mathbb{Z}^n$ it is true that $|R(\mathbb{Z}^n)| = 2n$ (see [8]). As the set $\mathrm{R}(\mathscr{L})$ needs to have n linearly independent vectors in order the volume of $\mathscr{V}(\mathscr{L})$ to be finite then $2n \leq |\mathrm{R}(\mathscr{L})|$ is a tight lower bound.

Proposition 2 (Volume of the Voronoi cell) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. It holds that $\mathrm{Vol}(\mathscr{L}) = \mathrm{Vol}(\mathscr{V}(\mathscr{L}))$.*

Proposition 3 (Relevant vectors generate the lattice) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. There exists a generating set $\vec{\mathrm{G}}$ of $\mathscr{L}$ such that $\vec{\mathrm{G}} \subseteq \mathrm{R}(\mathscr{L})$.*

Proof The Voronoi cell of $\mathscr{L}$ implies a tiling of $\mathbb{R}^n$. Thus, every vector in $\mathbb{R}^n$ can be reduced to a vector in $\mathscr{V}(\mathscr{L})$ through reductions by elements of $\mathrm{R}(\mathscr{L})$. As $\vec{0}$ is the only lattice vector in $\mathscr{V}(\mathscr{L})$ it follows that all lattice vectors are reduced to $\vec{0}$. Therefore, $\mathrm{R}(\mathscr{L})$ spans the entire lattice. □

So far we have mentioned a number of properties and definitions on the relevant vectors of a lattice. Computing them is however a different matter. The following result is the current state of the art on this.

Theorem 4 (Finding all relevant vectors [23]) *There exists a deterministic $\tilde{O}(2^{2n})$–time and $\tilde{O}(2^n)$–space algorithm which, given an n-rank lattice $\mathscr{L}$ with basis $\vec{\mathrm{B}}$, outputs the set of relevant vectors.*

4 Irreducibility of lattice vectors

4.1 The set of irreducible vectors

Inspired by number theoretic notions of (multiplicative) irreducibility, we introduce a similar concept for lattices (additively structured).

Definition 4 (Irreducibility) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$ and $\vec{v} \in \mathscr{L} \setminus \{\vec{0}\}$. The vector $\vec{v}$ is called k-irreducible iff $\nexists\, \vec{v}_1, \ldots, \vec{v}_k \in \mathscr{L}$ such that $\|\vec{v}_i\| < \|\vec{v}\|$ and $\vec{v}_1 + \cdots + \vec{v}_k = \vec{v}$. For the special case $k = 2$, $\vec{v}$ will be just called irreducible.

Remark 4 The definition of k-irreducible vectors implies that if a vector is k-irreducible then it is also $(k-1)$-irreducible. This observation allows the construction of a chain of subsets based on the notion of irreducibility.

In this work we are going to focus on the properties of 2-irreducibility. Further research on the notion of k-irreducibility for $k > 2$ is left for future research.

Definition 5 (Set of irreducible vectors) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. We define

$$\mathrm{Irr}(\mathscr{L}) := \{\vec{v} \in \mathscr{L} \mid \vec{v} \text{ is irreducible}\}. \tag{7}$$

Remark 5 It holds that $\vec{0} \notin \mathrm{Irr}(\mathscr{L})$. Also, if $\vec{v} \in \mathrm{Irr}(\mathscr{L})$ then $-\vec{v} \in \mathrm{Irr}(\mathscr{L})$.

The above properties hold for the set of relevant vectors as well and this is not a coincidence as we will see. First we show that this set is not empty, and indeed that it also contains vectors achieving the first successive minimum.

Lemma 3 (Shortest vectors are irreducible) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. It holds that:*

$$S_1(\mathscr{L}) \subseteq \mathrm{Irr}(\mathscr{L}). \tag{8}$$

Proof Let $\vec{v} \in S_1(\mathscr{L})$. Then clearly $\vec{v} \neq \vec{0}$. Assume that $\vec{v} \notin \mathrm{Irr}(\mathscr{L})$, so there exist $\vec{v}_1, \vec{v}_2 \in \mathscr{L}$ such that $\|\vec{v}_i\| < \|\vec{v}\|$ and $\vec{v}_1 + \vec{v}_2 = \vec{v}$. As $\vec{v} \in S_1(\mathscr{L})$ this implies that $\|\vec{v}_i\| < \lambda_1(\mathscr{L})$ and thus $\|\vec{v}_i\| = 0$. Hence, we get $\vec{v}_1 = \vec{v}_2 = \vec{0}$, which contradicts $\vec{v} \neq \vec{0}$. □

Remark 6 It can be easily checked that Lemma 3 would still hold under the notion of k-irreducibility for $k > 2$. Therefore we can conclude that k-irreducibility is not leading to a trivially empty set of vectors for $k > 2$. One may expect that it will also include a lattice basis.

We show that something similar occurs for the rest of the successive minima as well.

Definition 6 (Sublattice spanned by short vectors) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$ and $1 \leq i \leq n$. We define $\mathscr{L}_\lambda$ to be the sublattice spanned by all the vectors in $\mathscr{L}$ with norm strictly less than λ.

Proposition 4 (Identifying irreducible vectors) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$, and $\vec{v} \in \mathscr{L}$ satisfying $\|\vec{v}\| = \lambda_i := \lambda_i(\mathscr{L})$ for some $1 \leq i \leq n$. If $\vec{v} \notin \mathscr{L}_{\lambda_i}$ then $\vec{v}$ is irreducible.*

Proof It has been already proven in Lemma 3 that this is true for $i = 1$ so we can consider $i \geq 2$. Assume $\vec{v} \in \mathscr{L}$ such that $\|\vec{v}\| = \lambda_i(\mathscr{L})$ for some $2 \leq i \leq n$, $\vec{v} \notin \mathscr{L}_{\lambda_i}$ and $\vec{v}$ is not irreducible. Then there exist $\vec{v}_1, \vec{v}_2 \in \mathscr{L}$ such that $\vec{v} = \vec{v}_1 + \vec{v}_2$ and $\|\vec{v}_j\| < \|\vec{v}\|$ for $j = 1, 2$. Clearly $\vec{v}_j \neq 0$. As $\|\vec{v}_j\| < \|\vec{v}\| = \lambda_i(\mathscr{L})$ this implies that $\vec{v}_j \in \mathscr{L}_{\lambda_i}$ for $j = 1, 2$. This further implies that $\vec{v} = \vec{v}_1 + \vec{v}_2 \in \mathscr{L}_{\lambda_i}$, contradiction. □

Remark 7 Proposition 4 points out that a lattice vector achieving a successive minimum is not necessarily irreducible. An enlightening example of such an occasion is the following. Consider the lattice $\mathscr{L} = \mathscr{L}(\vec{\mathrm{B}})$ generated by the matrix

$$\vec{\mathrm{B}} = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 10 \end{pmatrix}. \tag{9}$$

Then $\lambda_1(\mathscr{L}) = 3$, $\lambda_2(\mathscr{L}) = 4$ and $\lambda_3(\mathscr{L}) = 10$. The vector $\vec{v} = (6, 8, 0)$ is such that $\|\vec{v}\| = \lambda_3(\mathscr{L})$ but $\vec{v}$ is not irreducible as it can be written as a sum of shorter lattice vectors i.e. $\vec{v} = (6, 0, 0) + (0, 8, 0)$. The reason why $\vec{v}$ fails to be irreducible is that it belongs to the sublattice $\mathscr{L}_{\lambda_3}$.

Corollary 1 (Irreducible vectors and successive minima) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. For every $i = 1, \ldots, n$ there exists a vector $\vec{v} \in \mathrm{Irr}(\mathscr{L})$ such that $\|\vec{v}\| = \lambda_i(\mathscr{L})$.*

Proof By Proposition 4 it suffices to show that for every $i = 1, \ldots, n$ there exists a vector $\vec{v} \in \mathscr{L}$ such that $\|\vec{v}\| = \lambda_i(\mathscr{L})$ and $\vec{v} \notin \mathscr{L}_{\lambda_i}$. Assume that for every vector $\vec{v} \in \mathscr{L}$ such that $\|\vec{v}\| = \lambda_i(\mathscr{L})$ for some fixed $2 \leq i \leq n$ it holds that $\vec{v} \in \mathscr{L}_{\lambda_i}$. For convenience we define $\lambda_0(\mathscr{L}) = 0$. Let k be $\min_{1 \leq j \leq i} j$ such that $\lambda_j(\mathscr{L}) = \lambda_i(\mathscr{L})$ and therefore $\lambda_{k-1}(\mathscr{L}) < \lambda_k(\mathscr{L}) = \lambda_i(\mathscr{L})$. Then $\mathscr{L}_{\lambda_i}$ has rank $k-1$ as $\lambda_{k-1}(\mathscr{L}) < \lambda_k(\mathscr{L})$. If $k - 1 = 0$ then we are done as this would imply $\vec{v} = \vec{0}$. If $k - 1 \geq 1$ then $\vec{v}$ belongs to the sublattice containing all the shorter vectors than it, $\mathscr{L}_{\lambda_i}$ and this sublattice is of rank $k-1$. Thus any choice of $k-1$ vectors such that $\max\{\|\vec{v}_1\|, \ldots, \|\vec{v}_{k-1}\|, \|\vec{v}\|\} = \|\vec{v}\|$ will result in a linearly dependent set. Hence it cannot be that $\lambda_k(\mathscr{L}) = \|\vec{v}\|$, contradiction. □

Apart from vectors reaching the successive minima, it can be shown that the set $\mathrm{Irr}(\mathscr{L})$ contains a generating set of the lattice as well.

Proposition 5 (Irreducible vectors generate the lattice) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. There exists a generating set $\vec{\mathrm{G}}$ of $\mathscr{L}$ such that $\vec{\mathrm{G}} \subseteq \mathrm{Irr}(\mathscr{L})$.*

Proof We will prove that the set $\mathrm{Irr}(\mathscr{L})$ spans the lattice and therefore it includes a generating set. Let $\vec{v} \in \mathscr{L}$. If $\nexists\, \vec{v}_1, \vec{v}_2 \in \mathscr{L}$ with $\|\vec{v}_i\| < \|\vec{v}\|$ such that $\vec{v}_1 + \vec{v}_2 = \vec{v}$ then $\vec{v} \in \mathrm{Irr}(\mathscr{L})$. If there exist such $\vec{v}_i$ then write $\vec{v} = \vec{v}_1 + \vec{v}_2$. If the $\vec{v}_i \in \mathrm{Irr}(\mathscr{L})$ then we are done. If not then further reduce the vectors $\vec{v}_i$ such that they are written as a sum of two strictly shorter vectors. As in each step the length of the vectors strictly reduces and there is a finite number of lattice points in $\mathscr{B}(\vec{0}, \|\vec{v}\|)$, after a finite number of steps we will reach a state where $\vec{v} = \sum \vec{p}_i$ and $\vec{p}_i \in \mathrm{Irr}(\mathscr{L})$. This concludes the proof. □

Given the result of Proposition 5 the following conjecture can be formulated.

Conjecture 1 Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. The set $\mathrm{Irr}(\mathscr{L})$ contains a basis of $\mathscr{L}$.[1]

Our next goal is to derive some more explicit descriptions of the set $\mathrm{Irr}(\mathscr{L})$.

Lemma 4 (Characterizing the set of irreducible vectors) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. It holds that*

$$\mathrm{Irr}(\mathscr{L}) = \{\vec{v} \in \mathscr{L} \setminus \{\vec{0}\} \mid \forall \vec{x} \in \mathscr{L} \text{ with } \|\vec{x}\| < \|\vec{v}\| \text{ it holds that } \|\vec{v} - \vec{x}\| \geq \|\vec{v}\|\} \tag{10}$$

$$= \{\vec{v} \in \mathscr{L} \setminus \{\vec{0}\} \mid \forall \vec{x} \in \mathscr{L} \text{ with } \|\vec{x}\| < \|\vec{v}\| \text{ it holds that } 2\langle \vec{v}, \vec{x} \rangle \leq \|\vec{x}\|^2\}. \tag{11}$$

[1] Conjecture 1 is not used in order to derive any conclusions in this work.

Proof Let $A = \{\vec{v} \in \mathscr{L} \setminus \{\vec{0}\} \,|\, \forall \vec{x} \in \mathscr{L} \text{ with } \|\vec{x}\| < \|\vec{v}\| \text{ it holds that } \|\vec{v} - \vec{x}\| \geq \|\vec{v}\|\}$. Let $\vec{p} \in \mathrm{Irr}(\mathscr{L})$ and $\vec{v} \in \mathscr{L}$ with $\|\vec{v}\| < \|\vec{p}\|$. Then as $\vec{p} \in \mathrm{Irr}(\mathscr{L})$ we get $\|\vec{p} - \vec{v}\| \geq \|\vec{p}\|$ because otherwise $\vec{p}$ would have a decomposition into two shorter vectors, thus $\vec{p} \in A$. This gives $\mathrm{Irr}(\mathscr{L}) \subseteq A$. Next, let $\vec{v} \in A$, and write $\vec{v} = \vec{v}_1 + \vec{v}_2$ for some $\vec{v}_1, \vec{v}_2 \in \mathscr{L}$. If $\|\vec{v}_1\| < \|\vec{v}\|$ then as $\vec{v} \in A$ we get $\|\vec{v} - \vec{v}_1\| \geq \|\vec{v}\|$ and hence we do not get a decomposition of $\vec{v}$ in two shorter vectors. If $\|\vec{v}_1\| \geq \|\vec{v}\|$ this is trivially true. Thus $\vec{v} \in \mathrm{Irr}(\mathscr{L})$. This implies equality (10) and equality (11) is an immediate consequence. This concludes the proof. □

Even though this lemma is rather straightforward it implies an interesting result for the set $\mathrm{Irr}(\mathscr{L})$.

Proposition 6 (Irreducible vectors are relevant) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. Every irreducible vector of $\mathscr{L}$ is also a relevant vector of $\mathscr{L}$, hence:*

$$\mathrm{Irr}(\mathscr{L}) \subseteq \mathrm{R}(\mathscr{L}). \tag{12}$$

Proof As we already saw by Theorem 1, we can write the set $\mathrm{R}(\mathscr{L})$ as $\mathrm{R}(\mathscr{L}) = \{\vec{v} \in \mathscr{L} \setminus \{\vec{0}\} \,|\, \langle \vec{v}, \vec{x} \rangle < \|\vec{x}\|^2 \ \forall \vec{x} \in \mathscr{L} \setminus \{\vec{0}, \vec{v}\}\}$ and we can further improve that description to

$$\mathrm{R}(\mathscr{L}) = \{\vec{v} \in \mathscr{L} \setminus \{\vec{0}\} \,|\, \forall \vec{x} \in \mathscr{L} \setminus \{\vec{0}\} \text{ with } \|\vec{x}\| < \|\vec{v}\| \text{ it holds that } \langle \vec{v}, \vec{x} \rangle < \|\vec{x}\|^2\}.$$

For the set of irreducible vectors we got from Lemma 4 that

$$\mathrm{Irr}(\mathscr{L}) = \{\vec{v} \in \mathscr{L} \setminus \{\vec{0}\} \,|\, \forall \vec{x} \in \mathscr{L} \text{ with } \|\vec{x}\| < \|\vec{v}\| \text{ it holds that } 2\langle \vec{v}, \vec{x} \rangle \leq \|\vec{x}\|^2\}.$$

Thus by carefully checking these two descriptions for the sets $\mathrm{R}(\mathscr{L})$ and $\mathrm{Irr}(\mathscr{L})$ it suffices to prove that if $\vec{v} \in \mathscr{L} \setminus \{\vec{0}\}$ and $\vec{x} \in \mathscr{L} \setminus \{\vec{0}\}$ with $\|\vec{x}\| < \|\vec{v}\|$ then $2\langle \vec{v}, \vec{x} \rangle \leq \|\vec{x}\|^2 \Rightarrow \langle \vec{v}, \vec{x} \rangle < \|\vec{x}\|^2$.
If $\langle \vec{v}, \vec{x} \rangle \leq 0$ this is trivially true as $\vec{x} \neq \vec{0}$. Also if $\langle \vec{v}, \vec{x} \rangle > 0$ then $\langle \vec{v}, \vec{x} \rangle < 2\langle \vec{v}, \vec{x} \rangle$ and the result follows. □

Remark 8 Combining the result of Lemma 3 and Proposition 6 we get that $S_1(\mathscr{L}) \subseteq \mathrm{Irr}(\mathscr{L}) \subseteq \mathrm{R}(\mathscr{L})$. Therefore $\mathrm{Irr}(\mathscr{L})$ is finite.

We already saw that for the case of root lattices it holds that $S_1(\mathscr{L}) = \mathrm{R}(\mathscr{L})$. This implies that for the root lattices it also holds that $S_1(\mathscr{L}) = \mathrm{Irr}(\mathscr{L}) = \mathrm{R}(\mathscr{L})$. Thus, the sets $S_1(\mathscr{L})$ and $\mathrm{R}(\mathscr{L})$ are tight inclusions of $\mathrm{Irr}(\mathscr{L})$.

We expect that in general though it will hold $S_1(\mathscr{L}) \subsetneqq \mathrm{Irr}(\mathscr{L}) \subsetneqq \mathrm{R}(\mathscr{L})$. A question that might be of interest is when and if $S_1(\mathscr{L}) = \mathrm{Irr}(\mathscr{L}) \subsetneqq \mathrm{R}(\mathscr{L})$ or $S_1(\mathscr{L}) \subsetneqq \mathrm{Irr}(\mathscr{L}) = \mathrm{R}(\mathscr{L})$ are possible.

We believe that lattices satisfying either of these properties will be very special and highly symmetric. The reason why we believe this, is that some already well known very special families of lattices satisfy these properties. Namely, in Appendix A we will prove the following two theorems.

Theorem 5 (The root lattices D_n^*) *Let $n \in \mathbb{N}$ with $n \geq 5$. Then for the lattice D_n^* it holds that $S_1(D_n^*) \subsetneqq \mathrm{Irr}(D_n^*) = \mathrm{R}(D_n^*)$. Furthermore $|\mathrm{Irr}(D_n^*)| = 2^n + 2n$.*

Theorem 6 (The root lattices A_n^*) *Let $n \in \mathbb{N}$ with $n \geq 3$. Then for the lattice A_n^* it holds that $S_1(A_n^*) = \mathrm{Irr}(A_n^*) \subsetneqq \mathrm{R}(A_n^*)$. Furthermore $|\mathrm{Irr}(A_n^*)| = 2(n+1)$.*

Additionally the famous Leech lattice Λ_{24} [8, p. 131] satisfies the property $S_1(\Lambda_{24}) = \mathrm{Irr}(\Lambda_{24}) \subsetneqq \mathrm{R}(\Lambda_{24})$. We will actually be able to prove in the next subsection that for every lattice that reaches the kissing number τ_n it holds that $S_1(\mathscr{L}) = \mathrm{Irr}(\mathscr{L})$.

4.2 A complete system of irreducible vectors

The special family of lattices D_n^* indicates that the set $\mathrm{Irr}(\mathscr{L})$ can become as big as $\mathrm{R}(\mathscr{L})$ and actually grow as much in size as 2^n. However, our goal is to obtain a subset of $\mathrm{Irr}(\mathscr{L})$ which is closely related to it but also provably smaller than 2^n.

Definition 7 (Equivalence relation on $\mathrm{Irr}(\mathscr{L})$) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. We define an equivalence relation on $\mathrm{Irr}(\mathscr{L})$ in the following way.

$$\text{Let } \vec{v}_1, \vec{v}_2 \in \mathrm{Irr}(\mathscr{L}) \quad \text{then} \quad \vec{v}_1 \sim \vec{v}_2 \quad \text{iff} \quad \|\vec{v}_1\| = \|\vec{v}_2\|. \tag{13}$$

From each equivalence class we will consider at least two representatives. We choose them in the following way and we will explain afterwards why.

Definition 8 (Representative set of equivalence classes) For each equivalence class $S = \{\vec{v}_1, \ldots, \vec{v}_m\}$ of $\mathrm{Irr}(\mathscr{L})$ according to (13) we choose a subset $\tilde{S} \subseteq S$ such that the following two conditions hold:

(i) If $\vec{v} \in \tilde{S}$ then also $-\vec{v} \in \tilde{S}$.
(ii) $\tilde{S}$ is a maximal subset of S such that for every pair of vectors $\vec{v}_1, \vec{v}_2 \in \tilde{S}$ with $\vec{v}_2 \neq -\vec{v}_1$ it holds that $\|\vec{v}_1 + \vec{v}_2\| \geq \|\vec{v}_1\|$.

The main motivation is that the new set of vectors which will be built under these rules will include irreducible vectors whose pairwise angle is "big" as we will prove later. However, there are several details of this definition which should be clarified. First of all, from the definition it follows that for an equivalence class we consider at least two representatives, which is not usually done. The reasons for this are the following.

Initially, for the subset of $\mathrm{Irr}(\mathscr{L})$ which we are trying to define, we would like it to inherit the property of $\mathrm{Irr}(\mathscr{L})$ that if $\vec{v}$ belongs to it then also $-\vec{v}$ belongs to it. A second, more important reason is that choosing only one representative per equivalence class could lead to a set that does not even span the lattice (for example in the case of root lattices, or whenever $S_1(\mathscr{L}) = \mathrm{Irr}(\mathscr{L})$).

The second condition of the definition implies that for every element $\vec{v}$ of a class S which is not included in $\tilde{S}$ there exists a vector $\tilde{\vec{v}} \in \mathscr{L}$ such that $\|\vec{v} - \tilde{\vec{v}}\| < \|\vec{v}\|$. From this point of view the remaining elements of a class S which are not included in $\tilde{S}$ can be generated by the elements of $\tilde{S}$ plus some strictly shorter vector. In order to ensure that this holds we take $\tilde{S}$ to be maximal. Also by taking $\tilde{S}$ to be maximal we make sure that the set $\tilde{S}$ contains as much information about the class as possible.

Remark 9 Choosing a representative set $\tilde{S}$ of a class S can be translated into a graph problem. We define a graph where the set of vertices is the equivalence class, and there exists an edge between two vertices iff the difference of the corresponding vectors is strictly shorter than both of them. Then choosing a set of representatives translates to finding a maximal subset of vertices that are not adjacent, while keeping the symmetry about $\vec{0}$. In terms of graph theory this can be phrased as finding a special independent set of the graph. This idea is further analysed in Appendix B.

Definition 9 (Complete system of irreducible vectors) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. We define a set $\mathrm{P} \subseteq \mathrm{Irr}(\mathscr{L})$ to be a complete system of irreducible vectors of $\mathscr{L}$ if it is of the form:

$$\mathrm{P} = \bigcup_{S \in \mathrm{Irr}(\mathscr{L})/\sim} \tilde{S}. \tag{14}$$

Remark 10 Below we denote by $\mathrm{P}(\mathscr{L})$ any one of the complete systems of irreducible vectors of $\mathscr{L}$. It is clear that there always exists such a set $\mathrm{P}(\mathscr{L})$ and it is not necessarily unique. In fact, even the size of $\mathrm{P}(\mathscr{L})$ can vary.

Remark 11 By the fact that for each class S of $\mathrm{Irr}(\mathscr{L})/\sim$ we have $\tilde{S} \subseteq S$ we get that $\mathrm{P}(\mathscr{L}) \subseteq \mathrm{Irr}(\mathscr{L})$. Also the class of $\mathrm{Irr}(\mathscr{L})/\sim$ containing all the shortest vectors i.e. $S_1(\mathscr{L})$ will be entirely included in $\mathrm{P}(\mathscr{L})$ as any pairwise sum of vectors (for non-trivial pairs) in this class will be longer or equally long by definition. Thus we can conclude that

$$S_1(\mathscr{L}) \subseteq \mathrm{P}(\mathscr{L}) \subseteq \mathrm{Irr}(\mathscr{L}) \subseteq \mathrm{R}(\mathscr{L}).$$

We will also give an example in order to illustrate this definition.

Example 1 Let $\mathscr{L} = \mathscr{L}(\vec{\mathrm{B}})$ be the lattice generated by the columns of the matrix

$$\vec{\mathrm{B}} = \begin{pmatrix} 0 & 0 & 0 & -1 & 3 \\ -1 & -1 & 0 & 1 & 0 \\ 0 & 0 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 & 1 \end{pmatrix}.$$

We find the sets $S_1(\mathscr{L})$, $\mathrm{P}(\mathscr{L})$, $\mathrm{Irr}(\mathscr{L})$, $\mathrm{R}(\mathscr{L})$.
In fact $\vec{\mathrm{B}}$ is an LLL-reduced basis [20] of the lattice. By means of enumeration one could verify that $S_1(\mathscr{L}) = \{\pm(0, -1, 0, 1, 0)\}$. By running an algorithm that computes the set of relevant vectors like [34] in SAGE [9] we get

$$\begin{aligned} \mathrm{R}(\mathscr{L}) = \{ & \pm (0, -1, 0, 1, 0), \\ & \pm (0, -1, 0, -1, 1), \\ & \pm (0, 0, 2, 0, 0), \\ & \pm (-1, 0, -1, 1, 2), \pm(-1, 0, 1, 1, 2), \pm(-1, 1, -1, 0, 2), \pm(-1, 1, 1, 0, 2), \\ & \pm (-1, 1, -1, 2, 1), \pm(-1, 1, 1, 2, 1), \pm(-1, 2, -1, 1, 1), \pm(-1, 2, 1, 1, 1), \\ & \pm (3, 1, 0, 0, 1), \pm(3, 0, 0, 1, 1), \\ & \pm (2, 1, -1, 1, 3), \pm(2, 1, 1, 1, 3), \\ & \pm (2, 2, -1, 2, 2), \pm(2, 2, 1, 2, 2)\}. \end{aligned}$$

(each line has vectors of equal norm). The next step is to find the set of irreducible vectors $\mathrm{Irr}(\mathscr{L})$. We consider the subset of $\mathrm{R}(\mathscr{L})$ containing relevant vectors which cannot be written as a sum of two strictly shorter vectors (by cross-checking with the set of relevant vectors). It turns out that this set just contains all the vectors achieving the successive minima thus it must be that this is $\mathrm{Irr}(\mathscr{L})$.

$$\begin{aligned} \mathrm{Irr}(\mathscr{L}) = \{ & \pm (0, -1, 0, 1, 0), \\ & \pm (0, -1, 0, -1, 1), \\ & \pm (0, 0, 2, 0, 0), \\ & \pm (-1, 0, -1, 1, 2), \pm(-1, 0, 1, 1, 2), \pm(-1, 1, -1, 0, 2), \pm(-1, 1, 1, 0, 2), \\ & \pm (3, 1, 0, 0, 1), \pm(3, 0, 0, 1, 1)\} \end{aligned}$$

The set $\mathrm{Irr}(\mathscr{L})$ contains 5 equivalence classes according to the equivalence relation (13). We denote them by C_i for $i = 1, \ldots, 5$. As we can see for the first three of them, computing a set of representatives $\tilde{C}_1, \tilde{C}_2, \tilde{C}_3$ is trivial as in these cases it will be $\tilde{C}_1 = C_1$, $\tilde{C}_2 = C_2$

Fig. 1 The graph of the class C_5

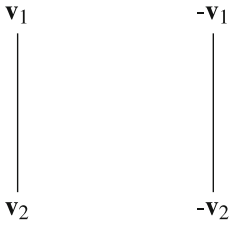
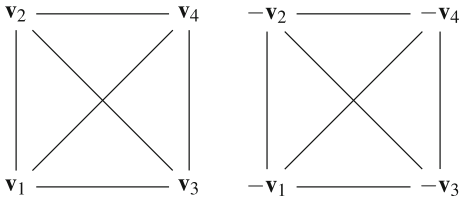


Fig. 2 The graph of the class C_4



and $\tilde{C}_3 = C_3$. The cases of C_4 and C_5 are more interesting. We start by examining C_5 as it contains less vectors. We set $\vec{v}_1 = (3, 1, 0, 0, 1)$ and $\vec{v}_2 = (3, 0, 0, 1, 1)$. Next we draw the corresponding graph with vertices the $\pm\vec{v}_1, \pm\vec{v}_2$ and edges if the pairwise differences are strictly shorter than $\|\vec{v}_1\|$.

The graph in Fig. 1 shows that we can take either $\tilde{C}_5 = \{\pm\vec{v}_1\}$ or $\tilde{C}_5 = \{\pm\vec{v}_2\}$. We are now going to do the same for the class C_4. We set $\vec{v}_1 = (-1, 0, -1, 1, 2)$,
$\vec{v}_2 = (-1, 0, 1, 1, 2)$, $\vec{v}_3 = (-1, 1, -1, 0, 2)$, $\vec{v}_4 = (-1, 1, 1, 0, 2)$.

The graph in Fig. 2 shows that we can take $\tilde{C}_4 = \{\pm\vec{v}_i\}$ for any $i = 1, 2, 3, 4$. Therefore one choice for the set $\mathrm{P}(\mathscr{L})$ is the following.

$$\begin{aligned}\mathrm{P}(\mathscr{L}) = \{ &\pm (0, -1, 0, 1, 0),\\ &\pm (0, -1, 0, -1, 1),\\ &\pm (0, 0, 2, 0, 0),\\ &\pm (-1, 0, -1, 1, 2),\\ &\pm (3, 1, 0, 0, 1)\}\end{aligned}$$

Remark 12 The above example should not mislead the reader to the assumption that the corresponding graph of each equivalence class will always have at least two connected components. It can happen that the graph of a class is connected. One such example can be derived from the family of lattices examined in Theorem 10.

One property of the set $\mathrm{Irr}(\mathscr{L})$ was that it includes a generating set of $\mathscr{L}$. We can show that $\mathrm{P}(\mathscr{L})$ inherits that property.

Proposition 7 (Complete system generates the lattice) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. Then for every $\mathrm{P}(\mathscr{L}) \subseteq \mathrm{Irr}(\mathscr{L})$ there exists a generating set $\vec{\mathrm{G}}$ of $\mathscr{L}$ such that $\vec{\mathrm{G}} \subseteq \mathrm{P}(\mathscr{L})$.*

Proof As in the proof of Proposition 5 for the set $\mathrm{Irr}(\mathscr{L})$ we will prove that the set $\mathrm{P}(\mathscr{L})$ spans the lattice and therefore it includes a generating set. However, in this case the proof is more technical. Let $\mathrm{P}(\mathscr{L})$ be a complete system of irreducible vectors of $\mathscr{L}$ as defined in (14). We have already shown that $\mathrm{Irr}(\mathscr{L})$ is finite as $\mathrm{Irr}(\mathscr{L}) \subseteq \mathrm{R}(\mathscr{L})$ and thus we can define $t := |\mathrm{Irr}(\mathscr{L})/\sim|$. We further set C_i for $i = 1, \ldots, t$ to be the equivalence classes in $\mathrm{Irr}(\mathscr{L})/\sim$. Hence, the set $\mathrm{P}(\mathscr{L})$ can be written as $\mathrm{P}(\mathscr{L}) = \cup_{i=1}^{t} \tilde{C}_i$. Each equivalence class

C_i contains all irreducible vectors of a specific length μ_i, and we can assume that we have ordered the C_i according to increasing μ_i. We define the following sequence of subsets of $\mathrm{Irr}(\mathscr{L})$:

$$A_i := \left(\bigcup_{j=1}^{i-1} C_i\right) \cup \left(\bigcup_{j=i}^{t} \tilde{C}_i\right) \quad \text{for} \quad i = 1, \ldots, t+1.$$

As for every i it holds that $\tilde{C}_i \subseteq C_i$ then it follows that $A_i(\mathscr{L}) \subseteq A_{i+1}(\mathscr{L})$ and thus

$$\mathrm{P}(\mathscr{L}) = A_1 \subseteq A_2 \subseteq \cdots \subseteq A_t \subseteq A_{t+1} = \mathrm{Irr}(\mathscr{L}).$$

We will prove by induction that each term of this sequence of sets spans the lattice $\mathscr{L}$.
Base case $i = t+1$ The set $\mathrm{Irr}(\mathscr{L}) = A_{t+1}$ spans the lattice as it was already shown in Proposition 5.
Induction hypothesis Assume that it holds for some $i = k$, i.e. A_k spans the lattice for some $k \in \{2, \ldots, t+1\}$.
Induction step Prove that A_{k-1} spans the lattice. By the definition of the sets A_i we can conclude that $A_{k-1} = A_k \setminus (C_{k-1} \setminus \tilde{C}_{k-1})$. By the induction hypothesis it suffices to show that the vectors in $C_{k-1} \setminus \tilde{C}_{k-1}$ can be generated by the vectors in A_{k-1}. Let $\vec{v} \in C_{k-1} \setminus \tilde{C}_{k-1}$. As $\vec{v} \in C_{k-1}$ but $\vec{v} \notin \tilde{C}_{k-1}$ this implies that there exists a $\tilde{\vec{v}} \in \tilde{C}_{k-1}$ such that $\|\vec{v} + \tilde{\vec{v}}\| < \|\vec{v}\|$. This holds because $\tilde{C}_{k-1}$ is maximal by definition. We set $\vec{w} = \vec{v} + \tilde{\vec{v}}$. Furthermore as $\|\vec{w}\| < \|\vec{v}\|$ then $\vec{w}$ is either irreducible or can be written as a sum of irreducible vectors shorter than $\|\vec{v}\|$. We use the ordering of the C_i. Thus by its definition the set A_{k-1} contains all the vectors in $\mathrm{Irr}(\mathscr{L})$ which are shorter than $\|\vec{v}\|$. Hence as, $\|\vec{w}\| < \|\vec{v}\|$ this implies that $\vec{w}$ can be generated by the vectors in A_{k-1}. So, concluding we wrote $\vec{v}$ as $\vec{v} = \vec{w} - \tilde{\vec{v}}$ where both $\vec{w}$ and $\tilde{\vec{v}}$ belong to A_{k-1}. This concludes the proof. □

For $\vec{v}_1, \vec{v}_2 \in \mathscr{L}$ we denote by $\vartheta(\vec{v}_1, \vec{v}_2)$ the angle formed by $\vec{v}_1, \vec{v}_2$.

Proposition 8 (Properties of complete system) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$, and $\vec{p}_1, \vec{p}_2 \in \mathrm{P}(\mathscr{L})$ such that $\vec{p}_1 \neq \pm\vec{p}_2$. Then it holds that*

(i) $\min\{\|\vec{p}_1 \pm \vec{p}_2\|\} \geq \max\{\|\vec{p}_1\|, \|\vec{p}_2\|\}$ *and*
(ii) $|\cos\vartheta(\vec{p}_1, \vec{p}_2)| \leq \frac{1}{2}$.

Proof (Part i) By Lemma 4 we have that

$$\mathrm{Irr}(\mathscr{L}) = \{\vec{v} \in \mathscr{L} \setminus \{\vec{0}\} \,|\, \forall \vec{x} \in \mathscr{L} \text{ with } \|\vec{x}\| < \|\vec{v}\| \text{ it holds that } \|\vec{v} - \vec{x}\| \geq \|\vec{v}\|\}.$$

Let $\vec{p}_1, \vec{p}_2 \in P(\mathscr{L})$ such that $\vec{p}_1 \neq \pm\vec{p}_2$. Without loss of generality we assume that $\|\vec{p}_2\| \leq \|\vec{p}_1\|$. Initially we will prove that $\|\vec{p}_1 + \vec{p}_2\| \geq \max\{\|\vec{p}_1\|, \|\vec{p}_2\|\}$.
Case 1 If $\|\vec{p}_2\| < \|\vec{p}_1\|$. Then $\vec{p}_1, \vec{p}_2 \in \mathrm{Irr}(\mathscr{L})$ and they are not in the same class. Using the description of Lemma 4 with $\vec{v} = \vec{p}_1 \in \mathrm{Irr}(\mathscr{L})$ and $\vec{x} = -\vec{p}_2$ we get $\|\vec{p}_1 + \vec{p}_2\| \geq \|\vec{p}_1\|$. But as $\|\vec{p}_2\| < \|\vec{p}_1\|$ we can conclude that $\|\vec{p}_1 + \vec{p}_2\| \geq \max\{\|\vec{p}_1\|, \|\vec{p}_2\|\}$.
Case 2 If $\|\vec{p}_2\| = \|\vec{p}_1\|$. Then $\vec{p}_1, \vec{p}_2 \in \mathrm{Irr}(\mathscr{L})$ and they are in the same class. Let $S \in \mathrm{Irr}(\mathscr{L})/\sim$ such that $\vec{p}_1, \vec{p}_2 \in S$. Then as $\vec{p}_1, \vec{p}_2 \in \mathrm{P}(\mathscr{L})$ we get that $\vec{p}_1, \vec{p}_2$ belong to the same $\tilde{S}$. Thus, by the definition of $\tilde{S}$ we can again conclude that $\|\vec{p}_1 + \vec{p}_2\| \geq \max\{\|\vec{p}_1\|, \|\vec{p}_2\|\}$. The result follows from the fact that for every $\vec{v} \in \mathrm{P}(\mathscr{L})$ also $-\vec{v} \in \mathrm{P}(\mathscr{L})$.
(Part ii) Let $\vec{p}_1, \vec{p}_2 \in \mathrm{P}(\mathscr{L})$ such that $\vec{p}_1 \neq \pm\vec{p}_2$. Without loss of generality we assume that $\|\vec{p}_2\| \leq \|\vec{p}_1\|$. By part (i) we get that $\|\vec{p}_1 \pm \vec{p}_2\| \geq \|\vec{p}_1\|$. This in turn implies that $2|\langle\vec{p}_1, \vec{p}_2\rangle| \leq \|\vec{p}_2\|^2$. Hence,

$$|\cos\vartheta(\vec{p}_1,\vec{p}_2)| = \frac{|\langle\vec{p}_1,\vec{p}_2\rangle|}{\|\vec{p}_1\|\|\vec{p}_2\|} \leq \frac{|\langle\vec{p}_1,\vec{p}_2\rangle|}{\|\vec{p}_2\|^2} \leq \frac{1}{2}.$$

□

We will use the following theorem in order to bound $|\mathrm{P}(\mathscr{L})|$.

Theorem 7 (Upper bound on kissing constant [18]) *Let $A(n,\phi_0)$ be the maximal size of any set C of points in $\mathbb{R}^n$ such that the angle between any two distinct vectors $\vec{v}_i,\vec{v}_j \in C$ (denoted $\phi_{\vec{v}_i,\vec{v}_j}$) is at least ϕ_0. If $0 < \phi_0 < 63°$, then for all sufficiently large n, $A(n,\phi_0) = 2^{cn}$ for some*

$$c \leq -\frac{1}{2}\log_2(1-\cos(\phi_0)) - 0.099. \tag{15}$$

Proposition 9 (Upper bound on $|\mathrm{P}(\mathscr{L})|$) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. It holds that $|\mathrm{P}(\mathscr{L})| < 2^{0.402n}$.*

Proof By using Theorem 7 with $\phi_0 = \frac{\pi}{3}$ (which can be deduced from Proposition 8) we get that $|\mathrm{P}(\mathscr{L})| = 2^{cn}$ with $c \leq -\frac{1}{2}\log_2(1-\cos(\frac{\pi}{3})) - 0.099$. Evaluating the right hand side of this inequality implies the result. □

Proposition 8 states the same condition that is also satisfied by the output of the GaussSieve algorithm described in [24]. As in the paper describing the GaussSieve algorithm [24] the size of $\mathrm{P}(\mathscr{L})$ can actually be bounded by the kissing number τ_n. Following the same arguments as in [24] we can argue that in practice we expect $\mathrm{P}(\mathscr{L}) \approx 2^{0.21n}$ which is a factor 2 smaller in the exponent than the provable bound $|\mathrm{P}(\mathscr{L})| < 2^{0.402n}$.

A result that might be of interest in the search for lattices reaching the kissing number is the following.

Theorem 8 (Lattices achieving the kissing number) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. If the lattice $\mathscr{L}$ is such that it reaches the kissing number τ_n then $S_1(\mathscr{L}) = \mathrm{Irr}(\mathscr{L})$.*

Proof As the lattice $\mathscr{L}$ reaches the kissing number τ_n, that implies $|S_1(\mathscr{L})| = \tau_n$. By Proposition 8 we can conclude that the angle between any two vectors in $\mathrm{P}(\mathscr{L})$ is at least $\pi/3$. This is also the minimal possible angle between the centers of two equal n-dimensional spheres which touch another central sphere of the same size without intersecting. Hence $|\mathrm{P}(\mathscr{L})| \leq \tau_n$. Combining this with $S_1(\mathscr{L}) \subseteq \mathrm{P}(\mathscr{L})$ and $|S_1(\mathscr{L})| = \tau_n$ implies that $\mathrm{P}(\mathscr{L}) = S_1(\mathscr{L})$. As the set $\mathrm{P}(\mathscr{L})$ was build from classes of $\mathrm{Irr}(\mathscr{L})$ and we showed that it actually contains only vectors of norm $\lambda_1(\mathscr{L})$ that means that there is only one class in $\mathrm{Irr}(\mathscr{L})/\sim$, namely the class of $S_1(\mathscr{L})$. But in this class there is no pair of vectors that adds to a shorter one, thus the whole class is included in $\mathrm{P}(\mathscr{L})$. That implies that $\mathrm{Irr}(\mathscr{L}) = \mathrm{P}(\mathscr{L}) = S_1(\mathscr{L})$. □

Remark 13 A similar result for the set $\mathrm{R}(\mathscr{L})$ is not possible. For example for the root lattice E_8 reaching the kissing number in dimension 8 it holds that $S_1(E_8) = \mathrm{R}(E_8)$ but for the Leech lattice Λ_{24} it holds that $S_1(\Lambda_{24}) \subsetneq \mathrm{R}(\Lambda_{24})$ (see [8]).

5 Computation of the set $\mathrm{P}(\mathscr{L})$

In the previous sections we investigated some properties of the set $\mathrm{P}(\mathscr{L})$ and its relation to the set $\mathrm{R}(\mathscr{L})$. Ultimately we aim for using it in the study of lattice problems instead of $\mathrm{R}(\mathscr{L})$

due to its provably smaller cardinality (see Sect. 6). However, in order to actually benefit from this replacement an algorithm that computes $\mathrm{P}(\mathscr{L})$ without using the set $\mathrm{R}(\mathscr{L})$ is needed. The goal of this section is to examine ways of computing the set $\mathrm{P}(\mathscr{L})$.

5.1 The "brute force" approach

If the set $\mathrm{Irr}(\mathscr{L})$ is given then the set $\mathrm{P}(\mathscr{L})$ can be computed by means of a graph-based technique already described in Remark 9 and further analysed in Appendix B. Thus, it suffices to describe an algorithm which computes the set $\mathrm{Irr}(\mathscr{L})$. The naive approach is to use the fact that $\mathrm{Irr}(\mathscr{L}) \subseteq \mathrm{R}(\mathscr{L})$. Hence, as a first step one can run the algorithm described in [23] in order to get the set $\mathrm{R}(\mathscr{L})$. Then having a superset of $\mathrm{Irr}(\mathscr{L})$ it suffices to remove all the reducible vectors from it. This can be done by iterating through $\mathrm{R}(\mathscr{L})$ and checking for each $\vec{r} \in \mathrm{R}(\mathscr{L})$ if there exists a $\vec{v} \in \mathrm{R}(\mathscr{L})$ such that $\|\vec{v}\| < \|\vec{r}\|$ and $\|\vec{r} - \vec{v}\| < \|\vec{r}\|$. If $\vec{r} \in \mathrm{Irr}(\mathscr{L})$ then by definition there will not exist a vector $\vec{v} \in \mathscr{L}$ such that $\|\vec{v}\| < \|\vec{r}\|$ and $\|\vec{r} - \vec{v}\| < \|\vec{r}\|$ and thus the algorithm will not discard any of the irreducible vectors. If the vector $\vec{r}$ is reducible then we need the following heuristic assumption.

Assumption 1 (Witness of reducibility) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$ with $\mathrm{Irr}(\mathscr{L}) \neq \mathrm{R}(\mathscr{L})$. If $\vec{r} \in \mathrm{R}(\mathscr{L}) \setminus \mathrm{Irr}(\mathscr{L})$ then $\exists \vec{v} \in \mathrm{R}(\mathscr{L})$ such that $\|\vec{v}\| < \|\vec{r}\|$ and $\|\vec{r} - \vec{v}\| < \|\vec{r}\|$.

Heuristic assumption 1 can be considered as the analogue of Lemma 1 for the set $\mathrm{Irr}(\mathscr{L})$. Lemma 1 guaranteed that for every non-relevant vector there would exist a relevant vector acting as a "witness" of "non-relevancy". Heuristic assumption 1 speculates that for every reducible relevant vector there exists a relevant vector acting as a "witness" of reducibility. This claim can be further supported by the heuristic expectation for the set $\mathrm{R}(\mathscr{L})$ to include most of the "short" lattice vectors. Some experimental support can be derived for low dimensional lattices from Table 1 and Fig. 3(a).

Remark 14 Under Heuristic assumption 1 and the result of [23] on computing the set $\mathrm{R}(\mathscr{L})$ we can conclude that computing the set $\mathrm{Irr}(\mathscr{L})$ by "brute-force" can take up to $\tilde{O}(2^{2n})$–time and $\tilde{O}(2^n)$–space. This complexity can serve as an upper bound on the computation of the set $\mathrm{Irr}(\mathscr{L})$. Combining this observation with the discussion in Appendix B can give an upper bound in the complexity of computing $\mathrm{P}(\mathscr{L})$. Namely, for lattices which are not extremely structured (i.e. $\max_{S \in \mathrm{Irr}(\mathscr{L})/\sim} |S| = \mathrm{poly}(n)$) we can conclude that computing $\mathrm{P}(\mathscr{L})$ from $\mathrm{Irr}(\mathscr{L})$ can take $O(\mathrm{poly}(n)|\mathrm{Irr}(\mathscr{L})|)$ time. Therefore the computation of $\mathrm{Irr}(\mathscr{L})$ dominates the time complexity, leading to an overall upper bound for $\mathrm{P}(\mathscr{L})$ of $\tilde{O}(2^{2n})$–time.

However the approach in the next section could offer a better performance.

5.2 Using the GaussSieve/MinkowskiSieve algorithms

As it was already mentioned in Sect. 4.2, it is expected that the output of the GaussSieve algorithm [24] will be closely related to a set $\mathrm{P}(\mathscr{L})$. This expectation was motivated by the fact that both sets, $\mathrm{P}(\mathscr{L})$ and the output of the GaussSieve, possess the property $\min\{\|\vec{v}_1 \pm \vec{v}_2\|\} \geq \max\{\|\vec{v}_1\|, \|\vec{v}_2\|\}$ for any pair of $\vec{v}_1 \neq \pm\vec{v}_2$ in the set. At this point it should be clarified that for our purposes we will consider a slightly modified version of the GaussSieve algorithm which will be described here.[2]

For our purposes we will use the GaussSieve algorithm 5.2 but with the modified version of the GaussReduce function 2. In this way the following conditions are met.

[2] This version is the one used in [33] as well.

Algorithm 1 The GaussSieve algorithm as described in [24]

Require: A basis $\vec{\mathrm{B}}$ of a lattice $\mathscr{L}(\vec{\mathrm{B}})$ and a $c > 0$.
Ensure: A list $L \subset \mathscr{L}$ s.t. $\min\{\|\vec{v}_1 \pm \vec{v}_2\|\} \geq \max\{\|\vec{v}_1\|, \|\vec{v}_2\|\}$ for all $\vec{v}_1, \vec{v}_2 \in L$.

function GaussSieve($\vec{\mathrm{B}}$, c)
  $L \leftarrow \{\vec{0}\}$, $S \leftarrow \{\}$, $K \leftarrow 0$
  while $K < c$ **do**
    if S is not empty **then**
      $\vec{v}_{\text{new}} \leftarrow$ S.pop()
    else
      $\vec{v}_{\text{new}} \leftarrow$ SampleGaussian($\vec{\mathrm{B}}$)
    end if
    $\vec{v}_{\text{new}} \leftarrow$ GaussReduce($\vec{v}_{\text{new}}, L, S$)
    if $\vec{v}_{\text{new}} = \vec{0}$ **then**
      $K \leftarrow K + 1$
    else
      $L \leftarrow L \cup \{\vec{v}_{\text{new}}\}$
    end if
  end while
end function

function GaussReduce($\vec{p}, L, S$)
  while $\exists \vec{v}_i \in L : \|\vec{v}_i\| \leq \|\vec{p}\| \wedge \|\vec{p} - \vec{v}_i\| \leq \|\vec{p}\|$ **do**
    $\vec{p} \leftarrow \vec{p} - \vec{v}_i$
  end while
  while $\exists \vec{v}_i \in L : \|\vec{v}_i\| > \|\vec{p}\| \wedge \|\vec{v}_i - \vec{p}\| \leq \|\vec{v}_i\|$ **do**
    $L \leftarrow L \setminus \{\vec{v}_i\}$
    S.push($\vec{v}_i - \vec{p}$)
  end while
  return $\vec{p}$
end function

Algorithm 2 The modified GaussReduce function

1: **function** PrimeGaussReduce($\vec{p}, L, S$)
2: **while** $\exists \vec{v}_i \in L : \|\vec{v}_i\| \leq \|\vec{p}\| \wedge \|\vec{p} \pm \vec{v}_i\| < \|\vec{p}\|$ **do**
3: $\vec{p} \leftarrow \vec{p} \pm \vec{v}_i$
4: **end while**
5: **while** $\exists \vec{v}_i \in L : \|\vec{v}_i\| > \|\vec{p}\| \wedge \|\vec{v}_i \pm \vec{p}\| < \|\vec{v}_i\|$ **do**
6: $L \leftarrow L \setminus \{\vec{v}_i\}$
7: S.push($\vec{v}_i \pm \vec{p}$)
8: **end while**
9: **return** $\vec{p}$
10: **end function**

(i) Any irreducible vector which has been added to the GaussSieve list L will never be removed from it.
(ii) Any irreducible vector encountered by the algorithm will be added to L provided that it can extend its class representative set already in L.

Lemma 5 (Modified GaussSieve algorithm) *The* GaussSieve *algorithm* 5.2 *equipped with the function PrimeGaussReduce* (*Algorithm* 2) *satisfies both properties* (i) *and* (ii).

Proof (Property *i*) The only way for a vector $\vec{v}_i \in L$ to be removed from the list L is by entering the **while** loop in line 5 of the PrimeGaussReduce function. Let $\vec{v}_i \in L$ and also $\vec{v}_i \in \mathrm{Irr}(\mathscr{L})$. In order for the algorithm to remove $\vec{v}_i$ from L it should encounter another vector $\vec{p}$ such that $\|\vec{v}_i\| > \|\vec{p}\|$ and $\|\vec{v}_i - \vec{p}\| < \|\vec{v}_i\|$ or $\|\vec{v}_i\| > \|-\vec{p}\|$ and $\|\vec{v}_i + \vec{p}\| < \|\vec{v}_i\|$ which contradicts the irreducibility of $\vec{v}$.
(Property *ii*) Assume that the function PrimeGaussReduce is called and in some iteration of the **while** loop in line 2, $\vec{p}$ becomes such that $\vec{p} \in \mathrm{Irr}(\mathscr{L})$. In order for $\vec{p}$ to not be added in L this would mean that $\vec{p}$ could be further modified by the **while** loop in line 2. Thus the algorithm should encounter another vector $\vec{v}_i \in L$ such that $\|\vec{v}_i\| \leq \|\vec{p}\|$ and $\|\vec{p} \pm \vec{v}_i\| < \|\vec{p}\|$. The case where $\|\vec{v}_i\| < \|\vec{p}\|$ and $\|\vec{p} \pm \vec{v}_i\| < \|\vec{p}\|$ violates the irreducibility of $\vec{p}$ and thus can be disregarded. This leaves only one possible case, namely $\|\vec{v}_i\| = \|\vec{p}\|$ and $\|\vec{p} \pm \vec{v}_i\| < \|\vec{p}\|$. This condition implies that $\vec{v}_i$ and $\vec{p}$ belong to the same equivalence

class and they are adjacent. Therefore this pair of vectors cannot belong to any set $\mathrm{P}(\mathscr{L})$ of $\mathscr{L}$. Hence $\vec{p}$ should not be included in L anyway and the algorithm correctly further reduces it. □

Remark 15 If the PrimeGaussReduce function in line 5 was the same as in the original GaussReduce, then the algorithm could encounter an instance where it would enter the loop with $\|\vec{v}_i\| > \|\vec{p}\|$, $\|\vec{v}_i - \vec{p}\| = \|\vec{v}_i\|$ and $\vec{v}_i, \vec{v}_i - \vec{p} \in \mathrm{Irr}(\mathscr{L})$. This could be possible if an equivalence class in $\mathrm{Irr}(\mathscr{L})$ was not trivial. In this case the algorithm would remove the vector $\vec{v}_i$ from the list and add its equivalent $\vec{v}_i - \vec{p}$ to S. As a result for these non-trivial classes the algorithm could behave in a bad way by repetitively removing and adding representatives of the same class.

Remark 16 If the PrimeGaussReduce function in line 2 was the same as in the original GaussReduce, then the algorithm could encounter an instance where it would enter the loop with $\|\vec{v}_i\| \leq \|\vec{p}\|$, $\|\vec{p} - \vec{v}_i\| = \|\vec{p}\|$ and $\vec{p}, \vec{p} - \vec{v}_i \in \mathrm{Irr}(\mathscr{L})$. Thus, $\vec{p}$ and $\vec{p} - \vec{v}_i$ are equivalent. In case $\|\vec{v}_i\| < \|\vec{p}\|$ then $\vec{p}$ and $\vec{p} - \vec{v}_i$ are also adjacent in the class graph and therefore in this case the algorithm would cycle through the adjacent vectors of $\vec{p}$. Therefore there is no need to perform a reduction in this case. In case $\|\vec{v}_i\| = \|\vec{p}\|$ then all three $\vec{p}, \vec{v}_i, \vec{p} - \vec{v}_i$ are equivalent but not adjacent. Hence in this case the algorithm does not make any progress by replacing $\vec{p}$ by $\vec{p} - \vec{v}_i$. Thus, there is no need to perform a reduction in this case as well. [3]

We consider the GaussSieve algorithm 5.2 equipped with the PrimeGaussReduce function (Algorithm 2). As stated in the description of GaussSieve algorithm 5.2, it terminates after reaching a number of c "collisions" (i.e. reductions to the zero vector). If for one run of the algorithm we let c tend to infinity then its list L will converge to a specific list of vectors as output. We denote by $\mathrm{GaussSieve}(\mathscr{L})$ such a list of vectors created by GaussSieve and possessing the property that it cannot be further modified by the algorithm. In the sequel, when we will refer to the convergence of the output of a sieving algorithm, it will be according to this notion of convergence.

In order to relate the sets $\mathrm{GaussSieve}(\mathscr{L})$ and $\mathrm{P}(\mathscr{L})$ we give the following definition.

Definition 10 (Partitioning $\mathrm{P}(\mathscr{L})$ by sign) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. Given a $\mathrm{P}(\mathscr{L}) \subseteq \mathrm{Irr}(\mathscr{L})$ we define $\mathrm{P}^+(\mathscr{L})$ and $\mathrm{P}^-(\mathscr{L})$ to be a partition of $\mathrm{P}(\mathscr{L})$ according to sign.

In other words, we take for $\mathrm{P}^+(\mathscr{L})$ some subset of $\mathrm{P}(\mathscr{L})$ such that of each pair $\pm\vec{v} \in \mathrm{P}^+(\mathscr{L})$ exactly one is in $\mathrm{P}^+(\mathscr{L})$. Of course, there are many choices for $\mathrm{P}^+(\mathscr{L})$ and $\mathrm{P}^-(\mathscr{L})$, any one will do.

Even though the output of GaussSieve converges to a set which is maximal in $\mathscr{L}$ under the property $\min\{\|\vec{v}_1 \pm \vec{v}_2\|\} \geq \max\{\|\vec{v}_1\|, \|\vec{v}_2\|\}$, the same is not true in general for the set $\mathrm{P}^+(\mathscr{L})$ as shown by experiments (Table 1). In particular, we can conclude by Lemma 5 that if we allow this modified version of the GaussSieve to run long enough i.e. it samples "enough" vectors, then the output will converge to a set $\mathrm{GaussSieve}(\mathscr{L})$, which will contain a $\mathrm{P}^+(\mathscr{L})$.

Hence we cannot claim that the output of GaussSieve converges to a set $\mathrm{P}^+(\mathscr{L})$ but only to a superset of it. The fact that a $\mathrm{P}^+(\mathscr{L})$ is not maximal in $\mathscr{L}$ under the property $\min\{\|\vec{v}_1 \pm \vec{v}_2\|\} \geq \max\{\|\vec{v}_1\|, \|\vec{v}_2\|\}$ implies the existence of vectors which are not irreducible but they also cannot be reduced by any of the vectors in $\mathrm{P}(\mathscr{L})$.

[3] However, as $\vec{p} - \vec{v}_i$ is not adjacent to both $\vec{p}$ and $\vec{v}_i$ an option could be to move $\vec{p} - \vec{v}_i$ to the stack S for further consideration.

The definition below of the set $\mathrm{P}_2(\mathscr{L})$ will help us in bounding the output of the GaussSieve algorithm. Also, the definition of the sets $\mathrm{P}_k(\mathscr{L})$ for $k > 2$ will help us in bounding the output of modified versions of "higher" sieving algorithms like the Triple and Quadruple MinkowskiSieve, described in [3].

Definition 11 (Pairwise irreducible system) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. Given a $\mathrm{P}(\mathscr{L}) \subseteq \mathrm{Irr}(\mathscr{L})$ we define

$$\mathrm{P}_2(\mathscr{L}) := \{\vec{v} \in \mathscr{L} \mid \nexists \vec{p} \in \mathrm{P}(\mathscr{L}) \text{ with } \|\vec{p}\| < \|\vec{v}\| \text{ and } \|\vec{v} - \vec{p}\| < \|\vec{v}\|\}.$$

A first remark on this definition is that as $\mathrm{P}(\mathscr{L}) \subseteq \mathscr{L}$ also $\mathrm{P}(\mathscr{L}) \subseteq \mathrm{P}_2(\mathscr{L})$. The output of the (modified) GaussSieve converges to a set GaussSieve$(\mathscr{L})$ including a set $\mathrm{P}^+(\mathscr{L})$. Therefore, every $\vec{v} \in \text{GaussSieve}(\mathscr{L})$ cannot be reduced by any $\vec{p} \in \mathrm{P}^+(\mathscr{L})$ and as GaussSieve$(\mathscr{L}) \subseteq \mathscr{L}$ we can conclude that GaussSieve$(\mathscr{L})$ can be bounded as follows:

$$\mathrm{P}^+(\mathscr{L}) \subseteq \text{GaussSieve}(\mathscr{L}) \subseteq \mathrm{P}_2(\mathscr{L}) \tag{16}$$

Under this set inequality GaussSieve$(\mathscr{L})$ can be viewed in the following way. A set GaussSieve$(\mathscr{L})$ can be considered as the closure of a $\mathrm{P}^+(\mathscr{L})$ in $\mathrm{P}_2(\mathscr{L})$ under the property of Gauss-reduction. In more detail GaussSieve$(\mathscr{L})$ can be viewed as the minimal (according to included vector norms) subset of a $\mathrm{P}_2(\mathscr{L})$ including $\mathrm{P}^+(\mathscr{L})$ and being a maximal subset of $\mathrm{P}_2(\mathscr{L})$ with the property of Gauss-reduction (i.e. $\min\{\|\vec{v}_1 \pm \vec{v}_2\|\} \geq \max\{\|\vec{v}_1\|, \|\vec{v}_2\|\}$).

Remark 17 It is unclear if the set $\mathrm{P}_2(\mathscr{L})$ is a finite or an infinite set.

Definition 12 (k-wise irreducible system) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$ and $k \in \mathbb{N}$ with $k \geq 2$. Given a $\mathrm{P}(\mathscr{L}) \subseteq \mathrm{Irr}(\mathscr{L})$ we define

$$\mathrm{P}_{k+1}(\mathscr{L}) := \{\vec{v} \in \mathrm{P}_k(\mathscr{L}) \mid \nexists \vec{p} \in \mathrm{P}^{(k)}(\mathscr{L}) \text{ with } \|\vec{p}\| < \|\vec{v}\| \text{ and } \|\vec{v} - \vec{p}\| < \|\vec{v}\|\}$$

where $\mathrm{P}^{(k)}(\mathscr{L})$ is defined as

$$\bigcup_{i=1}^{\lfloor k/2 \rfloor} \big\{\vec{v}_1 + \vec{v}_2 \mid \vec{v}_1 \in \mathrm{P}^{(i)}(\mathscr{L}),\ \vec{v}_2 \in \mathrm{P}^{(k-i)}(\mathscr{L}) \text{ and } \|\vec{v}_j\| < \|\vec{v}_1 + \vec{v}_2\|,\ \|\vec{v}_l\| \leq \|\vec{v}_1 + \vec{v}_2\| \text{ where } j, l \in \{1, 2\}\big\}$$

and $\mathrm{P}^{(1)}(\mathscr{L}) := \mathrm{P}(\mathscr{L})$.

Lemma 6 (Relating k-wise irreducible systems) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$ and $\mathrm{P}(\mathscr{L})$ be a subset of $\mathrm{Irr}(\mathscr{L})$. Then for the sequence $\mathrm{P}_k(\mathscr{L})$ given in Definition 12 it holds that*

(i) $\mathrm{P}_{k+1}(\mathscr{L}) \subseteq \mathrm{P}_k(\mathscr{L})$ *for every* $k \geq 2$.
(ii) $\lim_{k\to\infty} \mathrm{P}_k(\mathscr{L}) = \mathrm{Irr}(\mathscr{L})$.

So, in one line:

$$\mathrm{P}_2(\mathscr{L}) \supseteq \ldots \supseteq \mathrm{P}_k(\mathscr{L}) \supseteq \mathrm{P}_{k+1}(\mathscr{L}) \supseteq \ldots \supseteq \mathrm{Irr}(\mathscr{L}).$$

Proof First of all, as we chose a random but fixed $\mathrm{P}(\mathscr{L}) \subseteq \mathrm{Irr}(\mathscr{L})$ the sets $\mathrm{P}_k(\mathscr{L})$ are well-defined. Part *(i)* of the lemma is an immediate consequence of the definition of $\mathrm{P}_k(\mathscr{L})$. Initially we show that $\mathrm{Irr}(\mathscr{L}) \subseteq \mathrm{P}_k(\mathscr{L})$ for every $k \geq 2$. This follows directly by the definition of $\mathrm{P}_k(\mathscr{L})$ and the fact that $\mathrm{P}^{(k)}(\mathscr{L}) \subseteq \mathscr{L}$. By the (recursive) definition of $\mathrm{P}_k(\mathscr{L})$ it follows that it includes all vectors $\vec{v} \in \mathscr{L}$ such that they can not be reduced by any shorter vector in $\cup_{i=1}^{k-1} \mathrm{P}^{(i)}(\mathscr{L})$. Thus for part *(ii)* of the lemma it suffices to show that $\lim_{k\to\infty} \cup_{i=1}^{k} \mathrm{P}^{(i)}(\mathscr{L}) =$

$\mathscr{L}$. As $\mathrm{P}^{(i)}(\mathscr{L}) \subseteq \mathscr{L}$ for every $i \geq 1$ it follows that $\lim_{k\to\infty} \cup_{i=1}^{k} \mathrm{P}^{(i)}(\mathscr{L}) \subseteq \mathscr{L}$. It is only left proving the converse inequality. Let $\vec{v} \in \mathscr{L}$, it suffices to show that $\exists k \geq 1$ such that $\vec{v} \in \mathrm{P}^{(k)}(\mathscr{L})$.

A vector $\vec{v} \in \mathscr{L}$ can be repeatedly reduced as in the proof of Proposition 5 until it is written as a sum $\vec{v} = \sum_{i=1}^{l} \vec{p}_i$ of shorter vectors $\vec{p}_i \in \mathrm{Irr}(\mathscr{L})$ for some $l \geq 1$. This decomposition satisfies the recursive condition implied by the definition of the $\mathrm{P}^{(k)}(\mathscr{L})$. If all the vectors $\vec{p}_i \in \mathrm{Irr}(\mathscr{L})$ actually belong to $\mathrm{P}(\mathscr{L})$ then $\vec{v} \in \mathrm{P}^{(l)}(\mathscr{L})$ and we are done. If there exists some $\vec{p}_i \in \mathrm{Irr}(\mathscr{L}) \setminus \mathrm{P}(\mathscr{L})$ then $\vec{p}_i = \tilde{\vec{p}}_i + \vec{p}_i'$ where $\tilde{\vec{p}}_i \in \mathrm{P}(\mathscr{L})$ and $\|\vec{p}_i'\| < \|\vec{p}_i\|$, $\|\vec{p}_i\| = \|\tilde{\vec{p}}_i\|$ by the definition of $\mathrm{P}(\mathscr{L})$. Thus, $\vec{p}_i'$ can be further get decomposed into shorter vectors (like $\vec{v}$) and as $\|\vec{p}_i'\| < \|\vec{p}_i\|$ progress was made which implies that this decomposition will finish after finitely many steps as there is a finite number of lattice points in $\mathscr{B}(\vec{0}, \|\vec{v}\|)$. Therefore $\vec{v}$ can be repeatedly reduced until it is written as a sum of vectors in $\mathrm{P}(\mathscr{L})$, concluding the proof. □

We are now going to describe the "higher" sieving algorithms which we will consider. We have already mentioned the Triple and the Quadruple MinkowskiSieve described in [3]. The difference between the GaussSieve algorithm and these higher ones lies in the reduction function. Hence, if we equip Algorithm 5.2 with function PrimeMinkowskiReduce (Algorithm 3), we get the modified MinkowskiSieve which we are interested in.

The modification compared to the description in [3] appears in lines 10 and 21 of Algorithm 3, where the extra conditions $\|\vec{w}\| \leq \|\vec{p}\|$ and $\|\vec{w}\| < \|\vec{v}_{k-1}\|$ respectively are added. By adding these conditions it is guaranteed to get an output list which will satisfy properties (i) and (ii) like in Lemma 5 for the GaussSieve. Hence, based on these properties it can be concluded that the output list of vectors will again contain a set $\mathrm{P}^{+}(\mathscr{L})$. In order to ease our exposition we set the following notation.

Let $k \in \mathbb{N}$ with $k \geq 2$. We consider the k-MinkowskiSieve algorithm equipped with the function PrimeMinkowskiReduce (Algorithm 3). We denote by $\mathrm{MinkowskiSieve}_k(\mathscr{L})$ a list of vectors L created by this algorithm and possessing the property that L cannot be further modified by the algorithm. Note that for $k = 2$ one has $\mathrm{MinkowskiSieve}_2(\mathscr{L}) = \mathrm{GaussSieve}(\mathscr{L})$.

Remark 18 The output of the modified k-MinkowskiSieve algorithm will not be a list of vectors which will be k-Minkowski-reduced if $k > 2$ (for the Minkowski-reduced definition see [27]). If this was desired, then the lines 10 and 21 of Algorithm 3 should be modified in order to allow reductions by longer vectors as well. For a k-Minkowski-reduced list with $k > 4$ lines 9,10 and 20,21 of Algorithm 3 should also allow for the coefficients of the vectors $\vec{v}_i$, $\vec{p}$ and $\vec{v}_{k-1}$ to take more values than ± 1 (see for example [27, Theorem 2.2.2]).

The "higher" sieving algorithms which we considered by making the generalisation from the GaussSieve towards the MinkowskiSieve will contribute towards an asymptotic computational argument. But first we state a heuristic assumption which we will use.

Assumption 2 (Convergence of the MinkowskiSieve) Consider the k-MinkowskiSieve algorithm equipped with the function PrimeMinkowskiReduce (Algorithm 3). Then the output of this algorithm will converge to a set $\mathrm{MinkowskiSieve}_k(\mathscr{L})$.

Remark 19 Heuristic assumption 2 actually claims that the k-MinkowskiSieve does not diverge or enter an infinite loop. The experimental results in Sect. 5.3 (see Fig. 3) indicate that for $k \in \{2, 3, 4\}$ this seems to be a valid assumption. However, this is the only argument we have in favour of this assumption. We leave the investigation for concrete arguments supporting this heuristic assumption as an open problem for future research.

Algorithm 3 The modified MinkowskiReduce function

function PRIMEMINKOWSKIREDUCE($\vec{p}, L, S, k$)
  loop = true
  while loop **do**
    loop = false
    if $k > 2$ **then**
      PRIMEMINKOWSKIREDUCE($\vec{p}, L, S, k-1$)
    end if
    for all $\{\vec{v}_1, \ldots, \vec{v}_{k-1}\} \subset L$ s.t. $\|\vec{v}_i\| \leq \|\vec{p}\|$ **do**
      for all $\vec{w} \in \left\{\sum_{i=1}^{k-1} (-1)^{a_i} \vec{v}_i \mid a_i \in \{0,1\}\right\}$ **do**
        if $\|\vec{w}\| \leq \|\vec{p}\|$ and $\|\vec{p} - \vec{w}\| < \|\vec{p}\|$ **then**
          $\vec{p} \leftarrow \vec{p} - \vec{w}$
          loop = true
          goto next
        end if
      end for
    end for
    next:
  end while
  for all $\{\vec{v}_1, \ldots, \vec{v}_{k-1}\} \subset L$ with $\|\vec{v}_i\| \leq \|\vec{v}_{i+1}\|$ and s.t. $\|\vec{v}_{k-1}\| > \|\vec{p}\|$ **do**
    for all $\vec{w} \in \left\{(-1)^{a_0}\vec{p} + \sum_{i=1}^{k-2} (-1)^{a_i} \vec{v}_i \mid a_i \in \{0,1\}\right\}$ **do**
      if $\|\vec{w}\| < \|\vec{v}_{k-1}\|$ and $\|\vec{v}_{k-1} - \vec{w}\| < \|\vec{v}_{k-1}\|$ **then**
        $L \leftarrow L \setminus \{\vec{v}_{k-1}\}$
        S.push($\vec{v}_{k-1} - \vec{w}$)
      end if
    end for
  end for
  return $\vec{p}$
end function

Theorem 9 (Shape of *MinkowskiSieve*$_k(\mathscr{L})$) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. We consider the k-*MinkowskiSieve *algorithm equipped with the function PrimeMinkowskiReduce. Under Heuristic Assumption 2, as k increases the set* MinkowskiSieve$_k(\mathscr{L})$ *converges to a set* $\mathrm{P}^+(\mathscr{L})$.

Proof In order to simplify the proof and avoid ambiguities we make the following convention. Both sets $\mathrm{P}^+(\mathscr{L})$ and MinkowskiSieve$_k(\mathscr{L})$ are defined/constructed in such a way that for a vector $\vec{v}$ only one of $\pm\vec{v}$ belongs to the set. This allows many possible choices for these sets. In order to avoid this kind of ambiguities we make the convention that a vector $\vec{v}$ is included in the aforementioned sets only if its first non-zero coordinate is positive.

Initially we will prove that for every $k \geq 2$ there exists a set $\mathrm{P}^+(\mathscr{L})$ and a set $\mathrm{P}_k(\mathscr{L})$ such that

$$\mathrm{P}^+(\mathscr{L}) \subseteq \mathrm{MinkowskiSieve}_k(\mathscr{L}) \subseteq \mathrm{P}_k(\mathscr{L}). \tag{17}$$

Let $k \geq 2$ and MinkowskiSieve$_k(\mathscr{L})$ be the converging set of an execution of the k-MinkowskiSieve. As mentioned before, we can transfer Lemma 5 from the case of GaussSieve to the k-MinkowskiSieve algorithm described in this section. This implies that for every MinkowskiSieve$_k(\mathscr{L})$ there will exist a set $\mathrm{P}^+(\mathscr{L})$ such that $\mathrm{P}^+(\mathscr{L}) \subseteq$ MinkowskiSieve$_k(\mathscr{L})$. We fix this set $\mathrm{P}^+(\mathscr{L})$.

Let $\vec{v} \in$ MinkowskiSieve$_k(\mathscr{L})$ and $\vec{p}_1, \ldots, \vec{p}_{k-1} \in \mathrm{P}^+(\mathscr{L})$ with $\|\vec{p}_i\| < \|\vec{v}\|$. As the set MinkowskiSieve$_k(\mathscr{L})$ is k-reduced according to the notion implied by Algorithm 3 we can conclude that $\vec{v}$ cannot be reduced by any vector of the form $\sum_{i=1}^{l} (-1)^{a_i} \vec{p}_i$ for $1 \leq l \leq k-1$. As the vectors $\vec{p}_1, \ldots, \vec{p}_{k-1}$ belong to the set MinkowskiSieve$_k(\mathscr{L})$ as well, they are $k-1$-

reduced. This in turn implies that the vectors of the form $\sum_{i=1}^{l}(-1)^{a_i}\vec{p}_i$ belong to the set $\mathrm{P}^{(l)}(\mathscr{L})$ for $1 \leq l \leq k-1$. This holds for any tuple of l vectors in $\mathrm{P}^{+}(\mathscr{L})$. Hence, the set of vectors emerging from the union of all $\{\sum_{i=1}^{l}(-1)^{a_i}\vec{p}_i\}$ will be exactly $\mathrm{P}^{(l)}(\mathscr{L})$. This implies that $\vec{v}$ cannot be reduced by any vector in $\cup_{i=1}^{k-1}\mathrm{P}^{(i)}(\mathscr{L})$. This is equivalent to the condition a vector $\vec{v}$ has to satisfy according to definition 12 in order to belong to $\mathrm{P}_k(\mathscr{L})$. Thus we can conclude that $\mathrm{MinkowskiSieve}_k(\mathscr{L})$ is included in the $\mathrm{P}_k(\mathscr{L})$ implied by the set $\mathrm{P}(\mathscr{L}) = \mathrm{P}^{+}(\mathscr{L}) \cup (-\mathrm{P}^{+}(\mathscr{L}))$. This concludes the first part of the proof.

For the second part of the proof we distinguish between the cases of $\mathrm{Irr}(\mathscr{L}) = \mathrm{P}(\mathscr{L})$ and $\mathrm{Irr}(\mathscr{L}) \neq \mathrm{P}(\mathscr{L})$.

If it holds that $\mathrm{Irr}(\mathscr{L}) = \mathrm{P}(\mathscr{L})$ then apart from $\mathrm{P}(\mathscr{L})$ being uniquely determined the same holds for the sets $\mathrm{P}_k(\mathscr{L})$. Hence, for every $k \geq 2$ the boundary sets in (17) are uniquely determined. This enables a direct use of Lemma 6. As k increases the set $\mathrm{MinkowskiSieve}_k(\mathscr{L})$ will be contained to even smaller and smaller sets $\mathrm{P}_k(\mathscr{L})$ which converge to $\mathrm{Irr}(\mathscr{L})$ according to (i) and (ii) of Lemma 6. Therefore for the limit case it could be stated that

$$\mathrm{P}^{+}(\mathscr{L}) \subseteq \lim_{k\to\infty} \mathrm{MinkowskiSieve}_k(\mathscr{L}) \subseteq \mathrm{Irr}(\mathscr{L}). \tag{18}$$

But we assumed $\mathrm{Irr}(\mathscr{L}) = \mathrm{P}(\mathscr{L})$ and thus we can conclude that

$$\lim_{k\to\infty} \mathrm{MinkowskiSieve}_k(\mathscr{L}) = \mathrm{P}^{+}(\mathscr{L}).$$

In order to finish the proof we have to deal with the case $\mathrm{Irr}(\mathscr{L}) \neq \mathrm{P}(\mathscr{L})$. In this case, the sets $\mathrm{P}^{+}(\mathscr{L})$ and $\mathrm{P}_k(\mathscr{L})$ used in inequality (17) are not uniquely determined and therefore Lemma 6 cannot be used directly. In Lemma 6 it was shown that given the sequence of $\mathrm{P}_k(\mathscr{L})$ implied by any $\mathrm{P}(\mathscr{L})$ then $\lim_{k\to\infty}\mathrm{P}_k(\mathscr{L}) = \mathrm{Irr}(\mathscr{L})$. Hence any $\mathrm{P}_k(\mathscr{L})$ belongs to a sequence converging to the same limit, $\mathrm{Irr}(\mathscr{L})$. Interchanging terms $(\mathrm{P}_k(\mathscr{L}))$ among these sequences does not affect their limit. Therefore, we can again use inequality (17) and "take limits" leading to a result like (18). We have to be careful though. The right hand-side limit (i.e. $\mathrm{Irr}(\mathscr{L})$) is well-defined but the left one can cycle over all choices of $\mathrm{P}^{+}(\mathscr{L})$. This is expected as the limit of the sequence $\mathrm{MinkowskiSieve}_k(\mathscr{L})$ as $k \to \infty$ is not unique but depends on the choice of representatives made for each non-trivial class of vectors. For convenience we assume that $\forall k > k_0$ for some k_0 this choice stabilises to some random but fixed choice. Thus, we have again reached inequality (18).

We examine the sets in inequality (18) according to the Gauss-reduced property. Let $k \geq 2$, the set $\mathrm{MinkowskiSieve}_k(\mathscr{L})$ is a set in which the output of the algorithm converges to and also possesses the Gauss-reduced property by construction. This holds for every $k \geq 2$ and thus transfers to the limit as well, as $k \to \infty$. The set $\mathrm{P}^{+}(\mathscr{L})$ is not a maximal subset of $\mathscr{L}$ satisfying the Gauss-reduced property but due to its construction it is maximal in the set $\mathrm{Irr}(\mathscr{L})$. Hence, inequality (18) and maximality of $\mathrm{P}^{+}(\mathscr{L})$ in $\mathrm{Irr}(\mathscr{L})$ imply the result. □

The conclusion in Theorem 9 is supported by the experimental results given in Table 1.

Remark 20 Theorem 9 describes asymptotic behaviour of the modified MinkowskiSieve algorithm with the goal of providing a faster way of computing sets $\mathrm{P}^{+}(\mathscr{L})$. Even though, asymptotically, the algorithm possesses the desired behaviour, this does not make it immediately a computational tool for $\mathrm{P}^{+}(\mathscr{L})$. There are two obstacles towards that goal. The first one is, given a lattice $\mathscr{L}$ in dimension n, to find for which $k \geq 2$ to run k-MinkowskiSieve. This k should not be too high in order to be computationally efficient to run the algorithm. The second problem is finding for how long this k-MinkowskiSieve should run in order to approximate well enough a set $\mathrm{MinkowskiSieve}_k(\mathscr{L})$.

5.3 Experimental results

In this section we provide some experimental results which support our claims in the previous subsections. In particular, as a first step we computed the sets $\mathrm{R}(\mathscr{L})$, $\mathrm{Irr}(\mathscr{L})$, $\mathrm{P}(\mathscr{L})$ for 10 lattices in dimension 20 and afterwards we computed the output of the GaussSieve, the Triple and the Quadruple MinkowskiSieve. In order to generate 10 lattices in dimension 20 we used the Sage computer algebra system [9]. In particular we used Sage's "Hard lattice generator" with the following choice of parameters,

sage.crypto.gen_lattice(type='random', n=1, m=20, q=10^42, seed=*seed*)

and 10 different values of *seed*. Initially, using the OpenMP parallel implementation build for the projects [5, 13] we computed the set of relevant vectors $\mathrm{R}(\mathscr{L})$ for each lattice. On top of this code (which the authors were so kind to provide us) we implemented the method described in Sect. 5.1 and computed the set $\mathrm{Irr}(\mathscr{L})$. As for our experiments the lattices used were generated randomly, they did not possess any specific structure and hence $\mathrm{P}(\mathscr{L}) = \mathrm{Irr}(\mathscr{L})$ for all of them. This part of the experiments was performed on a node of the Lisa cluster [32] with a 16-core CPU (2.10GHz) and 96 GB of RAM. The computation of the sets $\mathrm{R}(\mathscr{L})$ and $\mathrm{Irr}(\mathscr{L})$ using the aforementioned implementation and hardware took about 5.5 seconds per lattice.

Finally, by modifying the already existing sieve implementations in FPLLL [33] we computed the output of the GaussSieve, Triple and Quadruple MinkowskiSieve as described in Sect. 5.2 for the same 10 lattices. The modifications which we made to the already existing FPLLL implementations were:

- A vector is allowed to be reduced only by a shorter vector.
- The termination condition is changed to a fixed number of collisions: $5 \cdot 10^5$ for the GaussSieve and 10^5 for the Triple and Quadruple MinkowskiSieve. These numbers were chosen to ensure the created list by the algorithm remains unchanged for "many" iterations before the algorithm terminates. These choices seem to not be optimal according to our experimental data and could possibly be further reduced.

This part of the experiments was performed on a Lenovo X250 laptop with 4 Intel Core i3-5010U CPU (2.10GHz) and 8 GB of RAM. The output of these experiments is summarised in Table 1.

Table 1 motivates a number of remarks about the involved sets. Initially, the number of relevant vectors observed was indeed close to the expected number $2 \cdot (2^{20} - 1)$. Also, the sets $\mathrm{Irr}(\mathscr{L})$ and $\mathrm{P}(\mathscr{L})$ were equal in all 10 cases, as we had assumed for random lattices without any underlying structure. The size of $\mathrm{P}(\mathscr{L})$ (and $\mathrm{Irr}(\mathscr{L})$ in this case) was observed to be some orders of magnitude smaller than the size of $\mathrm{R}(\mathscr{L})$ making it more appealing to use in practice.

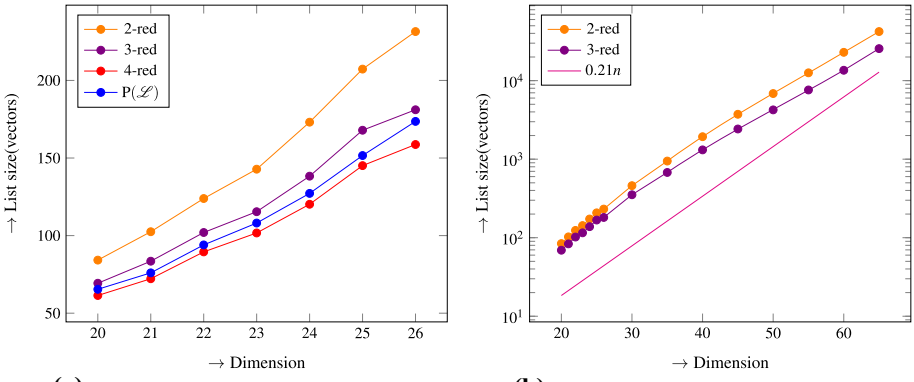
The right part of Table 1 justifies our idea to try and correlate the output of sieving algorithms with the set of irreducible vectors. Even though we cannot display here the lists of vectors which we computed but rather only their sizes, we observed the following behaviour. The list of vectors outputted by the GaussSieve contained the set $\mathrm{P}(\mathscr{L})$ in 8 out of the 10 cases and in the other two of them there was only 1 vector missing. This supports our claim that the output of GaussSieve converges to a superset of $\mathrm{P}(\mathscr{L})$. Also, as we moved to "higher" sieving algorithms like our modified version of the Triple and Quadruple MinkowskiSieve the output of the sieving algorithms approximated even closer the set $\mathrm{P}(\mathscr{L})$. Actually, it is not a coincidence that the numbers in the columns "4-red" and "$|\mathrm{Irr}(\mathscr{L})|$, $|\mathrm{P}(\mathscr{L})|$" in Table 1 differ only by a factor of 2, since the output of the Quadruple MinkowskiSieve in all 10 cases gave exactly a set $\mathrm{P}^+(\mathscr{L})$ as for every vector $\vec{v}$ it stores only one of $\pm\vec{v}$.

Table 1 The following tables describe the sizes of the lists involved in our experiments with 10 random lattices in dimension 20

Seed	$\lvert \mathrm{R}(\mathscr{L}) \rvert$	$\lvert \mathrm{Irr}(\mathscr{L}) \rvert$, $\lvert \mathrm{P}(\mathscr{L}) \rvert$
314	$2 \cdot 1048361$	$2 \cdot 66$
417	$2 \cdot 1048388$	$2 \cdot 70$
849	$2 \cdot 1048389$	$2 \cdot 68$
422	$2 \cdot 1048349$	$2 \cdot 67$
168	$2 \cdot 1048371$	$2 \cdot 60$
84	$2 \cdot 1048363$	$2 \cdot 64$
105	$2 \cdot 1048375$	$2 \cdot 62$
273	$2 \cdot 1048360$	$2 \cdot 60$
390	$2 \cdot 1048376$	$2 \cdot 66$
656	$2 \cdot 1048372$	$2 \cdot 71$

Seed	2-red	3-red	4-red
314	86	77	66
417	95	80	70
849	98	85	68
422	93	74	67
168	88	69	60
84	92	75	64
105	88	74	62
273	83	68	60
390	89	76	66
656	95	79	71

The first columns indicate the seed used for the generation of the lattice. The table on the top gives the sizes of the corresponding sets $\mathrm{R}(\mathscr{L})$, $\mathrm{Irr}(\mathscr{L})$ and $\mathrm{P}(\mathscr{L})$ for each lattice. The factor 2 is due to the sign symmetry. The table on the bottom shows the sizes of the lists generated by the modified GaussSieve, Triple and Quadruple MinkowskiSieve



(a) List sizes for dimensions 20–26. **(b)** List sizes for dimensions 20–65.

Fig. 3 Experimental results on the scaling of size of $\mathrm{P}(\mathscr{L})$ according to the dimension of $\mathscr{L}$. Each point in the graphs corresponds to the average value taken amongst 10 lattices. The labels k-red are used to indicate the output of the modified sieve algorithms described in this work and not the ones in the literature [3, 24] (Color figure online)

Another question which could be investigated experimentally is how the expected size of $\mathrm{P}(\mathscr{L})$ behaves as the dimension of $\mathscr{L}$ increases. In order to develop an intuition about this behaviour we performed a number of experiments in dimensions 20–65, the results of which are shown in Fig. 3. Likewise in our experiments in dimension 20 we used the modified OpenMP parallel implementation from [5, 13] and the modified sieve implementations in FPLLL [33]. For each dimension we depict the average value amongst 10 lattices. However, as in this case we dealt with higher dimensions we reduced the number of collisions in the termination condition of the sieve algorithms to

- GaussSieve: 10,000 collisions
- Triple MinkowskiSieve: 2500 collisions
- Quadruple MinkowskiSieve: 2000 collisions.

Therefore the results in Fig. 3 related to sieving algorithms should only be interpreted as approximations of the algorithm's converging set size. As we will discuss later, estimating the accuracy of this approximation is left for future research. Figure 3a illustrates the result of our experiments in dimensions 20–26. We believe that for these "smaller" dimensions the approximations are "more" accurate and that is why we show them separately. Another reason is that running the OpenMP Voronoi implementation beyond these dimensions has a substantial memory requirement (tens of GB).

Computing a least squares fit for the points in the blue curve (which indicates the correct expected values for $|\mathrm{P}(\mathscr{L})|$ under Assumption 1) gives the formula $2^{0.237n+1.286}$ which reasonably matches the heuristic expectation for the size of $\mathrm{P}(\mathscr{L})$, namely $2^{0.21n}$. Furthermore Fig. 3a reveals that the GaussSieve gives only a superset of $\mathrm{P}(\mathscr{L})$ even for small dimensions. The Triple and Quadruple MinkowskiSieve are much closer to the blue curve. The difference between the Triple and Quadruple MinkowskiSieve is that the one lies above the blue curve and the other below it. As we already observed in Table 1 the Triple MinkowskiSieve will probably remain above it. However the Quadruple MinkowskiSieve possess the potential to reach the "correct" curve asymptotically. Of course this could also be far from the truth for higher dimensions.

In order to put these curves more into perspective we created Fig. 3b which shows the average output sizes of the GaussSieve and Triple MinkowskiSieve for dimensions 20–65. We did not draw the curve of the Quadruple MinkowskiSieve as it also turns out to be quite time costly for dimensions higher than 30. At this point we must emphasize that the used modified sieving algorithms take more time in order to terminate due to the modifications which aim not in solving SVP but computing close approximations of $\mathrm{P}(\mathscr{L})$. For instance the modified Triple MinkowskiSieve in dimension 65 took on average 3 days in order to terminate for each lattice. However this is only the average observed time. Actually one of the ten lattices used proved to be an "easier case", terminating in under 2 h.

Even though these results provide some intuition on what kind of relation it could be expected between the set of irreducible vectors and sieving algorithms, they also raise some questions.

A first question which would be interesting is examining the termination condition for the sieving algorithm. In our experiments we made a specific choice on the number of collisions but this was done by trial and error and could be possibly improved. In other words, we ask for a termination condition, which if it is satisfied by a sieving algorithm (as used in this section) it guarantees that the algorithm has reached a list of vectors which cannot be further modified by the algorithm.

A second question that arises is up to what level of sieving we should get in order to either get exactly a set $\mathrm{P}(\mathscr{L})$ or a "very good" approximation of it. In this case the Quadruple

MinkowskiSieve was enough, but this might not be the case for higher dimensional lattices. Thus it would be interesting to know how does this index increase according to the dimension. So, given some termination condition, how close can a sieving algorithm approximate a set $\mathrm{P}(\mathscr{L})$?

If these questions receive an answer it will help in making sieving algorithms a way to either compute exactly or approximately a set $\mathrm{P}(\mathscr{L})$ of a lattice $\mathscr{L}$. This would be very interesting as it will provide a way to compute a set $\mathrm{P}(\mathscr{L})$ (exactly or approximately) without having to compute the set $\mathrm{R}(\mathscr{L})$ which is a very costly computation.

6 Applications of $\mathrm{P}(\mathscr{L})$

Even though the sets $\mathrm{Irr}(\mathscr{L})$ and $\mathrm{P}(\mathscr{L})$ might be of interest in their own, examining their relation to already existing lattice problems and algorithms is a natural question that arises. We choose to focus on the set $\mathrm{P}(\mathscr{L})$ as it seems to be the easier to compute/approximate with existing lattice algorithms.

6.1 $\mathrm{P}(\mathscr{L})$ in the study of shortest vector(s) problems

The results in Sect. 4 provide some interesting conclusions about the relation of the set $\mathrm{P}(\mathscr{L})$ to well known lattice problems. A first observation in Sect. 4.2 was that $S_1(\mathscr{L})$ is included in $\mathrm{P}(\mathscr{L})$. This leads to the following result.

Proposition 10 (Finding $\mathrm{P}(\mathscr{L})$ implies solving SVP) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. Computing a set $\mathrm{P}(\mathscr{L})$ provides a solution to the SVP and the kissing number problem.*

The relation $S_1(\mathscr{L}) \subseteq \mathrm{P}(\mathscr{L})$, implies that two classic lattice problems can be solved given a $\mathrm{P}(\mathscr{L})$. Of course this holds for any superset of $\mathrm{P}(\mathscr{L})$ as well. We combine this observation with the inclusion $\mathrm{P}^+(\mathscr{L}) \subseteq \mathrm{MinkowskiSieve}_k(\mathscr{L})$ for $k \geq 2$ shown in the proof of Theorem 9. This provides some extra (heuristic) evidence that some sieving algorithms will indeed output a solution to SVP or the kissing number problem if they run long enough. This is no surprise as sieving algorithms were devised for solving SVP.

Examining the relation of SIVP to the set $\mathrm{P}(\mathscr{L})$ is probably a more interesting question. By Corollary 1 we know that for every $i = 1, \ldots, n$ there exists a vector $\vec{v} \in \mathrm{Irr}(\mathscr{L})$ such that $\|\vec{v}\| = \lambda_i(\mathscr{L})$. The following proposition completes this result.

Proposition 11 (Finding $\mathrm{P}(\mathscr{L})$ implies solving SIVP) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$. Computing a set $\mathrm{P}(\mathscr{L})$ provides a solution to the SIVP.*

Proof Let $\vec{v}_1, \ldots, \vec{v}_n$ be a set of linearly independent vectors in $\mathscr{L}$ such that $\|\vec{v}_i\| = \lambda_i(\mathscr{L})$ for $i = 1, \ldots, n$. We distinguish two cases.
Case 1 $\nexists i \geq 2$ such that $\lambda_1(\mathscr{L}) \leq \lambda_{i-1}(\mathscr{L}) < \lambda_i(\mathscr{L}) = \lambda_{i+1}(\mathscr{L})$. This implies that there exists a $k \geq 1$ such that

$$\lambda_1(\mathscr{L}) = \cdots = \lambda_k(\mathscr{L}) < \lambda_{k+1}(\mathscr{L}) < \cdots < \lambda_n(\mathscr{L}).$$

Then by $S_1(\mathscr{L}) \subseteq \mathrm{P}(\mathscr{L})$ it follows that $\vec{v}_1, \ldots, \vec{v}_k$ belong to $\mathrm{P}(\mathscr{L})$. In addition, by Corollary 1 and the definition of $\mathrm{P}(\mathscr{L})$ it follows that all the $\vec{v}_{k+1}, \ldots, \vec{v}_n$ will be included in $\mathrm{P}(\mathscr{L})$.
Case 2 $\exists i \geq 2$ such that $\lambda_1(\mathscr{L}) \leq \lambda_{i-1}(\mathscr{L}) < \lambda_i(\mathscr{L}) = \lambda_{i+1}(\mathscr{L})$. Let $i \geq 2$ such that the condition holds. We set $k = \max\{j > i \,|\lambda_i(\mathscr{L}) = \lambda_j(\mathscr{L})\}$. Hence,

$$\lambda_1(\mathscr{L}) \leq \lambda_{i-1}(\mathscr{L}) < \lambda_i(\mathscr{L}) = \lambda_{i+1}(\mathscr{L}) = \cdots = \lambda_k(\mathscr{L}).$$

We will show that $\vec{v}_i, \ldots, \vec{v}_k \in \mathrm{P}(\mathscr{L})$. Let $j \in \{i, \ldots, k\}$ we set $\mathscr{L}_{\lambda_j}$ to be the sublattice of $\mathscr{L}$ spanned by all the vectors in $\mathscr{L}$ strictly shorter than λ_j. As $\lambda_i(\mathscr{L}) = \lambda_j(\mathscr{L})$ it follows that $\mathscr{L}_{\lambda_j} = \mathscr{L}_{\lambda_i}$ which has rank $i-1$. Assume that $\vec{v}_j \in \mathscr{L}_{\lambda_j}$. Then we would get that the set $\{\vec{v}_1, \ldots, \vec{v}_{i-1}, \vec{v}_j\}$ is a set of linearly dependent vectors. Contradiction. Thus $\vec{v}_j \notin \mathscr{L}_{\lambda_j}$ and by Proposition 4 we get that $\vec{v}_j \in \mathrm{Irr}(\mathscr{L})$. This holds for any $i \leq j \leq k$ and therefore we get that all $\vec{v}_j$ with $i \leq j \leq k$ belong to $\mathrm{Irr}(\mathscr{L})$. In order to show that they also do belong to a $\mathrm{P}(\mathscr{L})$ it suffices to show that for every μ, ν such that $i \leq \mu < \nu \leq k$ it holds that $\|\vec{v}_\nu - \vec{v}_\mu\| \geq \lambda_i(\mathscr{L})$. Assume that there exist μ, ν such that $i \leq \mu < \nu \leq k$ and $\|\vec{v}_\nu - \vec{v}_\mu\| < \lambda_i(\mathscr{L})$. Then it follows that $\vec{v}_\nu - \vec{v}_\mu \in \mathscr{L}_{\lambda_i}$. The set of vectors $\{\vec{v}_1, \ldots, \vec{v}_{i-1}, \vec{v}_\mu, \vec{v}_\nu\}$ is a linearly independent set and thus the same holds for $\{\vec{v}_1, \ldots, \vec{v}_{i-1}, \vec{v}_\nu - \vec{v}_\mu\}$. This implies a set of i linearly independent vectors in the lattice $\mathscr{L}_{\lambda_i}$ which is of rank $i-1$, contradiction.
Concluding, let $\vec{v}_l$ belong to the considered linearly independent set of vectors achieving the successive minima. If $\|\vec{v}_l\| = \|\vec{v}_{l+1}\|$ or $\|\vec{v}_l\| = \|\vec{v}_{l-1}\|$ then $\vec{v}_l \in \mathrm{P}(\mathscr{L})$ by the proof in "case 2". If $\|\vec{v}_{l-1}\| < \|\vec{v}_l\| < \|\vec{v}_{l+1}\|$ then $\vec{v}_l \in \mathrm{P}(\mathscr{L})$ by the same argument used in "case 1". □

Remark 21 Obtaining a set of the shortest vector(s), given a set $\mathrm{P}(\mathscr{L})$, amounts to scanning the entire set $\mathrm{P}(\mathscr{L})$ a number of times. Thus, sorting $\mathrm{P}(\mathscr{L})$ can be avoided.

6.2 Using $\mathrm{P}(\mathscr{L})$ in CVPP algorithms

One main problem in lattice theory is the closest vector problem. A straightforward way of using the set $\mathrm{R}(\mathscr{L})$ in order to solve CVPP was described in [31]. In that work, an algorithm called the iterative slicer is given which takes as input the set $\mathrm{R}(\mathscr{L})$ and a target vector $\vec{t}$ and outputs a closest lattice vector to $\vec{t}$ (Algorithm 4). The main idea behind this algorithm is to iteratively reduce the target vector $\vec{t}$ by the relevant vectors until the resulting vector $\vec{t}'$ is contained in the Voronoi cell $\mathscr{V}(\mathscr{L})$ of the lattice. Once this condition is satisfied it is known that $\vec{t} - \vec{t}'$ is a closest lattice point to $\vec{t}$. This algorithm is shown to terminate after a finite number of iterations.

Algorithm 4 The iterative slicer [31]

Require: The set $\mathrm{R}(\mathscr{L})$ and a target vector $\vec{t}$.
Ensure: A vector $\vec{s} \in \mathscr{L}$ closest to $\vec{t}$.
1: $\vec{t}' \leftarrow \vec{t}$
2: **for** every $\vec{r} \in \mathrm{R}(\mathscr{L})$ **do**
3: **if** $\|\vec{t}' \pm \vec{r}\| < \|\vec{t}'\|$ **then**
4: $\vec{t}' \leftarrow \vec{t}' \pm \vec{r}$
5: restart the **for** loop
6: **end if**
7: **end for**
8: $\vec{s} = \vec{t} - \vec{t}'$
9: **return** $\vec{s}$

Inspired by the iterative slicer, in [23] an algorithm is described to provably solve the CVPP in $\tilde{O}(2^{2n})$-time by using the set $\mathrm{R}(\mathscr{L})$ as the preprocessing data. The difference between Algorithm 4 and the algorithm in [23] is that the latter selects the relevant vectors in a specific order for reduction. This results in a $\tilde{O}(2^{2n})$-time and $\tilde{O}(2^n)$-space algorithm. This work was further improved in [4] by optimising the use of the preprocessing data.

However, using the set $\mathrm{R}(\mathscr{L})$ in practice is not convenient due to its expected size of about $2^{n+1} - 2$ vectors. One way to reduce the memory requirements could be the use of a compact

Algorithm 5 The tuple slicer

Require: A set $\mathrm{P}(\mathscr{L})$, a $C \in \mathbb{N}$ and a target vector $\vec{t}$.
Ensure: A vector $\vec{s} \in \mathscr{L}$ closest to $\vec{t}$.
1: $\vec{t}' \leftarrow \vec{t}$
2: **for** $l = 1$ **to** C **do**
3: **for** all $\{\vec{v}_1, \ldots, \vec{v}_l\} \subset \mathrm{P}(\mathscr{L})$ **do**
4: $\vec{w} \leftarrow \sum_{i=1}^{l} \vec{v}_i$
5: **if** $\|\vec{t}' - \vec{w}\| < \|\vec{t}'\|$ **then**
6: $\vec{t}' \leftarrow \vec{t}' - \vec{w}$
7: restart the outer **for** loop
8: **end if**
9: **end for**
10: **end for**
11: $\vec{s} = \vec{t} - \vec{t}'$
12: **return** $\vec{s}$

representation of $\mathrm{R}(\mathscr{L})$ like the one described in [17]. In such a scenario a superset of $\mathrm{R}(\mathscr{L})$ would be generated on the fly by a CVPP algorithm which would only use a smaller set of vectors in order to generate $\mathrm{R}(\mathscr{L})$.

Another way would be to use a subset of $\mathrm{R}(\mathscr{L})$ instead of the entire set. Such an approach was introduced in [19]. In that work an approximate Voronoi cell is defined as the cell implied by a list of short lattice vectors which is potentially a subset of the set $\mathrm{R}(\mathscr{L})$. That lead to a heuristic algorithm for CVPP using the approach of Micciancio–Voulgaris but with more practical time and space complexities.

We describe a CVPP algorithm (the tuple slicer, Algorithm 5) using the set $\mathrm{P}(\mathscr{L})$, and we discuss its advantages and disadvantages against already existing approaches. We distinguish two cases.

If $C = 1$ in Algorithm 5 then it just uses a subset of $\mathrm{R}(\mathscr{L})$. In this case the analysis of the algorithm just follows under the "approximate Voronoi cell" approach where a specific choice has been made on the used subset. The advantage in this case is that it is guaranteed that the used list of vectors is a subset of $\mathrm{R}(\mathscr{L})$.

If $C > 1$ Algorithm 5 behaves similar to the tuple sieving approach in [3]. A vector is reduced not only by a single vector but also by the sums of small tuples of vectors in the used list. Hence, a target vector $\vec{t}$ is reduced by a superset of $\mathrm{P}(\mathscr{L})$. If this superset includes the set $\mathrm{R}(\mathscr{L})$ then [31, Lemma 5] guarantees the correctness of the algorithm. This depends on the value of C. We can prove that there always exists a value of C which guarantees the inclusion of $\mathrm{R}(\mathscr{L})$ in the generated superset.

Remark 22 In line 3 of Algorithm 5 it considers sets of vectors $\{\vec{v}_1, \ldots, \vec{v}_l\}$ such that $\vec{v}_i \neq -\vec{v}_j$ but it could be that $\vec{v}_i = \vec{v}_j$.

Definition 13 (k-wise sum of $\mathrm{P}(\mathscr{L})$) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$ and k a positive integer. We define

$$k\,\mathrm{P}(\mathscr{L}) = \left\{ \sum_{i=1}^{j} \vec{p}_i \mid \vec{p}_i \in \mathrm{P}(\mathscr{L}) \text{ and } j = 1, \ldots, k \right\}.$$

Proposition 12 (Finding $\mathrm{R}(\mathscr{L})$ via $k\,\mathrm{P}(\mathscr{L})$) *Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$ and $\mathrm{P}(\mathscr{L})$ a complete system of irreducible vectors of it. Then there exists a positive integer $n_0 \in \mathbb{N}$ such that $\mathrm{R}(\mathscr{L}) \subseteq n_0\,\mathrm{P}(\mathscr{L})$.*

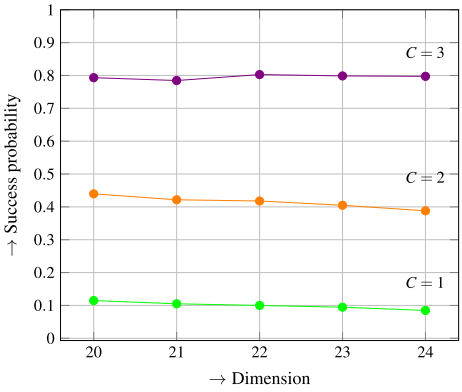


Fig. 4 Preliminary experimental results on the success probability of Algorithm 5. The algorithm was tested on lattices of dimensions 20, 21, 22, 23, 24. For each dimension the algorithm was tested with input $C = 1, 2, 3$ against 10,000 CVP instances. Each of the 10,000 CVP blocks was formed by 10 smaller blocks of 1000 CVPs corresponding to 10 lattices. Each point in the graph corresponds to the ratio of correct answers out of the 10,000 CVP instances

Proof By Proposition 7 there exists a generating set $\vec{\mathrm{G}} \subseteq \mathrm{P}(\mathscr{L})$ with $|\vec{\mathrm{G}}| = l \geq n$. Let $\vec{r} \in \mathrm{R}(\mathscr{L})$, then there exists an $\vec{x} \in \mathbb{Z}^l$ such that $\vec{\mathrm{G}}\vec{x} = \vec{r}$. With $\vec{x} = (x_1, \ldots, x_l)$ set $m_{\vec{r}} = \|\vec{x}\|_1 = \sum_{i=1}^{l} |x_i|$. Then $\vec{r} \in m_{\vec{r}}\,\mathrm{P}(\mathscr{L})$. Set $m = \max_{\vec{r} \in \mathrm{R}(\mathscr{L})}\{m_{\vec{r}}\}$. As $\mathrm{R}(\mathscr{L})$ is finite then m is finite and $\forall \vec{r} \in \mathrm{R}(\mathscr{L})$ it holds that $\vec{r} \in m\,\mathrm{P}(\mathscr{L})$. □

The used superset is computed on the fly. This allows for a time–memory trade-off. The algorithm loses on time complexity as it examines a larger list of vectors but it gains on the memory requirement as it stores a provably smaller subset of $\mathrm{R}(\mathscr{L})$. In more detail the space complexity of the algorithm is proportional to $|\,\mathrm{P}(\mathscr{L})|$ which can be bounded by $O(\tau_n)$. The time complexity will depend on the size of $\mathrm{P}(\mathscr{L})$ but also on the parameter C. Following the analysis of [23] we can argue that the time complexity of Algorithm 5 will be $O(|\,\mathrm{P}(\mathscr{L})|^C \cdot 2^n\,\mathrm{poly}(n))$.

Remark 23 From Theorem 12 it follows that if Algorithm 5 was to be applied to the lattice family A_n^*, it should consider a value of C as high as $(n+1)/2$ in order for $\mathrm{R}(A_n^*)$ to be included in the used superset. Therefore, a provable upper bound on C alone will not lead to any good bound for the time complexity of Algorithm 5 in a provable setting.

Considering Algorithm 5 in a heuristic setting seems to be a more appealing choice. In such a scenario the requirements of the algorithm can be relaxed in mainly two directions. The first one is using an approximation (a superset) of $\mathrm{P}(\mathscr{L})$ instead of the set itself. Hence, the output of the MinkowskiSieve as described in Sect. 5.2 could serve as such a choice. Furthermore, choosing a specific approximation of $\mathrm{P}(\mathscr{L})$ can allow fixing the value of the parameter C in the following way.

By a heuristic result of [19] we know that if a list L containing $2^{n/2+o(n)}$ lattice vectors of norm less than $\sqrt{2}\lambda_1(\mathscr{L})$ is used as input to the iterative slicer then the success probability of the algorithm is close to 1. Following this guideline, a value for the parameter C can be chosen in a way that guarantees that the set of all vectors used for reduction in Algorithm 5 contains a list of $2^{n/2+o(n)}$ shortest lattice vectors.

Further options can be examined if it is allowed for the used slicing algorithm to succeed with probability much smaller than 1. In such a case the results in [10, 11] provide a way of relating the success probability to size of the used preprocessed list and hence in our case C.

We briefly experimented on the relation of the success probability of Algorithm 5 and the parameter C. The results can be found in Fig. 4. From these results we get a first indication that the success probability of Algorithm 5 increases as the value of C increases. Unfortunately, extending these experiments to moderate dimensions was infeasible, as the exact

computation of $\mathrm{P}(\mathscr{L})$ would require hundreds or thousands of GB of RAM (using a "brute force" approach). Therefore, obtaining a specific guideline on how to choose a value for C remains an open question.

Acknowledgements We thank Filipe Cabeleira, Artur Mariano and Gabriel Falcao for providing their parallel implementation computing $\mathrm{R}(\mathscr{L})$ as described in [5], which we used for our experimental results in Sect. 5.3. The authors thank Daniel Dadush for helpful discussions regarding how efficiently can the set $\mathrm{R}(\mathscr{L})$ be generated by $\mathrm{P}(\mathscr{L})$. We also thank Noah Stephens-Davidowitz for spotting a mistake in an earlier version of the paper. Finally, we thank the anonymous reviewers for their suggestions that led to improvements in the paper.

Appendix A Corner cases among $S_1(\mathscr{L})$, $\mathrm{Irr}(\mathscr{L})$, and $R(\mathscr{L})$

In Sect. 4.1 we posed two questions regarding the set $\mathrm{Irr}(\mathscr{L})$: if and when the corner cases $S_1(\mathscr{L}) \subsetneq \mathrm{Irr}(\mathscr{L}) = \mathrm{R}(\mathscr{L})$ and $S_1(\mathscr{L}) = \mathrm{Irr}(\mathscr{L}) \subsetneq \mathrm{R}(\mathscr{L})$ are possible. In this section we will give a partial answer to these questions by examining some already known families of special lattices, the duals of the root lattices D_n and A_n (see [8, Chapter 4]). For $n \in \mathbb{N}$ with $n \geq 5$ we write [4]:

$$\mathscr{L}_n = 2D_n^*. \tag{19}$$

Then a basis of $\mathscr{L}_n$ is the following (see [8, p. 120]):

$$\vec{\mathrm{B}}_n = \{2e_i \mid 1 \leq i \leq n-1\} \cup \{1^n\}, \tag{20}$$

where 1^n represents the all-1 vector.

Theorem 10 (Properties of $\mathscr{L}_n$) *For every $n \in \mathbb{N}$ with $n \geq 5$*

$$S_1(\mathscr{L}_n) = \{\pm 2e_i \mid 1 \leq i \leq n\} \quad \textit{and}$$
$$\mathrm{Irr}(\mathscr{L}_n) = \mathrm{R}(\mathscr{L}_n) = \{\pm 2e_i \mid 1 \leq i \leq n\} \cup \{\pm 1\}^n.$$

Proof By the definition of the lattice $\mathscr{L}_n$ it is clear that $\{\pm 2e_i \mid 1 \leq i \leq n\} \cup \{\pm 1\}^n \subset \mathscr{L}_n$. We will prove this theorem in three steps.
The first step is to show that $S_1(\mathscr{L}_n) = \{\pm 2e_i \mid 1 \leq i \leq n\}$.
The second step will be to show that $\mathrm{R}(\mathscr{L}_n) \subseteq \{\pm 2e_i \mid 1 \leq i \leq n\} \cup \{\pm 1\}^n$.
Finally in the third step we will prove that $\{\pm 2e_i \mid 1 \leq i \leq n\} \cup \{\pm 1\}^n \subseteq \mathrm{Irr}(\mathscr{L}_n)$. These three steps imply the result as $\mathrm{Irr}(\mathscr{L}_n) \subseteq \mathrm{R}(\mathscr{L}_n)$.

[4] We choose to work with a scaling of D_n^* as in this way we get a lattice in $\mathbb{Z}^n$, which is easier to work with.

The "defining property" of the lattice $\mathscr{L}_n$, that if $\vec{v} = (v_1, \ldots, v_n) \in \mathscr{L}_n$ then $v_i \equiv v_j \pmod 2$ for all $1 \leq i, j \leq n$, will be used throughout the proof.

Step 1 Obtaining that $S_1(\mathscr{L}_n) = \{\pm 2e_i \mid 1 \leq i \leq n\}$ is trivial and is left as an exercise to the reader.

Step 2 Let $\vec{v} \notin \mathrm{R}(\mathscr{L}_n)$ and $\vec{v} \neq \vec{0}$. Then by Theorem 1 we know that there exists a vector $\vec{x} \in \mathscr{L}_n \setminus \{\vec{0}, \vec{v}\}$ such that $\langle \vec{v}, \vec{x} \rangle \geq \|x\|^2$. We will prove that for every vector $\vec{v} \in \mathscr{L}_n \setminus (\{\pm 2e_i \mid 1 \leq i \leq n\} \cup \{\pm 1\}^n \cup \{\vec{0}\})$ there exists a vector $\vec{x} \in \mathscr{L}_n \setminus \{\vec{0}, \vec{v}\}$ such that $\langle \vec{v}, \vec{x} \rangle \geq \|x\|^2$. This implies the desired property $\mathrm{R}(\mathscr{L}_n) \subseteq \{\pm 2e_i \mid 1 \leq i \leq n\} \cup \{\pm 1\}^n$.
Let $\vec{v} \in \mathscr{L}_n \setminus (\{\pm 2e_i \mid 1 \leq i \leq n\} \cup \{\pm 1\}^n \cup \{\vec{0}\})$, we distinguish two cases.
Case 1: Let $\vec{v}$ be such that $\vec{v} = (v_1, \ldots, v_n)$ with $v_i \equiv 1 \pmod 2$ for all v_i. We already showed in step 1 of the proof that the shortest vectors with odd coordinates are the $\{\pm 1\}^n$. As $\vec{v}$ does not belong to this set, $|v_i| \geq 1$ for all v_i, and there exists at least one v_j such that $|v_j| \geq 3$. Consider the vector $\vec{x} = (\mathrm{sign}(v_1), \ldots, \mathrm{sign}(v_n))$. This is a valid lattice vector as $\vec{x} \in \{\pm 1\}^n \subset \mathscr{L}_n$ and $\vec{x} \neq \vec{v}$ as $\vec{v} \in \mathscr{L}_n \setminus (\{\pm 2e_i \mid 1 \leq i \leq n\} \cup \{\pm 1\}^n)$. We check the inner product of $\vec{v}$ and $\vec{x}$.

$$\langle \vec{v}, \vec{x} \rangle = \sum_{i=1}^{n} \mathrm{sign}(v_i) v_i = \sum_{i=1}^{n} |v_i| \geq n + 2 > n = \|\vec{x}\|^2$$

This proves that $\vec{v} \notin \mathrm{R}(\mathscr{L}_n)$.
Case 2: Let $\vec{v}$ be such that $\vec{v} = (v_1, \ldots, v_n)$ with $v_i \equiv 0 \pmod 2$ for all v_i. As $\vec{v}$ is a non-zero vector then it has at least one non-zero coordinate, let it be v_j. Also as v_j is even we can conclude that $|v_j| \geq 2$. We consider the vector $\vec{x} = 2\,\mathrm{sign}(v_j) e_j$. This is a valid lattice vector as $\vec{x} \in \{\pm 2e_i \mid 1 \leq i \leq n\} \subset \mathscr{L}_n$ and $\vec{x} \neq \vec{v}$ as $\vec{v} \in \mathscr{L}_n \setminus (\{\pm 2e_i \mid 1 \leq i \leq n\} \cup \{\pm 1\}^n)$. We check the inner product of $\vec{v}$ and $\vec{x}$.

$$\langle \vec{v}, \vec{x} \rangle = \sum_{i=1}^{n} x_i v_i = 2\,\mathrm{sign}(v_j) v_j = 2|v_j| \geq 4 = \|\vec{x}\|^2$$

This proves that again $\vec{v} \notin \mathrm{R}(\mathscr{L}_n)$ concluding the proof of the second step.

Step 3 In this step we want to prove that $\{\pm 2e_i \mid 1 \leq i \leq n\} \cup \{\pm 1\}^n \subseteq \mathrm{Irr}(\mathscr{L}_n)$. In step 1 we already showed that $S_1(\mathscr{L}_n) = \{\pm 2e_i \mid 1 \leq i \leq n\}$ and we know that $S_1(\mathscr{L}_n) \subseteq \mathrm{Irr}(\mathscr{L}_n)$ hence, we only have to show that $\{\pm 1\}^n \subseteq \mathrm{Irr}(\mathscr{L}_n)$. Assume that $\vec{v} \in \{\pm 1\}^n$ and $\vec{v} \notin \mathrm{Irr}(\mathscr{L}_n)$. Thus there are two strictly shorter vectors $\vec{v}_1$ and $\vec{v}_2$ such that $\vec{v} = \vec{v}_1 + \vec{v}_2$. In step 1 of the proof we showed that the vectors in $\{\pm 1\}^n$ are the shortest ones among those with odd coordinates. Therefore as $\vec{v}_1$ and $\vec{v}_2$ are strictly shorter than $\vec{v}$ then it must be that they have even coordinates. This implies that $\vec{v}$ can be written as a sum of vectors with even coordinates. This is a contradiction, as a sum of even numbers is never odd. □

As a scaling of a lattice $\mathscr{L}$ has the same properties as $\mathscr{L}$ we get Theorem 5 already mentioned in Sect. 4.1.

Theorem 11 (Properties of D_n^*) *Let $n \in \mathbb{N}$ with $n \geq 5$. Then for the lattice D_n^* it holds that $S_1(D_n^*) \subsetneq \mathrm{Irr}(D_n^*) = \mathrm{R}(D_n^*)$. Furthermore $|\mathrm{Irr}(D_n^*)| = 2^n + 2n$.*

This proves that $S_1(\mathscr{L}) \subsetneq \mathrm{Irr}(\mathscr{L}) = \mathrm{R}(\mathscr{L})$ is possible for every dimension $n \geq 5$. In order to complete this result from this point of view we give another three lattices, one for each of the dimensions $n = 2, 3, 4$ that possess the same property.

For $n = 2, 3, 4$ we write $\mathscr{L}_2 = \mathscr{L}(\vec{\mathrm{B}}_2), \mathscr{L}_3 = \mathscr{L}(\vec{\mathrm{B}}_3), \mathscr{L}_4 = \mathscr{L}(\vec{\mathrm{B}}_4)$ with $\vec{\mathrm{B}}_2, \vec{\mathrm{B}}_3, \vec{\mathrm{B}}_4$ being

$$\vec{\mathrm{B}}_2 = \begin{pmatrix} 3 & 1 \\ 0 & 1 \end{pmatrix} \quad \vec{\mathrm{B}}_3 = \begin{pmatrix} 3 & 0 & 1 \\ 0 & 3 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \vec{\mathrm{B}}_4 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 3 & 0 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{21}$$

We leave it to the reader to verify our claim for these three lattices.
Our next goal is to derive a similar result for the case $S_1(\mathscr{L}) = \mathrm{Irr}(\mathscr{L}) \subsetneqq \mathrm{R}(\mathscr{L})$. In order to do so we will use a scaling of the lattices A_n^*.

For $n \in \mathbb{N}$ with $n \geq 3$, we write

$$\mathscr{M}_n = (n+1)A_n^*. \tag{22}$$

Then a basis of $\mathscr{M}_n$ is formed by the columns of $\vec{\mathrm{B}}_n$ (see [8, p. 115]), where

$$\vec{\mathrm{B}}_n = \begin{pmatrix} -n & 1 & 1 & \cdot & 1 \\ 1 & -n & 1 & \cdot & 1 \\ 1 & 1 & -n & \cdot & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \cdot & -n \\ 1 & 1 & 1 & \cdot & 1 \end{pmatrix} \tag{23}$$

is an $(n+1) \times n$ matrix.

Remark 24 By the given basis $\vec{\mathrm{B}}_n$ for $\mathscr{M}_n$ we can immediately observe that if $\vec{v} = (v_1, v_2, \ldots, v_{n+1}) \in \mathscr{M}_n$ then $v_i \equiv v_j \pmod{n+1}$. Additionally $\sum_{i=1}^{n+1} v_i = 0$.

Theorem 12 (Properties of $\mathscr{M}_n$) *For every $n \in \mathbb{N}$ with $n \geq 3$,*

$$S_1(\mathscr{M}_n) = \mathrm{Irr}(\mathscr{M}_n) = \{\pm(-n^1, 1^n)\} \quad \textit{and}$$
$$\mathrm{R}(\mathscr{M}_n) = \left\{ \pm(\alpha^\beta, (-\beta)^\alpha) \,|\, \beta = n+1-\alpha\,,\ 1 \leq \alpha \leq \frac{n+1}{2} \right\}.$$

Proof We set $A = \{\pm(\alpha^\beta, (-\beta)^\alpha) \,|\, \beta = n+1-\alpha\,,\ 1 \leq \alpha \leq (n+1)/2\}$. We will prove this theorem in four steps. The first step is to show that $S_1(\mathscr{M}_n) = \{\pm(-n^1, 1^n)\}$. The second step will be to show that $\mathrm{R}(\mathscr{M}_n) \subseteq A$. The third step will be to show that $\mathrm{Irr}(\mathscr{M}_n) \subseteq \{\pm(-n^1, 1^n)\}$ and finally in the fourth step we will show that $A \subseteq \mathrm{R}(\mathscr{M}_n)$.

Step 1 The vectors $\{\pm(-n^1, 1^n)\}$ have squared length $n^2 + n$ and hence we get $\lambda_1^2(\mathscr{L}) \leq n^2 + n$. This implies that a vector achieving $\lambda_1(\mathscr{L})$ cannot have a coordinate v_j such that $|v_j| \geq n+1$. Therefore a vector achieving $\lambda_1(\mathscr{L})$ belongs to A. The squared length of a vector in A is $\beta\alpha^2 + \alpha\beta^2 = (n+1)\alpha\beta$ which minimizes for $\alpha = 1$.

Step 2 Let $\vec{v} \in \mathscr{M}_n \setminus (A \cup \{\vec{0}\})$ and write it as $\vec{v} = (v_1, \ldots, v_{n+1})$. Then there will exist at least one coordinate of $\vec{v}$, let it be v_j, such that $|v_j| \geq n+1$. This can be proved by a contradiction argument. Assume that there was no coordinate in $\vec{v}$ such that $|v_j| \geq n+1$ then it would hold that $|v_i| \leq n$ for all $1 \leq i \leq n$ and by the fact that $v_i \equiv v_j \pmod{n+1}$ we can conclude that there would be at most two possible values for $|v_i|$. But the set A contains all such vectors of the lattice, hence that would imply $\vec{v} \in A$, contradiction. We set $\vec{x}$ to be the vector having $\mathrm{sign}(v_j)n$ in the j-th position and $-\mathrm{sign}(v_j)$ in all other places. This is a

valid lattice vector and $\vec{x} \neq \vec{v}$ as $\vec{v} \in \mathscr{M}_n \setminus A$. We check the inner product of $\vec{v}$ and $\vec{x}$:

$$\langle \vec{v}, \vec{x} \rangle = \sum_{i=1}^{n+1} v_i x_i = |v_j| n - \operatorname{sign}(v_j) \sum_{\substack{i=1 \\ i \neq j}}^{n+1} v_i = |v_j| n + |v_j| \geq (n+1)^2 > \|\vec{x}\|^2.$$

This proves that $\vec{v} \notin \mathrm{R}(\mathscr{M}_n)$ concluding the proof of the second step.

Step 3 Let $\vec{v} \in \mathscr{M}_n \setminus (\{\pm(-n^1, 1^n)\} \cup \{\vec{0}\})$ and write it as $\vec{v} = (v_1, \ldots, v_{n+1})$. Then we will show that $\vec{v}$ is reducible. By step 2 of the proof we know that $\mathrm{R}(\mathscr{M}_n) \subseteq A$ and as we know that $\mathrm{Irr}(\mathscr{M}_n) \subseteq \mathrm{R}(\mathscr{M}_n)$ we can restrict our choice to $\vec{v} \in A \setminus \{\pm(-n^1, 1^n)\}$. As $\vec{v} \in A$ we can write $\vec{v} = \pm(\alpha^\beta, (-\beta)^\alpha)$ with $\beta = n + 1 - \alpha$ for some $1 < \alpha \leq (n+1)/2$. By Lemma 4 it suffices to find a lattice vector $\vec{x}$ with $\|\vec{x}\| < \|\vec{v}\|$ and such that $2\langle \vec{v}, \vec{x} \rangle > \|\vec{x}\|^2$. Let $\gamma = \max\{|\alpha|, |\beta|\}$ and the j-th coordinate of $\vec{v}$ be such that $|v_j| = \gamma$. Consider $\vec{x}$ to be the vector with $\operatorname{sign}(v_j)n$ in the j-th position and $-\operatorname{sign}(v_j)$ in all other places. This is a valid lattice vector and $\|\vec{x}\| < \|\vec{v}\|$ as $\vec{x} \in S_1(\mathscr{M}_n)$ but $\vec{v} \notin S_1(\mathscr{M}_n)$. Then

$$\begin{aligned} 2\langle \vec{v}, \vec{x} \rangle &= 2 \sum_{i=1}^{n+1} v_i x_i = 2\left(|v_j| n - \operatorname{sign}(v_j) \sum_{\substack{i=1 \\ i \neq j}}^{n+1} v_i \right) = 2(\gamma n + \gamma) \\ &= 2(n+1)\gamma \geq (n+1)^2 > \|\vec{x}\|^2 \end{aligned}$$

as $\gamma \geq (n+1)/2$. This proves that $\vec{v} \notin \mathrm{Irr}(\mathscr{M}_n)$ and therefore $\mathrm{Irr}(\mathscr{M}_n) \subseteq \{\pm(-n^1, 1^n)\}$.

Step 4 By [7, Theorem 3] and the fact that the vectors $(-n^1, 1^n)$ form a strictly obtuse superbasis of $\mathscr{M}_n$ (see [7]) it follows that $A \subseteq \mathrm{R}(\mathscr{M}_n)$ and finally $\mathrm{R}(\mathscr{M}_n) = A$. □

This implies Theorem 6 already mentioned in Sect. 4.1.

Theorem 13 (Properties of A_n^*) *Let $n \in \mathbb{N}$ with $n \geq 3$. Then for the lattice A_n^* it holds that $S_1(A_n^*) = \mathrm{Irr}(A_n^*) \subsetneq \mathrm{R}(A_n^*)$. Furthermore $|\mathrm{Irr}(A_n^*)| = 2(n+1)$.*

This proves that $S_1(\mathscr{L}) = \mathrm{Irr}(\mathscr{L}) \subsetneq \mathrm{R}(\mathscr{L})$ is possible for every dimension $n \geq 3$. In order to complete the result from this point of view we give another lattice in dimension $n = 2$ that possess the same property: $\mathscr{M}_2 = \mathscr{L}(\vec{\mathrm{B}}_2)$, where

$$\vec{\mathrm{B}}_2 = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}. \tag{24}$$

We leave it to the reader to verify this.

Appendix B Some graph-theoretical aspects

In Sect. 4.2 we introduced the notion of a complete system of irreducible vectors and we gave an example of how the set $\mathrm{P}(\mathscr{L})$ can be computed. In that example the use of graph theoretical tools was demonstrated in order to compute the set $\mathrm{P}(\mathscr{L})$ given the set $\mathrm{Irr}(\mathscr{L})$. A natural question that arises is how costly this step can be.

In order to answer this question a few graph theory definitions are necessary. Graphs will de denoted by $\Gamma = (V, E)$, where V is the set of vertices and E is the set of edges. If $e = \{u, v\} \in E$ then we say that u and v are adjacent.

Definition 14 (Independent set) Given a simple graph $\Gamma = (V, E)$ an independent set is a subset of vertices $U \subseteq V$, such that no two vertices in U are adjacent. An independent set is maximal if no vertex can be added without violating independence. An independent set of maximum cardinality is called maximum and its cardinality is denoted by $\alpha(\Gamma)$.

Definition 15 (Class graph) Let $\mathscr{L}$ be a full rank lattice in $\mathbb{R}^n$ and $S \in \mathrm{Irr}(\mathscr{L})/\sim$. We define $\Gamma_{\mathscr{L}}(S)$ to be the graph where the set of vertices $V = S$ and there exists an edge between $\vec{v}_1, \vec{v}_2 \in V$ iff $\|\vec{v}_1 - \vec{v}_2\| < \|\vec{v}_1\|$.

Computing $\mathrm{P}(\mathscr{L})$ out of $\mathrm{Irr}(\mathscr{L})$ amounts to solving a maximal independence set instance in $\Gamma_{\mathscr{L}}(S)$ for every class $S \in \mathrm{Irr}(\mathscr{L})/\sim$. Therefore the complexity of this task highly depends on the size of the equivalence classes $S \in \mathrm{Irr}(\mathscr{L})/\sim$ and $|\mathrm{Irr}(\mathscr{L})/\sim|$. For average-case lattices the computational step from $\mathrm{Irr}(\mathscr{L})$ to $\mathrm{P}(\mathscr{L})$ should almost always be trivial, i.e. $\mathrm{P}(\mathscr{L}) = \mathrm{Irr}(\mathscr{L})$, as for all $S \in \mathrm{Irr}(\mathscr{L})/\sim$ it is expected that $|S| = 2$. In these cases the set $\mathrm{P}(\mathscr{L})$ is uniquely determined.

However, in case the underlying lattice $\mathscr{L}$ possesses any kind of structure or symmetries it is expected that there will be equivalence classes $S \in \mathrm{Irr}(\mathscr{L})/\sim$ with $|S| > 2$. In these cases the computational task of finding a maximal independent set in the corresponding class graph is not trivial anymore. In such cases the first step is to construct the corresponding graph $\Gamma_{\mathscr{L}}(S)$, which will take time $O(|S|^2)$. Then, naively computing a maximal independent set (which should always include both $\pm\vec{v}$) will take time $O(|S|m)$ where m is the number of edges in $\Gamma_{\mathscr{L}}(S)$ but, there are better performing algorithms for this task [21]. If we denote by h the maximum size of a class in $\mathrm{Irr}(\mathscr{L})/\sim$ then the time complexity of computing $\mathrm{P}(\mathscr{L})$ out of $\mathrm{Irr}(\mathscr{L})$ will be bounded by $O(h^2|\mathrm{Irr}(\mathscr{L})|)$.

Thus if there does not exist a class S with $|S|$ exponential to the dimension n then computing $\mathrm{P}(\mathscr{L})$ out of $\mathrm{Irr}(\mathscr{L})$ will take time $\tilde{O}(|\mathrm{Irr}(\mathscr{L})|)$. In practice, stumbling upon a lattice $\mathscr{L}$ possessing a class $S \in \mathrm{Irr}(\mathscr{L})/\sim$ where $|S|$ is exponential to the dimension n is highly unlikely as such a lattice would be extremely structured. For the sake of mathematical curiosity (and nice graph pictures) we briefly investigate such a case of lattices, namely the $\mathscr{L}_n$ for $n \geq 5$ defined in Appendix A. For our exposition we will need the following definition.

Definition 16 (Cayley graph) Let G be a group and $T \subseteq G$ a generating set of G. The Cayley graph of G generated by T, denoted $\mathrm{Cay}(G, T)$ is the directed graph $\Gamma = (V, E)$ where $V = G$ and $E = \{(g, gs) \mid g \in G,\ s \in T\}$.

If $T = T^{-1}$ (T is closed under inverse) then $\mathrm{Cay}(G, T)$ is an undirected graph.

In Appendix A we saw that $\mathrm{Irr}(\mathscr{L}_n) = \{\pm 2\vec{e}_i \mid 1 \leq i \leq n\} \cup \{\pm 1\}^n$. Hence $\mathrm{Irr}(\mathscr{L}_n)$ contains two equivalence classes of sizes $2n$ and 2^n respectively. The class $S_2 := \{\pm 1\}^n$ which we will study can be viewed as the group $G = \mathbb{Z}_2^n$. Two elements of S_2 are connected if their difference is shorter than $\sqrt{n}$, thus it is a sum of less than $n/4$ elements from the set $\{\pm 2\vec{e}_i \mid 1 \leq i \leq n\}$. In turn this implies that two elements of S_2 are connected in $\Gamma_{\mathscr{L}_n}(S_2)$ if they differ by a sign in less than $n/4$ of their coordinates. Thus we can now use the following observation.

$$\Gamma_{\mathscr{L}_n}(S_2) \cong \mathrm{Cay}(G,\ T_{\lceil n/4 \rceil}(G)) \tag{25}$$

Where $G = \mathbb{Z}_2^n$ and $T_{\lceil n/4 \rceil}(G) := \{\vec{x} \in G \mid 1 \leq |\mathrm{supp}(\vec{x})| < \lceil n/4 \rceil\}$ and $\mathrm{supp}(\vec{x})$ denotes the support of $\vec{x}$. In our case $T_{\lceil n/4 \rceil}(G) = T^{-1}_{\lceil n/4 \rceil}(G)$ and thus $\mathrm{Cay}(G, T_{\lceil n/4 \rceil}(G))$ is an undirected graph.[5] We are interested in the maximal independent sets of the graph $\Gamma_{\mathscr{L}_n}(S_2)$, but not in

[5] Such type of Cayley graphs are of an interest in coding theory as independent sets of $\mathrm{Cay}(\mathbb{Z}_q^n, T_d(\mathbb{Z}_q^n))$ with $T_d(\mathbb{Z}_q^n) = \{x \in \mathbb{Z}_q^n \mid 1 \leq |\mathrm{supp}(\vec{x})| < d\}$ correspond to q-ary (n, d) codes.

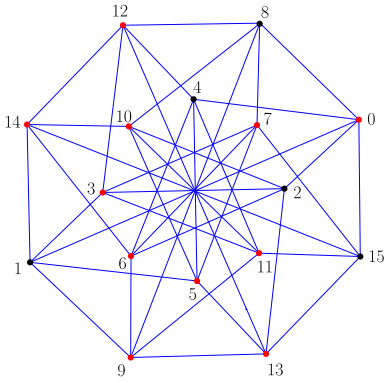


Fig. 5 The uncoloured Cayley graph $\mathrm{Cay}(\mathbb{Z}_2^4, \varphi(T_1(\mathbb{Z}_2^5)))$ with generating set $\varphi(T_1(\mathbb{Z}_2^5)) = \{(0,0,0,1), (0,0,1,0), (0,1,0,0), (1,0,0,0), (1,1,1,1)\}$. For convenience, instead of labelling the vertices of the graph by the elements of $\mathbb{Z}_2^4$, we consider the elements of $\mathbb{Z}_2^4$ as binary representations and assign the corresponding integer (e.g. $(1,0,0,0)$ maps to 8). This graph can be used in order to compute a representative set $\tilde{S_2}$ for the class $S_2 = \{\pm 1\}^5$ in the $\mathscr{L}_5 = 2D_5^*$ lattice. One such set is implied by the black vertices of the graph. The graph possesses 40 maximal independent sets of cardinality 4 and 16 maximal independent sets of cardinality 5 (maximum)

all of them. It is additionally required that for every vector $\vec{v} \in \tilde{S_2}$ also $-\vec{v} \in \tilde{S_2}$. This could be phrased as we work "modulo sign". This property can be translated algebraically by working in the quotient group $H = Z_2^n/\langle(1,\ldots,1)\rangle$ instead of $G = \mathbb{Z}_2^n$. Using the group isomorphism

$$\varphi : \mathbb{Z}_2^n/\langle(1,\ldots,1)\rangle \to \mathbb{Z}_2^{n-1}$$
$$(x_i)_{i=1}^n + \langle(1,\ldots,1)\rangle \mapsto (x_n + x_i)_{i=1}^{n-1}$$

we can transfer the problem to the graph $\Gamma_{sign} = \mathrm{Cay}(\mathbb{Z}_2^{n-1}, \varphi(T_{\lceil n/4\rceil}(G)))$. Each independent set in Γ_{sign} implies an independent set in $\Gamma_{\mathscr{L}_n}(S_2)$ which possesses the extra property of the "sign symmetry". A first remark regarding the set of maximal independent sets of Γ_{sign} is that it is invariant under the group action of $\mathbb{Z}_2^{n-1}$. For example, if we consider the graph in Fig. 5, all 16 maximal independent sets of cardinality 5 can be generated by acting with $\mathbb{Z}_2^4$ to the given independent set formed by the black vertices.

We briefly experimented with Γ_{sign} for the first few values $n = 5, 6, 7$ in order to get a first indication of how many maximal independent sets such a graph may have and how much their size can vary (Table 2).

As the number of maximal independent sets seems to grow super-exponentially in the dimension n we stopped at $n = 7$. Even though experimental results are useful in order to get intuition, theoretical results are those which give the final answer to a question. In our case there are some theoretical results, originating both from graph theory and coding theory which bound the sizes we experimented with.

– Let $\Gamma = (V, E)$ be a graph with $|V| = N$. In [26] it is proven that Γ can have up to $3^{N/3}$ maximal cliques in the worst case, a bound which is tight. Complementary this also proves that a graph Γ with N vertices can possess up to $3^{N/3}$ maximal independent sets in the worst case.

  The results in the same work also imply that the number of different sizes of maximal

Table 2 Using SAGE [9] we computed all possible sizes of a maximal independent set of $\mathrm{Cay}(\mathbb{Z}_2^{n-1}, \varphi(T_{\lceil n/4 \rceil}(\mathbb{Z}_2^n)))$ for $n = 5, 6, 7$ and the corresponding frequency of these sizes

$n = 5$								
Cardinality	4	5						
Frequency	40	16						
$n = 6$								
Cardinality	6	8	11	16				
Frequency	320	300	32	2				
$n = 7$								
Cardinality	8	14	16	17	18	19	20	22
Frequency	240	1920	625548	203840	67200	13440	2800	64

independent sets is upper bounded by $N - \log_2 N$ which is shown to be tight in the worst case.

- Let Γ be an m-regular graph with N vertices. In [30] it is shown that $\alpha(\Gamma)$ can be upper bounded by $\min\{\lfloor N/2 \rfloor, N - m\}$. This bound is obtainable. In the same work, a lower bound for $\alpha(\Gamma)$ is given, depending on m and N. However this bound is not uniform but depends on number theoretic properties of N, m. In our case the appropriate lower bound for $\alpha(\Gamma)$ would be $\lceil N/(m+1) \rceil$.
- As the graph family in question, $\Gamma_{\mathscr{L}_n}(S_2)$ (and Γ_{sign}) is specific, better upper bounds can be obtained than the general ones given in [30]. This is achieved with the use of coding theory [16]. In more detail, $\alpha(\Gamma_{\mathscr{L}_n}(S_2)) = A_2(n, \lceil n/4 \rceil)$. This equality enables the use of already known upper bounds on $A_2(n, \lceil n/4 \rceil)$ from coding theory such as the Hamming bound [15]. The lower bound implied by [30] for $\alpha(\Gamma_{\mathscr{L}_n}(S_2))$ is equivalent to the Gilbert-Varshamov bound for $A_2(n, \lceil n/4 \rceil)$.

References

1. Agrell E., Eriksson T., Vardy A., Zeger K.: Closest point search in lattices. Trans. Inform. Theory **48**(8), 2201–2214 (2002).
2. Albrecht M., Ducas L., Herold G., Kirshanova E., Postlethwaite E., Stevens M.: The general sieve kernel and new records in lattice reduction. In: Proceedings of the 38th EUROCRYPT, pp. 717–746. Springer, New York (2019).
3. Bai S., Laarhoven T., Stehlé D.: Tuple lattice sieving. LMS J. Comput. Math. **19**(A), 146–162 (2016).
4. Bonifas N., Dadush D.: Short paths on the Voronoi graph and the closest vector problem with preprocessing. In: Proceedings of the 26th SODA, pp. 295–314. ACM-SIAM, New York (2015).
5. Cabeleira F., Mariano A., Falcao G.: Memory-optimized Voronoi cell-based parallel kernels for the shortest vector problem on lattices. In: Proceedings of the 27th EUSIPCO, pp. 1–5. IEEE (2019).
6. Chen Y., Nguyen P.Q.: BKZ 2.0: better lattice security estimates. In: Proceedings of the 17th ASIACRYPT, pp. 1–20. Springer, New York (2011).
7. Conway J.H., Sloane N.J.: Low-dimensional lattices. VI. Voronoi reduction of three-dimensional lattices. In: Proceedings of the Mathematical and Physical Sciences, vol. 436, pp. 55–68. The Royal Society (1992).
8. Conway J.H., Sloane N.J.: Sphere packings, lattices and groups. Springer, New York (1998).
9. The Sage Developers: Sagemath, the Sage Mathematics Software System. https://www.sagemath.org (2019).
10. Doulgerakis E., Laarhoven T., de Weger B.: Finding closest lattice vectors using approximate Voronoi cells. In: Proceedings of the 10th PQCRYPTO, pp. 3–22. Springer, New York (2019).

11. Ducas L., Laarhoven T., van Woerden W.: The randomized slicer for CVPP: sharper, faster, smaller, batchier. In: Proceedings of the 23rd PKC, pp. 3–36. Springer, New York (2020).
12. Elkies N.: Theta functions and weighted theta functions of euclidean lattices, with some applications. http://people.math.harvard.edu/~elkies/aws09.pdf (2009).
13. Falcao G., Cabeleira F., Mariano A., Paulo Santos L.: Heterogeneous implementation of a Voronoi cell-based SVP solver. IEEE Access **7**, 127012–127023 (2019).
14. Gama N., Nguyen P.Q., Regev O.: Lattice enumeration using extreme pruning. In: Proceedings of the 29th EUROCRYPT, pp. 257–278. Springer, New York (2010).
15. Hamming R.: Error detecting and error correcting codes. Bell Syst. Tech. J. **29**(2), 147–160 (1950).
16. Hopkins M.: Representation-theoretic techniques for independence bounds of Cayley graphs. Bachelor thesis (2018).
17. Hunkenschröder C., Reuland G., Schymura M.: On compact representations of Voronoi cells of lattices. In: Proceedings of the 20th IPCO, Lecture Notes in Computer Science, vol. 11480, pp. 261–274. Springer, Berlin (2019).
18. Kabatiansky G., Levenshtein V.: Bounds for packings on a sphere and in space. Problemy Peredachi Informatsii **14**, 3–25 (1978).
19. Laarhoven T.: Sieving for closest lattice vectors (with preprocessing). In: Proceedings of the 23rd SAC, pp. 523–542. Springer, Berlin (2016).
20. Lenstra A.K., Lenstra H.W. Jr., Lovász L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4), 515–534 (1982).
21. Luby M.: A simple parallel algorithm for the maximal independent set problem. SIAM J. Comput. **15**(4), 1036–1053 (1986).
22. Micciancio D., Goldwasser S.: Complexity of Lattice Problems: A Cryptographic Perspective. Kluwer Academic Publishers, Boston (2002).
23. Micciancio D., Voulgaris P.: A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. In: Proceedings of the 42nd STOC, pp. 351–358. ACM Press, New York (2010).
24. Micciancio D., Voulgaris P.: Faster exponential time algorithms for the shortest vector problem. In: Proceedings of the 21st SODA, pp. 1468–1480. ACM-SIAM, New York (2010).
25. Minkowski H.: In: Gesammelte Abhandlungen von Hermann Minkowski, vol. 2, pp. 103–121 (1911).
26. Moon J.W., Moser L.: On cliques in graphs. Israel J. Math. **3**, 23–28 (1965).
27. Nguyen P.Q., Stehlé D.: Low-dimensional lattice basis reduction revisited. ACM Trans. Algorith. **5**(4), 46:1-46:48 (2009).
28. Nguyen P.Q., Vidick T.: Sieve algorithms for the shortest vector problem are practical. J. Math. Cryptol. **2**(2), 181–207 (2008).
29. Rajan D.S., Shende A.M.: A characterization of root lattices. Discret. Math. **161**, 309–314 (1996).
30. Rosenfeld M.: Independent sets in regular graphs. Israel J. Math. **2**, 262–272 (1964).
31. Sommer N., Feder M., Shalvi O.: Finding the closest lattice point by iterative slicing. SIAM J. Discret. Math. **23**(2), 715–731 (2009).
32. SURFsara: The Lisa cluster. https://userinfo.surfsara.nl/systems/lisa (2019).
33. The FPLLL development team: fplll, a lattice reduction library. https://github.com/fplll/fplll (2019).
34. Viterbo E., Biglieri E.: Computing the Voronoi cell of a lattice: the diamond-cutting algorithm. IEEE Trans. Inform. Theory **42**, 161–171 (1996).
35. Voronoi G.: Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs. J. Reine Angew. Math. **134**, 198–287 (1908).