Improved Analysis of Kannan's Shortest Lattice Vector Algorithm

Abstract. The security of lattice-based cryptosystems such as NTRU, GGH and Ajtai-Dwork essentially relies upon the intractability of computing a shortest non-zero lattice vector and a closest lattice vector to a given target vector in high dimensions. The best algorithms for these tasks are due to Kannan, and, though remarkably simple, their complexity estimates have not been improved since more than twenty years. Kannan's algorithm for solving the shortest vector problem is in particular crucial in Schnorr's celebrated block reduction algorithm, on which are based the best known attacks against the lattice-based encryption schemes mentioned above. Understanding precisely Kannan's algorithm is of prime importance for providing meaningful key-sizes. In this paper we improve the complexity analyses of Kannan's algorithms and discuss the possibility of improving the underlying enumeration strategy.

Keywords. Lattice reduction, complexity analysis, lattice-based cryptosystems.

1 Introduction

A lattice L is a discrete subgroup of some $\mathbb{R}^n$. Such an object can always be represented as the set of integer linear combinations of no more than n vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$. If these vectors are linearly independent, we say that they are a basis of the lattice L. The most famous algorithmic problem associated with lattices is the so-called Shortest Vector Problem (SVP). Its computational variant is to find a non-zero lattice vector of smallest Euclidean length — this length being the minimum $\lambda(L)$ of the lattice — given a basis of the lattice. Its decisional variant is known to be NP-hard under randomised reductions [2], even if one only asks for a vector whose length is no more than $2^{(\log d)^{1-\epsilon}}$ times the length of a shortest vector [12] (for any $\epsilon > 0$).

SVP is of prime importance in cryptography since a now quite large family of public-key cryptosystems rely more or less on it. The Ajtai-Dwork cryptosystem [4] relies on d^c-SVP for some $c > 0$, where $f(d)$-SVP is the problem of finding the shortest non-zero vector in the lattice L, knowing that it is unique in the sense that any vector that is of length less than $f(d) \cdot \lambda(L)$ is parallel to it. The GGH cryptosystem [11] relies on special instances of the Closest Vector Problem (CVP), a non-homogeneous version of SVP. Finally, one strongly suspects that in NTRU [15] – the only realistic lattice-based cryptosystem nowadays, the private key can be read on the coordinates of a shortest vector of the Coppersmith-Shamir lattice [8]. The best known generic attacks on these encryption schemes are based on solving SVP. It is therefore highly important to know precisely what complexity is achievable, both in theory and practice, in particular to select meaningful key-sizes.

In practice, when one wants to obtain good approximations of the lattice minimum, one uses Schnorr's block-based algorithms [24, 25]. These algorithms use internally either Kannan's algorithm, or the lattice point enumeration procedure on which it relies. This is by far the most time-consuming part of these algorithms. In fact, the corresponding routine in Shoup's NTL [26] relies on a much slower algorithm described in [25] ($2^{O(d^2)}$ instead of $d^{O(d)}$). The problem is that the enumeration is performed on a basis which is not sufficiently pre-processed (only LLL-reduced). It works well in low dimension, but it can be checked that it is sub-optimal even in moderate dimensions (say 40): the efficiency gap between enumerating from an LLL-reduced basis and from an HKZ-reduced basis shows that there is much room for improving the strategy of [25] by pre-processing the basis before starting the enumeration.

Two main algorithms are known for solving SVP. The first one, which is deterministic, is based on the exhaustive enumeration of lattice points within a small convex set. It is known as Fincke-Pohst's enumeration algorithm [9] in the algorithmic number theory community. In the cryptography community, it is known as Kannan's algorithm [16], which is quite similar to the one of

Fincke and Pohst. There are two main differences between both: firstly, in Kannan's algorithm, a long pre-computation on the basis is performed before starting the enumeration process; secondly, Kannan enumerates points in a hyper-parallelepiped whereas Fincke and Pohst do it in an hyper-ellipsoid contained in Kannan's hyper-parallelepiped – though it may be that Kannan chose the hyper-parallelepiped in order to simplify the complexity analysis. Kannan obtained a $d^{d+o(d)}$ complexity bound (in all the complexity bounds mentioned in the introduction, there is an implicit multiplicative factor that is polynomial in the bit-size of the input). In 1985, Helfrich [13] refined Kannan's analysis, and obtained a $d^{d/2+o(d)}$ complexity bound. On the other hand, Ajtai, Kumar and Sivakumar [5] described a probabilistic algorithm of complexity $2^{O(d)}$. The best exponent constant is likely to be small. Nevertheless, unless a breakthrough modification is introduced, this algorithm is bound to remain impractical even in moderate dimension since it also requires an exponential space (at least 2^d in dimension d). On the contrary, the deterministic algorithm of Kannan requires a polynomial space.

Our main result is to lower Helfrich's complexity bound on Kannan's algorithm, from $d^{\frac{d}{2}+o(d)} \approx d^{0.5 \cdot d}$ to $d^{\frac{d}{2e}+o(d)} \approx d^{0.184 \cdot d+o(d)}$. This may explain why Kannan's algorithm is tractable even in moderate dimensions (higher than 40). Our analysis can also be adapted to Kannan's algorithm that solves the Closest Vector Problem: it decreases Helfrich's complexity bound from $d^{d+o(d)}$ to $d^{d/2+o(d)}$. The complexity improvement on Kannan's SVP algorithm directly provides better worst-case efficiency/quality trade-offs in Schnorr's block-based algorithms [24, 25, 10].

It must be noted that if one follows our analysis step by step, the derived $o(d)$ may be large when evaluated for some practical d: the constants hidden in the "$o(d)$" are improvable (for some of them it may be easy, for others it is probably much harder). No effort was made to improve them, and we believe that it would have complicated the proof with irrelevant details. In fact, most of our analysis consists of estimating the number of lattice points within convex bodies, and showing that the approximation by the volume is valid. By replacing this discretisation by heuristic volume estimates, one obtains very small heuristic hidden constants.

Our complexity improvement is based on a fairly simple idea. It is equivalent to generate all lattice points within a ball and to generate all integer points within an ellipsoid (consider the ellipsoid defined by the quadratic form naturally associated with the given lattice basis). Fincke and Pohst noticed that it was more efficient to work with the ellipsoid than to consider a parallelepiped containing it: indeed, when the dimension increases, the ratio of the two volumes shrinks to 0 very quickly. Amazingly, in his analysis, instead of considering the ellipsoid, Kannan bounds the volume of the parallelepiped. Using rather involved technicalities, we bound the volume of the ellipsoid (in fact, the number of integer points within it). Some parts of our proof could be of independent interest. For example, we show that for any Hermite-Korkine-Zolotarev-reduced (HKZ-reduced for short) lattice basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$, and any subset I of $\{1, \ldots, d\}$, we have:

$$\frac{\|\boldsymbol{b}_1\|^{|I|}}{\prod_{i \in I} \|\boldsymbol{b}_i^*\|} \leq \sqrt{d}^{|I|\left(1+\log \frac{d}{|I|}\right)},$$

where $(\boldsymbol{b}_i^*)_{i \leq d}$ is the Gram-Schmidt orthogonalisation of the basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$. This inequality generalises the results of [24] on the quality of HKZ-reduced bases.

Road-Map of the Paper. In Section 2, we recall some basic definitions and properties on lattice reduction. Section 3 is devoted to the description of Kannan's algorithm and Section 4 to its complexity analysis. In Section 5, we give without much detail our sibling result on CVP, as well as very direct consequences of our result for Schnorr's block-based algorithms.

NOTATION. All logarithms are natural logarithms, i.e., $\log(e) = 1$. Let $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ be the Euclidean norm and inner product of $\mathbb{R}^n$. Bold variables are vectors. We use the bit complexity model. The notation $\mathcal{P}(n_1, \ldots, n_i)$ means $(n_1 \cdot \ldots \cdot n_i)^c$ for some constant $c > 0$. If x is real, we denote by $\lfloor x \rceil$ a closest integer to it (with any convention for making it unique) and we define the centred fractional part $\{x\}$ as $x - \lfloor x \rceil$. We use the notation $\mathrm{frac}(x)$ to denote the classical fractional part of x, i.e., the quantity $x - \lfloor x \rfloor$. Finally, for any integers a and b, we define $[\![a, b]\!]$ as $[a, b] \cap \mathbb{Z}$.

2 Background on Lattice Reduction

We assume the reader is familiar with the geometry of numbers and its algorithmic aspects. Complete introductions to Euclidean lattices algorithmic problems can be found in [21] and [23].

Gram-Schmidt orthogonalisation. Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ be linearly independent vectors. Their *Gram-Schmidt orthogonalisation* (GSO) $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_d^*$ is the orthogonal family defined recursively as follows: the vector $\boldsymbol{b}_i^*$ is the component of the vector $\boldsymbol{b}_i$ which is orthogonal to the linear span of the vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}$. We have $\boldsymbol{b}_i^* = \boldsymbol{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \boldsymbol{b}_j^*$ where $\mu_{i,j} = \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^* \rangle}{\|\boldsymbol{b}_j^*\|^2}$. For $i \leq d$ we let $\mu_{i,i} = 1$. Notice that the GSO family depends on the order of the vectors. If the $\boldsymbol{b}_i$'s are integer vectors, the $\boldsymbol{b}_i^*$'s and the $\mu_{i,j}$'s are rational.

Lattice volume. The volume of a lattice L is defined as $\det(L) = \prod_{i=1}^d \|\boldsymbol{b}_i^*\|$, where the $\boldsymbol{b}_i$'s are any basis of L. It does not depend on the choice of the basis of L and can be interpreted as the geometric volume of the parallelepiped naturally spanned by the basis vectors.

Minimum and SVP. Another important lattice invariant is the minimum. The *minimum* $\lambda(L)$ is the radius of the smallest closed ball centred at the origin containing at least one non-zero lattice vector. The most famous lattice problem is the *shortest vector problem*. We give here its computational variant: given a basis of a lattice L, find a lattice vector whose norm is exactly $\lambda(L)$.

CVP. We give here the computational variant of the *closest vector problem*: given a basis of a lattice L and a target vector in the real span of L, find a closest vector of L to the target vector.

The volume and the minimum of a lattice cannot behave independently. Hermite [14] was the first to bound the ratio $\frac{\lambda(L)}{(\det L)^{1/d}}$ as a function of the dimension only, but his bound was later on greatly improved by Minkowski in his *Geometrie der Zahlen* [22]. *Hermite's constant* γ_d is defined as the supremum over d dimensional lattices L of the ratio $\frac{\lambda(L)^2}{(\det L)^{2/d}}$. In particular, we have $\gamma_d \leq \frac{d+4}{4}$ (see [19]), which we will refer to as *Minkowski's theorem*. Unfortunately, the proof of Minkowski's theorem is not constructive. In practice, one often starts with a lattice basis, and tries to improve its quality. This process is called lattice reduction. The most usual ones are probably the LLL and HKZ reductions. Before defining them, we need the concept of size-reduction.

Size-reduction. A basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ is *size-reduced* if its GSO family satisfies $|\mu_{i,j}| \leq 1/2$ for all $1 \leq j < i \leq d$.

HKZ-reduction. A basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ is said to be *Hermite-Korkine-Zolotarev-reduced* if it is size-reduced, the vector $\boldsymbol{b}_1$ reaches the first lattice minimum, and the projections of the $(\boldsymbol{b}_i)_{i \geq 2}$'s orthogonally to the vector $\boldsymbol{b}_1$ are an HKZ-reduced basis. The following immediately follows from this definition and Minkowski's theorem. It is the sole property on HKZ-reduced bases that we will use:

Lemma 1. *If $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ is HKZ-reduced, then for any $i \leq d$, we have:*

$$\|\boldsymbol{b}_i^*\| \leq \sqrt{\frac{d-i+5}{4}} \cdot \left(\prod_{j \geq i} \|\boldsymbol{b}_j^*\| \right)^{\frac{1}{d-i+1}}.$$

HKZ-reduction is very strong, but very expensive to compute. On the contrary, LLL-reduction is fairly cheap, but an LLL-reduced basis is of much lower quality.

LLL-reduction [18]. A basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ is *LLL-reduced* if it is size-reduced and if its GSO satisfies the $(d-1)$ Lovász conditions: $\frac{3}{4} \cdot \left\|\boldsymbol{b}_{\kappa-1}^*\right\|^2 \leq \left\|\boldsymbol{b}_\kappa^* + \mu_{\kappa,\kappa-1}\boldsymbol{b}_{\kappa-1}^*\right\|^2$. The LLL-reduction implies that the norms $\|\boldsymbol{b}_1^*\|, \ldots, \|\boldsymbol{b}_d^*\|$ of the GSO vectors never drop too fast: intuitively, the vectors are not far from being orthogonal. Such bases have useful properties, like providing exponential approximations to SVP and CVP. In particular, their first vector is relatively short. More precisely:

Theorem 1 ([18]). *Let $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ be an LLL-reduced basis of a lattice L. Then we have $\|\boldsymbol{b}_1\| \leq 2^{\frac{d-1}{4}} \cdot (\det L)^{1/d}$. Moreover, there exists an algorithm that takes as input any set of integer vectors and outputs in deterministic polynomial time an LLL-reduced basis of the lattice they span.*

In the following, we will also need the fact that if the set of vectors given as input to the LLL algorithm starts with a shortest non-zero lattice vector, then this vector is not changed during the execution of the algorithm: the output basis starts with the same vector.

3 Kannan's SVP Algorithm

Kannan's SVP algorithm [16] relies on multiple calls to the so-called short lattice points enumeration procedure. The latter aims at computing all vectors of a given lattice that are in the hyper-sphere centred in $\mathbf{0}$ and some prescribed radius. Variants of the enumeration procedure are described in [1].

3.1 Short Lattice Points Enumeration

Let $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ be a basis of a lattice $L \subset \mathbb{Z}^n$ and let $A \in \mathbb{Z}$. Our goal is to find all lattice vectors $\sum_{i=1}^d x_i \boldsymbol{b}_i$ of squared Euclidean norm $\leq A$. The enumeration works as follows. Suppose that $\|\sum_i x_i \boldsymbol{b}_i\|^2 \leq A$ for some integers x_i's. Then, by considering the components of the vector $\sum_i x_i \boldsymbol{b}_i$ on each of the $\boldsymbol{b}_i^*$'s, we obtain:

$$\begin{aligned}
(x_d)^2 \cdot \|\boldsymbol{b}_d^*\|^2 &\leq A, \\
(x_{d-1} + \mu_{d,d-1}x_d)^2 \cdot \|\boldsymbol{b}_{d-1}^*\|^2 &\leq A - (x_d)^2 \cdot \|\boldsymbol{b}_d^*\|^2, \\
&\cdots \\
\left(x_i + \sum_{j=i+1}^d \mu_{j,i} x_j\right)^2 \cdot \|\boldsymbol{b}_i^*\|^2 &\leq A - \sum_{j=i+1}^d l_j, \\
&\cdots \\
\left(x_1 + \sum_{j=2}^d \mu_{j,i} x_j\right)^2 \cdot \|\boldsymbol{b}_1\|^2 &\leq A - \sum_{j=2}^d l_j,
\end{aligned}$$

where $l_i = (x_i + \sum_{j>i} x_j \mu_{j,i})^2 \cdot \|\boldsymbol{b}_i^*\|^2$. The algorithm of Figure 1 mimics the equations above. It is easy to see that the bit-cost of this algorithm is bounded by the number of loop iterations times a polynomial in the bit-size of the input. We will prove that if the input basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ is sufficiently reduced and if $A = \|\boldsymbol{b}_1\|^2$, then the number of loop iterations is $d^{\frac{d}{2e}+o(d)}$.

```
Input: An integral lattice basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$, a bound $A \in \mathbb{Z}$.
Output: All vectors in $L(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ that are of squared norm $\leq A$.
 1. Compute the rational $\mu_{i,j}$'s and $\|\boldsymbol{b}_i^*\|^2$'s.
 2. $\boldsymbol{x}{:=}\boldsymbol{0}, \boldsymbol{l}{:=}\boldsymbol{0}, S{:=}\emptyset$.
 3. $i{:=}1$. While $i \leq d$, do
 4.    $l_i{:=}(x_i + \sum_{j>i} x_j \mu_{j,i})^2 \|\boldsymbol{b}_i^*\|^2$.
 5.    If $i = 1$ and $\sum_{j=1}^{d} l_j \leq A$, then $S{:=}S \cup \{\boldsymbol{x}\}$, $x_1{:=}x_1 + 1$.
 6.    If $i \neq 1$ and $\sum_{j \geq i} l_j \leq A$, then
 7.       $i{:=}i - 1$, $x_i{:=}\left\lceil -\sum_{j>i}(x_j \mu_{j,i}) - \sqrt{\frac{A - \sum_{j>i} l_j}{\|\boldsymbol{b}_i^*\|^2}} \right\rceil$.
 8.    If $\sum_{j \geq i} l_j > A$, then $i{:=}i + 1$, $x_i{:=}x_i + 1$.
 9. Return $S$.
```

Fig. 1. The Enumeration Algorithm.

3.2 Solving SVP

To solve SVP, Kannan provides an algorithm that computes HKZ-reduced bases, see Figure 2. The cost of the enumeration procedure dominates the overall cost and mostly depends on the quality (i.e., the slow decrease of the $\|\boldsymbol{b}_i^*\|$'s) of the input basis. The main idea of Kannan's algorithm is thus to spend a lot of time pre-computing a basis of excellent quality before calling the enumeration procedure. More precisely, it pre-computes a basis which satisfies the following definition:

Definition 1 (Quasi-HKZ-Reduction). *A basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ is quasi-HKZ-reduced if it is size-reduced, if $\|\boldsymbol{b}_2^*\| \geq \|\boldsymbol{b}_1^*\|/2$ and if once projected orthogonally to $\boldsymbol{b}_1$, the other $\boldsymbol{b}_i$'s are HKZ-reduced.*

```
Input: An integer lattice basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$.
Output: An HKZ-reduced basis of the same lattice.
 1. LLL-reduce the basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$.
 2. Do
 3.    Compute the projections $(\boldsymbol{b}_i')_{i \geq 2}$ of the $\boldsymbol{b}_i$'s orthogonally to $\boldsymbol{b}_1$.
 4.    HKZ-reduce the $(d-1)$-dimensional basis $(\boldsymbol{b}_2', \ldots, \boldsymbol{b}_d')$.
 5.    Extend the obtained $(\boldsymbol{b}_i')_{i \geq 2}$'s into vectors of $L$ by adding to them rational multiples of $\boldsymbol{b}_1$, in such a way that we have $|\mu_{i,1}| \leq 1/2$ for any $i > 1$.
 6. While $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ is not quasi-HKZ-reduced.
 7. Call the enumeration procedure to find all lattice vectors of length $\leq \|\boldsymbol{b}_1\|$. Let $\boldsymbol{b}_0$ be a shortest non-zero vector among them.
 8. $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d){:=}\mathrm{LLL}(\boldsymbol{b}_0, \ldots, \boldsymbol{b}_d)$.
 9. Compute the projections $(\boldsymbol{b}_i')_{i \geq 2}$'s of the $\boldsymbol{b}_i$'s orthogonally to the vector $\boldsymbol{b}_1$.
 10. HKZ-reduce the $(d-1)$-dimensional basis $(\boldsymbol{b}_2', \ldots, \boldsymbol{b}_d')$.
 11. Extend the obtained $(\boldsymbol{b}_i')_{i \geq 2}$'s into vectors of $L$ by adding to them rational multiples of $\boldsymbol{b}_1$, in such a way that we have $|\mu_{i,1}| \leq 1/2$ for any $i > 1$.
```

Fig. 2. Kannan's SVP Algorithm.

Several comments need to be made on the algorithm of Figure 2. Steps 4 and 10 are recursive calls. Nevertheless, one should be careful because the $\boldsymbol{b}_i'$'s are rational vectors, whereas the input of the algorithm must be integral. One must therefore scale the vectors by a common factor. Steps 5 and 11 can be performed for example by expressing the reduced basis vectors as integer linear combinations of the initial ones, using these coefficients to recover lattice vectors and subtracting a correct multiple of the vector $\boldsymbol{b}_1$. In Step 7, it is alway possible to choose such a vector $\boldsymbol{b}_0$, since this enumeration always provides non-zero solutions (the vector $\boldsymbol{b}_1$ is a one of them).

3.3 Cost of Kannan's SVP Solver

We recall briefly Helfrich's complexity analysis [13] of Kannan's algorithm and explain our complexity improvement. Let $C(d,n,B)$ be the worst-case complexity of the algorithm of Figure 2 when given as input a d-dimensional basis which is embedded in $\mathbb{Z}^n$ and whose coefficients are smaller than B in absolute value. Kannan [16] and Helfrich [13] show the following properties:

- It computes an HKZ-reduced basis of the lattice spanned by the input vectors.
- All arithmetic operations performed during the execution are of cost $\mathcal{P}(d,n,\log B)$. This implies that the cost $C(d,n,B)$ can be bounded by $C(d)\cdot\mathcal{P}(\log B,n)$ for some function $C(d)$.
- The number of iterations of the loop of Steps 2–6 is bounded by $O(1)+\log d$.
- The cost of the call to the enumeration procedure at Step 7 is bounded by $\mathcal{P}(\log B,n)\cdot d^{d/2+o(d)}$.

From these properties and those of the LLL algorithm as recalled in the previous section, it is easy to obtain the following equation:

$$C(d)\leq (O(1)+\log d)(C(d-1)+\mathcal{P}(d))+\mathcal{P}(d)+d^{\frac{d}{2}+o(d)}.$$

One can then derive the bound $C(d,B,n)\leq\mathcal{P}(\log B,n)\cdot d^{\frac{d}{2}+o(d)}$.

The main result of this paper is to improve this complexity upper bound to $\mathcal{P}(\log B,n)\cdot d^{\frac{d}{2e}+o(d)}$. In fact, we show the following:

Theorem 2. *Given as inputs a quasi-HKZ-reduced basis $(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_d)$ and $A=\|\boldsymbol{b}_1\|^2$, the number of loop iterations during the execution of the enumeration algorithm as described in Figure 1 is bounded by $\mathcal{P}(\log B)\cdot 2^{O(d)}\cdot d^{\frac{d}{2e}}$, where $B=\max_i\|\boldsymbol{b}_i\|$. As a consequence, given a d-dimensional basis of n-dimensional vectors whose entries are integers with absolute values $\leq B$, one can compute an HKZ-reduced basis of the lattice they span in deterministic time $\mathcal{P}(\log B,n)\cdot d^{\frac{d}{2e}+o(d)}$.*

4 Complexity of the Enumeration Procedure

This section is devoted to proving Theorem 2.

4.1 From the Enumeration Procedure to Integer Points in Hyper-ellipsoids

In this subsection, we do not assume anything on the input basis $(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_d)$ and on the input bound A. Up to some polynomial in d and $\log B$, the complexity of the enumeration procedure of Figure 1 is the number of loop iterations. This number of iterations is itself bounded by:

$$\sum_{i=1}^{d}\left|\left\{(x_i,\ldots,x_d)\in\mathbb{Z}^{d-i+1},\|\sum_{j=i}^{d}x_j\boldsymbol{b}_j^{(i)}\|^2\leq A\right\}\right|,$$

where $\boldsymbol{b}_j^{(i)} = \boldsymbol{b}_j - \sum_{k<i} \mu_{j,k} \boldsymbol{b}_k^*$ is the vector $\boldsymbol{b}_j$ once projected orthogonally to the linear span of the vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}$. Indeed, the truncated coordinate $(x_i, \ldots, x_d)$ is either a valid one, i.e., we have $|\sum_{j=i}^d x_j \boldsymbol{b}_j^{(i)}\|^2 \leq A$, or $(x_i - 1, \ldots, x_d)$ is a valid one, or $(x_{i+1}, \ldots, x_d)$ is a valid one. In fact, if $(x_i, \ldots, x_d)$ is a valid truncated coordinate, only two non-valid ones related to that one can possibly be considered during the execution of the algorithm: $(x_i + 1, \ldots, x_d)$ and $(x_{i-1}, x_i \ldots, x_d)$ for at most one integer x_{i-1}.

Consider the quantity $\left|\left\{(x_i, \ldots, x_d) \in \mathbb{Z}^{d-i+1}, \| \sum_{j=i}^d x_j \boldsymbol{b}_j^{(i)}\|^2 \leq A\right\}\right|$. By applying the change of variable $x_j \leftarrow x_j - \left\lfloor \sum_{k>j} \mu_{k,j} x_k \right\rceil$, we obtain:

$$\begin{aligned}
\sum_{i \leq d} |\{(x_i, \ldots, x_d) \in \mathbb{Z}^{d-i+1} \ , \ \| \sum_{j \geq i} x_j \boldsymbol{b}_j^{(i)} \|^2 \leq A\}| \\
&\leq \sum_{i \leq d} |\{(x_i, \ldots, x_d) \in \mathbb{Z}^{d-i+1}, \sum_{j \geq i} (x_j + \sum_{k>j} \mu_{k,j} x_k)^2 \cdot \|\boldsymbol{b}_j^*\|^2 \leq A\}| \\
&\leq \sum_{i \leq d} |\{(x_i, \ldots, x_d) \in \mathbb{Z}^{d-i+1}, \sum_{j \geq i} (x_j + \{\sum_{k>j} \mu_{k,j} x_k \ \})^2 \cdot \|\boldsymbol{b}_j^*\|^2 \leq A\}|.
\end{aligned}$$

If x is an integer and $\epsilon \in [-1/2, 1/2]$, then we have the relation $(x + \epsilon)^2 \geq x^2/4$. If $x = 0$, this is obvious, and otherwise we use the inequality $|\epsilon| \leq 1/2 \leq |x|/2$. As a consequence, up to a polynomial factor, the complexity of the enumeration is bounded by:

$$\sum_{i \leq d} \left| \left\{ (x_i, \ldots, x_d) \in \ \mathbb{Z}^{d-i+1}, \sum_{j \geq i} x_j^2 \cdot \|\boldsymbol{b}_j^*\|^2 \leq 4A \right\} \right|.$$

For any $i \leq d$, we define the ellipsoid $\mathcal{E}_i = \left\{(y_i, \ldots, y_d) \in \mathbb{R}^{d-i+1}, \sum_{j \geq i} y_j^2 \cdot \|\boldsymbol{b}_j^*\|^2 \leq 4A\right\}$, as well as the quantity $N_i = |\mathcal{E}_i \cap \mathbb{Z}^{d-i+1}|$. We want to bound the sum of the N_i's. We now fix some index i. The following sequence of relations is inspired from [20, Lemma 1].

$$\begin{aligned}
N_i = \sum_{(x_i, \ldots, x_d) \in \mathbb{Z}^{d-i+1}} \mathbf{1}_{\mathcal{E}_i}(x_i, \ldots, x_d) &\leq \exp\left(d\left(1 - \sum_{j \geq i} x_j^2 \frac{\|\boldsymbol{b}_j^*\|^2}{4A}\right)\right) \\
&\leq e^d \cdot \prod_{j \geq i} \sum_{x \in \mathbb{Z}} \exp\left(-x^2 \frac{d\|\boldsymbol{b}_i^*\|^2}{4A}\right) = e^d \cdot \prod_{j \geq i} \Theta\left(\frac{d\|\boldsymbol{b}_j^*\|^2}{4A}\right),
\end{aligned}$$

where $\Theta(t) = \sum_{x \in \mathbb{Z}} \exp(-tx^2)$ is defined for $t > 0$. Notice that $\Theta(t) = 1 + 2\sum_{x \geq 1} \exp(-tx^2) \leq 1 + 2\int_0^\infty \exp(-tx^2) dx = 1 + \sqrt{\frac{\pi}{t}}$. Hence $\Theta(t) \leq \frac{1+\sqrt{\pi}}{\sqrt{t}}$ for $t \leq 1$ and $\Theta(t) \leq 1 + \sqrt{\pi}$ for $t \geq 1$. As a consequence, we have:

$$N_i \leq (4e(1 + \sqrt{\pi}))^d \cdot \prod_{j \geq i} \max\left(1, \frac{\sqrt{A}}{\sqrt{d}\|\boldsymbol{b}_i^*\|}\right). \tag{1}$$

One thus concludes that the cost of the enumeration procedure is bounded by:

$$\mathcal{P}(n, \log A, \log B) \cdot 2^{O(d)} \cdot \max_{I \subset [\![1,d]\!]} \left(\frac{(\sqrt{A})^{|I|}}{(\sqrt{d})^{|I|} \prod_{i \in I} \|\boldsymbol{b}_i^*\|}\right).$$

4.2 The Case of Quasi-HKZ-Reduced Bases

We know suppose that $A = \|\boldsymbol{b}_1\|^2$ and that the input basis $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_d)$ is quasi-HKZ-reduced. Our first step is to strengthen the quasi-HKZ-reducedness hypothesis to an HKZ-reducedness hypothesis. Let $I \subset [\![1, d]\!]$. If $1 \notin I$, then, because of the quasi-HKZ-reducedness assumption:

$$\frac{\|\boldsymbol{b}_1\|^{|I|}}{(\sqrt{d})^{|I|}\prod_{i\in I}\|\boldsymbol{b}_i^*\|} \leq 2^d \frac{\|\boldsymbol{b}_2^*\|^{|I|}}{(\sqrt{d})^{|I|}\prod_{i\in I}\|\boldsymbol{b}_i^*\|}.$$

Otherwise if $1 \in I$, then we have, by removing $\|\boldsymbol{b}_1^*\|$ from the product $\prod_{i\in I-\{1\}}\|\boldsymbol{b}_i^*\|$:

$$\frac{\|\boldsymbol{b}_1\|^{|I|}}{(\sqrt{d})^{|I|}\prod_{i\in I}\|\boldsymbol{b}_i^*\|} \leq 2^d \frac{\|\boldsymbol{b}_2^*\|^{|I|-1}}{(\sqrt{d})^{|I|-1}\prod_{i\in I-\{1\}}\|\boldsymbol{b}_i^*\|}.$$

As a consequence, in order to obtain Theorem 2, it suffices to prove the following:

Theorem 3. *Let $\boldsymbol{b}_1, \dots, \boldsymbol{b}_d$ be an HKZ-reduced basis. Let $I \subset [\![1, d]\!]$. Then,*

$$\frac{\|\boldsymbol{b}_1\|^{|I|}}{\prod_{i\in I}\|\boldsymbol{b}_i^*\|} \leq (\sqrt{d})^{|I|\left(1+\log\frac{d}{|I|}\right)} \leq (\sqrt{d})^{\frac{d}{e}+|I|}.$$

4.3 A Property on the Geometry of HKZ-Reduced Bases

In this section, we prove Theorem 3, which is the last missing part to obtain the announced result. Some parts of the proof are fairly technical and have been postponed to the appendix (this is the case for the proofs of Lemmata 2–5). As a guide, the reader should consider the typical case where $(\boldsymbol{b}_i)_{1\leq i\leq d}$ is an HKZ-reduced basis for which $(\|\boldsymbol{b}_i^*\|)_i$ is a non-increasing sequence. In that case, the shape of the interval I that is provided by Equation(1) is much simpler: it is an interval $[\![i, d]\!]$ starting at some index i. Lemmata 4 and 2 (which should thus be considered as the core of the proof) and the fact that $x \log x \geq -1/e$ for $x \in [0, 1]$ are sufficient to deal with such simple intervals, and thus to provide the result.

The difficulties arise when the shape of the set I under study becomes more complicated. Though the proof is technically quite involved, the strategy itself can be summed up in a few words. We split our HKZ-reduced basis into *blocks* (defined by the expression of I as a union of intervals), i.e., groups of consecutive vectors $b_i, b_{i+1}, \dots, b_{j-1}$ such that $i, \dots, k-1 \notin I$ and $k, \dots, j-1 \in I$. The former vectors will be the "large ones", and the latter the "small ones". Over each block, Lemma 4 relates the average size of the small vectors to the average size of the whole block. We consider the blocks by decreasing indices (in Lemma 6), and use an amortised analysis to combine finely the local behaviours on blocks to obtain a global bound. This recombination is extremely tight, and in order to get the desired bound we use "parts of vectors" (non-integral powers of them). This is why we need to introduce the $\tilde{\pi}$ (in Definition 3). A final convexity argument provided by Lemma 3 gives the result.

In the sequel, $(\boldsymbol{b}_i)_{1\leq i\leq d}$ is an HKZ-reduced basis of a lattice L of dimension $d \geq 2$.

Definition 2. *For any $I \subset [\![1, d]\!]$, we define $\pi_I = \left(\prod_{i\in I}\|\boldsymbol{b}_i^*\|\right)^{\frac{1}{|I|}}$. Moreover, if $k \in [\![1, d-1]\!]$, we define $\Gamma_d(k) = \prod_{i=d-k}^{d-1}\gamma_{i+1}^{\frac{1}{2i}}$.*

For technical purposes in the proof of Lemma 6, we also need the following definition.

Definition 3. *If $1 \leq a < b \leq d$, where a is real and b is an integer, we define:*

$$\tilde{\pi}_{[a,b]} = \left(\|\boldsymbol{b}^*_{\lfloor a \rfloor}\|^{1-a+\lfloor a \rfloor} \cdot \prod_{i=\lfloor a \rfloor+1}^{b} \|\boldsymbol{b}^*_i\| \right)^{\frac{1}{b+1-a}} = \left(\pi_{[\![\lfloor a \rfloor, b]\!]}\right)^{\frac{(b+1-\lfloor a \rfloor)(1-a+\lfloor a \rfloor)}{b+1-a}} \cdot \left(\pi_{[\![\lfloor a \rfloor+1, b]\!]}\right)^{\frac{(b-\lfloor a \rfloor)(a-\lfloor a \rfloor)}{b+1-a}}.$$

Note that Definition 3 naturally extends Definition 2, since $\tilde{\pi}_{[a,b]} = \pi_{[\![a,b]\!]}$ when a is an integer.

We need estimates on the order of magnitude of $\varGamma$, and a technical lemma allowing us to recombine such estimates. Basically, the following lemma is a precise version of the identity:

$$\log \varGamma_d(k) \approx \int_{x=d-k}^{d} \frac{x}{2} \log x \,\mathrm{d}x \approx \frac{\log^2(d) - \log^2(d-k)}{4} \lesssim \frac{\log d}{2} \log \frac{d}{d-k}.$$

Lemma 2. *For all $1 \leq k < d$, we have $\varGamma_d(k) \leq \sqrt{d}^{\log \frac{d}{d-k}}$.*

The following lemma derives from the convexity of the function $x \mapsto x \log x$.

Lemma 3. *Let $\varDelta \geq 1$, and define $F_\varDelta(k,d) = \varDelta^{-k \log \frac{k}{d}}$. We have, for all integer t, for all integers $k_1, \ldots, k_t$ and $d_1, \ldots, d_t$ such that $1 \leq k_i < d_i$ for all $i \leq t$,*

$$\prod_{i \leq t} F_\varDelta(k_i, d_i) \leq F_\varDelta\left(\sum_{i \leq t} k_i, \sum_{i \leq t} d_i\right).$$

We now give an "averaged" version of [24, Lemma 4]. For completeness, we give its proof in appendix. This provides the result claimed in Theorem 3 for any interval $I = [\![i,j]\!]$, for any $i \leq j \leq d$.

Lemma 4. *For all $k \in [\![0, d-1]\!]$, we have*

$$\pi_{[\![1,k]\!]} \leq (\varGamma_d(k))^{d/k} \cdot \pi_{[\![k+1,d]\!]} \quad \text{and} \quad \pi_{[\![k+1,d]\!]} \geq (\varGamma_d(k))^{-1} \cdot (\det L)^{1/d} \geq \sqrt{d}^{\log \frac{d-k}{d}} (\det L)^{1/d}.$$

The following lemma extends Lemma 4 to the case where k is not necessarily an integer. Its proof is conceptually simple, but involves rather heavy elementary calculus. It would be simpler to obtain it with a relaxation factor. The result is nevertheless worth the effort since the shape of the bound is extremely tractable in the sequel.

Lemma 5. *If $1 \leq x_1 < x_2 < d$ are real and in $[1, d)$, then $\tilde{\pi}_{[x_2,d]} \geq \sqrt{d}^{\log \frac{d-x_2}{d-x_1}} \cdot \tilde{\pi}_{[x_1,d]}$.*

We prove Theorem 3 by induction on the number of intervals occurring in the expression of the set I as a union of intervals. The following lemma is the induction step. This is a recombination step, where we join one block (between the indices 1 and v, the "small vectors" being those between $u+1$ and v) to one or more already considered blocks on its right. An important point is to ensure that the densities δ_i defined below actually decrease.

Lemma 6. *Let $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ be an HKZ-reduced basis. Let $v \in [\![2, d]\!]$, $I \subset [\![v+1, d]\!]$ and $u \in [\![1, v]\!]$. Assume that:*

$$\pi_I^{|I|} \geq \prod_{i<t} \left(\pi^{|I_i|}_{[\![\alpha_i+1, \alpha_{i+1}]\!]} \cdot \sqrt{d}^{|I_i| \log \delta_i}\right),$$

where $I_i = I \cap \llbracket \alpha_i + 1, \alpha_{i+1} \rrbracket, \delta_i = \frac{|I_i|}{\alpha_{i+1}-\alpha_i}$ *is the density of* I *in* $\llbracket \alpha_i + 1, \alpha_{i+1} \rrbracket$*, and the integers* t *and* α_i*'s, and the densities* δ_i *satisfy* $t \geq 1$*,* $v = \alpha_1 < \alpha_2 < \ldots < \alpha_t \leq d$ *and* $1 \geq \delta_1 > \ldots > \delta_{t-1} > 0$*.*

Then, we have

$$\pi_{I'}^{|I'|} \geq \prod_{i<t'} \left(\pi_{\llbracket \alpha'_i+1, \alpha'_{i+1} \rrbracket}^{|I'_i|} \cdot \sqrt{d}^{|I'_i| \log \delta'_i} \right),$$

where $I' = \llbracket u+1, v \rrbracket \cup I, I'_i = I' \cap \llbracket \alpha'_i + 1, \alpha'_{i+1} \rrbracket, \delta'_i = \frac{|I'_i|}{\alpha'_{i+1}-\alpha'_i}$ *and the integers* t' *and* α'_i*'s, and the densities* δ'_i *satisfy* $t' \geq 1$*,* $0 = \alpha'_1 < \alpha'_2 < \ldots < \alpha'_{t'} \leq d$ *and* $1 \geq \delta'_1 > \ldots > \delta'_{t'-1} > 0$*.*

Proof. Assume first that $\frac{v-u}{v} \geq \delta_1$, Then, thanks to Lemma 4,

$$\pi_{I'}^{|I'|} = \pi_{\llbracket u+1,v \rrbracket}^{v-u} \cdot \pi_I^{|I|} \geq \pi_{\llbracket 1,v \rrbracket}^{v-u} \cdot \sqrt{d}^{(v-u)\frac{v-u}{v}} \cdot \pi_I^{|I|},$$

we are done with $t' = t+1$, $\alpha'_1 = 1$, $\alpha'_k = \alpha_{k-1}$, $\delta'_1 = \frac{v-u}{v}$, $\delta'_k = \delta_{k-1}$.

Otherwise, we let $\lambda_1 > 0$ be such that $\frac{v-u}{v-\lambda_1} = \delta_1 = \frac{v-u+|I_1|}{\alpha_2 - \lambda_1}$, where the first equality defines λ_1 and the second one follows. Note that this implies:

$$\tilde{\pi}_{[\lambda_1, v]}^{v-u} \cdot \pi_{\llbracket v+1, \alpha_2 \rrbracket}^{|I_1|} = \tilde{\pi}_{[\lambda_1, \alpha_2]}^{v-u+|I_1|}.$$

Then, we have, by using Lemma 5,

$$\begin{aligned}
\pi_{I'}^{|I'|} &= \pi_{\llbracket u+1,v \rrbracket}^{v-u} \cdot \pi_I^{|I|} \\
&\geq \left(\tilde{\pi}_{[\lambda_1,v]}^{v-u} \cdot \sqrt{d}^{(v-u)\log\frac{v-u}{v-\lambda_1}} \right) \cdot \prod_{i<t} \left(\pi_{\llbracket \alpha_i+1, \alpha_{i+1} \rrbracket}^{|I_i|} \cdot \sqrt{d}^{|I_i| \log \delta_i} \right) \\
&\geq \left(\tilde{\pi}_{[\lambda_1,v]}^{v-u} \cdot \pi_{\llbracket v+1, \alpha_2 \rrbracket}^{|I_1|} \cdot \sqrt{d}^{(v-u)\log\frac{v-u}{v-\lambda_1} + |I_1| \cdot \log \delta_1} \right) \cdot \prod_{i=2}^{t-1} \left(\pi_{\llbracket \alpha_i+1, \alpha_{i+1} \rrbracket}^{|I_i|} \cdot \sqrt{d}^{|I_i| \log \delta_i} \right) \\
&\geq \left(\tilde{\pi}_{[\lambda_1,\alpha_2]}^{v-u+|I_1|} \cdot \sqrt{d}^{(v-u+|I_1|)\log\frac{v-u+|I_1|}{\alpha_2-\lambda_1}} \right) \cdot \prod_{i=2}^{t-1} \left(\pi_{\llbracket \alpha_i+1, \alpha_{i+1} \rrbracket}^{|I_i|} \cdot \sqrt{d}^{|I_i| \log \delta_i} \right),
\end{aligned}$$

If $\frac{v-u+|I_1|}{\alpha_2} > \frac{|I_2|}{\alpha_3-\alpha_2}$, we conclude as in the first step, putting $t' = t$, $\alpha'_1 = 1$, $\alpha'_k = \alpha_k$ for $k \geq 2$, $\delta'_1 = (v-u+|I_1|)/\alpha_2$, $\delta'_k = \delta_k$ for $k \geq 2$. If this is not the case, we let λ_2 be such that:

$$\frac{v-u+|I_1|}{\alpha_2 - \lambda_2} = \delta_2 = \frac{v-u+|I \cap \llbracket \alpha_1+1, \alpha_3 \rrbracket|}{\alpha_3 - \lambda_2}.$$

Notice that since $\delta_1 = \frac{v-u+|I_1|}{\alpha_2-\lambda_1} > \delta_2$, we have $\lambda_2 < \lambda_1$. A similar sequence of inequalities, using Lemma 5 to relate $\tilde{\pi}_{[\lambda_1,\alpha_2]}$ to $\tilde{\pi}_{[\lambda_2,\alpha_2]}$, leads to the following lower bound on $\pi_{I'}^{|I'|}$:

$$\left(\tilde{\pi}_{[\lambda_2,\alpha_3]}^{v-u+|I\cap\llbracket\alpha_1+1,\alpha_3\rrbracket|} \cdot \sqrt{d}^{(v-u+|I\cap\llbracket\alpha_1+1,\alpha_3\rrbracket|)\log\frac{v-u+|I\cap\llbracket\alpha_1+1,\alpha_3\rrbracket|}{\alpha_3-\lambda_2}} \right) \cdot \prod_{i=3}^{t-1} \left(\pi_{\llbracket \alpha_i+1, \alpha_{i+1} \rrbracket}^{|I_i|} \cdot \sqrt{d}^{|I_i| \log \delta_i} \right)$$

We can proceed in the same way, constructing $\lambda_2 > \lambda_3 > \ldots$. Suppose first that the construction stops at some point. We have:

$$\pi_{I'}^{|I'|} \geq \left(\pi_{\llbracket 1, \alpha_{k+1} \rrbracket}^{|I' \cap \llbracket 1,\alpha_{k+1} \rrbracket|} \cdot \sqrt{d}^{|I'\cap\llbracket 1,\alpha_{k+1}\rrbracket| \log \frac{|I'\cap\llbracket 1,\alpha_{k+1}\rrbracket|}{\alpha_{k+1}}} \right) \cdot \prod_{i=k+1}^{t-1} \left(\pi_{\llbracket \alpha_i+1, \alpha_{i+1} \rrbracket}^{|I_i|} \sqrt{d}^{|I_i| \log \delta_i} \right).$$

We can then conclude, by putting $t' = t - k + 1$, $\alpha'_1 = 1, \alpha'_j = \alpha_{j+k-1}$ for $j > 1$, $\delta'_1 = |I' \cap [\![1, \alpha_{k+1}]\!]\,|/\alpha_{k+1}$, $\delta'_j = \delta_{j+k-1}$ for $j > 1$.

Otherwise, we end up with:

$$\pi_{I'}^{|I'|} \geq \tilde{\pi}_{[\lambda_{t-2},\alpha_{t-1}]}^{|I'|} \cdot \sqrt{d}^{\,|I'|\log\frac{|I'\cap[\![1,\alpha_{t-1}]\!]|}{\alpha_{t-1}-\lambda_{t-2}}},$$

to which we can apply Lemma 5 to obtain $\pi_{I'}^{|I'|} \geq \pi_{[\![1,\alpha_{t-1}]\!]}^{|I'|} \cdot \sqrt{d}^{\,|I'|\log\frac{|I'\cap[\![1,\alpha_{t-1}]\!]|}{\alpha_{t-1}}}$, which is again in the desired form, with $t' = 2$, $\alpha'_1 = 1$, $\alpha'_2 = \alpha_{t-1}$, $\delta'_1 = \frac{|I'\cap[\![1,\alpha_{t-1}]\!]|}{\alpha_{t-1}}$ □

Theorem 3 now follows from successive applications of Lemma 6, as follows:

Proof of Theorem 3. Lemma 6 gives us, by induction on the size of the considered set I, that for all $I \subset [\![1, d]\!]$, we have:

$$\pi_I^{|I|} \geq \prod_{i<t} \left(\pi_{[\![\alpha_i+1,\alpha_{i+1}]\!]}^{|I_i|} \cdot \sqrt{d}^{\,|I_i|\log\delta_i} \right),$$

where $I_i = I \cap [\![\alpha_i + 1, \alpha_{i+1}]\!]$, and the integers t and α_i's, and the densities $\delta_i = \frac{|I_i|}{\alpha_{i+1}-\alpha_i}$ satisfy $t \geq 1$, $0 = \alpha_1 < \alpha_2 < \ldots < \alpha_t \leq d$ and $1 \geq \delta_1 > \ldots > \delta_{t-1} > 0$. By using Lemma 3 with $\Delta{:=}\sqrt{d}, k_i{:=}\,|I_i|$ and $d_i{:=}\alpha_{i+1} - \alpha_i$, we immediately obtain:

$$\pi_I^{|I|} \geq \left(\sqrt{d}^{\,|I|\log\frac{|I|}{\alpha_t-\alpha_1}} \right) \cdot \left(\prod_{i<t} \pi_{[\![\alpha_i+1,\alpha_{i+1}]\!]}^{|I_i|} \right).$$

For convenience, we define $\delta_t = 0$. Because of the definition of the α_i's, we have:

$$\begin{aligned}\prod_{i<t} \pi_{[\![\alpha_i+1,\alpha_{i+1}]\!]}^{|I_i|} &= \prod_{i<t} \left(\pi_{[\![\alpha_i+1,\alpha_{i+1}]\!]}^{\alpha_{i+1}-\alpha_i} \right)^{\delta_i} = \prod_{i<t} \prod_{i\leq j<t} \left(\pi_{[\![\alpha_i+1,\alpha_{i+1}]\!]}^{\alpha_{i+1}-\alpha_i} \right)^{\delta_j-\delta_{j+1}} \\ &= \prod_{j<t} \left(\prod_{i\leq j} \pi_{[\![\alpha_i+1,\alpha_{i+1}]\!]}^{\alpha_{i+1}-\alpha_i} \right)^{\delta_j-\delta_{j+1}} = \prod_{j<t} \left(\pi_{[\![1,\alpha_{j+1}]\!]}^{\alpha_{j+1}} \right)^{\delta_j-\delta_{j+1}}.\end{aligned}$$

By using $t - 1$ times Minkowski's theorem, we obtain that:

$$\begin{aligned}\pi_I^{|I|} &\geq \sqrt{d}^{\,|I|\log\frac{|I|}{d}} \cdot (\|\boldsymbol{b}_1\|/\sqrt{d})^{\sum_{j<t}\alpha_{j+1}(\delta_j-\delta_{j+1})} \\ &\geq \sqrt{d}^{\,|I|\log\frac{|I|}{d}} \cdot (\|\boldsymbol{b}_1\|/\sqrt{d})^{\sum_{j<t}(\alpha_{j+1}-\alpha_j)\delta_j} \\ &\geq \sqrt{d}^{\,|I|\left(\log\frac{|I|}{d}-1\right)} \cdot \|\boldsymbol{b}_1\|^{|I|}.\end{aligned}$$

The final inequality of the theorem is just the fact that $x \mapsto x\log(d/x)$ is maximal for $x = d/e$. □

Note that if $\max I < d$, we can apply the result to the HKZ-reduced basis $(b_1, \ldots, b_{\max I})$. In the case where $I = \{i\}$, we recover the result of [24] that

$$\|b_i^*\| \geq (\sqrt{i})^{-\log i - 1} \cdot \|b_1\|. \quad (2)$$

Still, our result is significantly better to what would have been obtained by combining several relations of the type of Equation (2), when $|I|$ grows large. For instance, for a worst case of our analysis where I is roughly the interval $[d(1-1/e), d]$, this strategy would yield a lower bound of the form $\|b_1\|^{d/e} \cdot \sqrt{d}^{(d/e)\log d}$, which is worse than Helfrich's analysis.

5 CVP and Other Related Problems

In this section, we describe what can be obtained by adapting our technique to the Closest Vector Problem and other problems related to strong lattice reduction. We only describe the proofs at a high level, since they are relatively straightforward.

In CVP, we are given a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ and a target vector $\boldsymbol{t}$, and we look for a lattice vector that is closest to $\boldsymbol{t}$. The first step of Kannan's CVP algorithm is to HKZ-reduce the $\boldsymbol{b}_i$'s. Then one adapts the enumeration algorithm of Figure 1 for CVP. For the sake of simplicity, we assume that $\|\boldsymbol{b}_1^*\|$ is the largest of the $\|\boldsymbol{b}_i^*\|$'s (we refer to Kannan's proof [16] for the general case). By using Babai's nearest hyperplane algorithm [6], we see that there is a lattice vector $\boldsymbol{b}$ at distance less than $\sqrt{d} \cdot \|\boldsymbol{b}_1\|$ of the target vector $\boldsymbol{t}$. As a consequence, if we take $A = d \cdot \|\boldsymbol{b}_1\|$ in the adaptation of the enumeration procedure, we are sure to find a solution. The analysis then reduces (at the level of Equation (1)) to bound the ratio $\frac{\|\boldsymbol{b}_1\|^d}{\prod_{i \leq d} \|\boldsymbol{b}_i^*\|}$, which can be done with Minkowski's theorem.

Theorem 4. *Given a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ and a target vector $\boldsymbol{t}$, all of them in $\mathbb{R}^n$ and with integer coordinates whose absolute values are smaller than some B, one can find all vectors in the lattice spanned by the $\boldsymbol{b}_i$'s that are closest to $\boldsymbol{t}$ in deterministic time $\mathcal{P}(\log B, n) \cdot d^{d/2+o(d)}$.*

The best deterministic complexity bound previously known for this problem was $\mathcal{P}(\log B, n) \cdot d^{d+o(d)}$ (see [13, 7]). Our result can also be adapted to enumerating all vectors of a lattice that are of length below a prescribed bound, which is in particular useful in the context of computing lattice theta series.

Another important consequence of our analysis is a significant worst-case bound improvement of Schnorr's block-based strategy [24] to compute relatively short vectors. More precisely, if we take the bounds given in [10] for the quality of Schnorr's semi-$2k$ reduction and for the transference reduction, we obtain the table of Figure 3. Each entry of the table gives the upper bound of the quantity $\frac{\|\boldsymbol{b}_1\|}{(\det L)^{1/d}}$ which is reachable for a computational effort of 2^t, for t growing to infinity. To sum up, the multiplicative exponent constant is divided by $e \approx 2.7$. The table upper bounds can be adapted to the quantity $\frac{\|\boldsymbol{b}_1\|}{\lambda_1(L)}$ by squaring them.

	Semi-$2k$ reduction	Transference reduction
Using Helfrich's complexity bound	$\lesssim 2^{\frac{\log 2}{2} \frac{d \log^2 t}{t}} \approx 2^{0.347 \frac{d \log^2 t}{t}}$	$\lesssim 2^{\frac{1}{4} \frac{d \log^2 t}{t}} \approx 2^{0.250 \frac{d \log^2 t}{t}}$
Using the improved complexity bound	$\lesssim 2^{\frac{\log 2}{2e} \frac{d \log^2 t}{t}} \approx 2^{0.128 \frac{d \log^2 t}{t}}$	$\lesssim 2^{\frac{1}{4e} \frac{d \log^2 t}{t}} \approx 2^{0.092 \frac{d \log^2 t}{t}}$

Fig. 3. Worst-case bounds for block-based reduction algorithms.

Let us finish by mentioning that work under progress seems to show, by using a technique due to Ajtai [3], that our analyses are sharp, in the sense that for all $\varepsilon > 0$, we can build HKZ-reduced bases for which the number of steps of Kannan's algorithm would be of the order of $d^{d(\frac{1}{2e}-\varepsilon)}$.

References

1. E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, 48(8):2201–2214, 2002.
2. M. Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of the 30th Symposium on the Theory of Computing (STOC 1998)*, pages 284–293. ACM Press, 1998.
3. M. Ajtai. The worst-case behavior of Schnorr's algorithm approximating the shortest nonzero vector in a lattice. In *Proceedings of the 35th Symposium on the Theory of Computing (STOC 2003)*, pages 396–406. ACM Press, 2003.
4. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the 29th Symposium on the Theory of Computing (STOC 1997)*, pages 284–293. ACM Press, 1997.
5. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the 33rd Symposium on the Theory of Computing (STOC 2001)*, pages 601–610. ACM Press, 2001.
6. L. Babai. On Lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.
7. J. Bömer. Closest vectors, successive minima and dual-HKZ bases of lattices. In *Proceedings of the 2000 International Colloquium on Automata, Languages and Programming (ICALP 2000)*, volume 1853 of *Lecture Notes in Computer Science*, pages 248–259. Springer-Verlag, 2000.
8. D. Coppersmith and A. Shamir. Lattice attacks on NTRU. In *Proceedings of Eurocrypt 1997*, volume 1233 of *Lecture Notes in Computer Science*, pages 52–61. Springer-Verlag, 1997.
9. U. Fincke and M. Pohst. A procedure for determining algebraic integers of given norm. In *Proceedings of EUROCAL*, volume 162 of *Lecture Notes in Computer Science*, pages 194–202, 1983.
10. N. Gama, N. Howgrave-Graham, H. Koy, and P. Nguyen. Rankin's constant and blockwise lattice reduction. In *Proceedings of Crypto 2006*, number 4117 in Lecture Notes in Computer Science, pages 112–130. Springer-Verlag, 2006.
11. O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *Proceedings of Crypto 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 112–131. Springer-Verlag, 1997.
12. I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. Submitted.
13. B. Helfrich. Algorithms to construct Minkowski reduced and Hermite reduced lattice bases. *Theoretical Computer Science*, 41:125–139, 1985.
14. C. Hermite. Extraits de lettres de M. Hermite à M. Jacobi sur différents objets de la théorie des nombres, deuxième lettre. *Journal für die reine und angewandte Mathematik*, 40:279–290, 1850.
15. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: a ring based public key cryptosystem. In *Proceedings of the 3rd Algorithmic Number Theory Symposium (ANTS III)*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer-Verlag, 1998.
16. R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the 15th Symposium on the Theory of Computing (STOC 1983)*, pages 99–108. ACM Press, 1983.
17. J. C. Lagarias, W. H. Lenstra, and C. P. Schnorr. Korkine-Zolotarev bases and successive minimal of a lattice and its reciprocal lattice. *Combinatorica*, 10:333–348, 1990.
18. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:513–534, 1982.
19. J. Martinet. *Perfect Lattices in Euclidean Spaces.* Springer-Verlag, 2002.
20. J. Mazo and A. Odlyzko. Lattice points in high-dimensional spheres, 1990.
21. D. Micciancio and S. Goldwasser. *Complexity of lattice problems: a cryptographic perspective.* Kluwer Academic Press, 2002.
22. H. Minkowski. *Geometrie der Zahlen.* Teubner-Verlag, 1896.
23. O. Regev. Lecture notes of *lattices in computer science*, taught at the Computer Science Tel Aviv University. Available at `http://www.cs.tau.il/~odedr`.
24. C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
25. C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematics of Programming*, 66:181–199, 1994.
26. V. Shoup. NTL, Number Theory C++ Library. Available at `http://www.shoup.net/ntl/`.

Proof of Lemma 2

We prove the result by induction on k.For $k = 1$, the bound easily follows from $\gamma_d \leq (d+4)/4$. Suppose now that the result holds for some $k \in [\![1, d-2]\!]$, and that we want to show that it holds for $k+1$. Notice that we can suppose that $d \geq 3$. Define $G_d(k) = \frac{1}{2}\log d \log \frac{d}{d-k}$. Then for any $\lambda > 0$,

$$G_d(k+\lambda) - G_d(k) = -\frac{1}{2}\log d \log \frac{d-k-\lambda}{d-k} \geq \frac{1}{2}\frac{\lambda \log d}{d-k}.$$

Taking $\lambda = 1$, we see that $G_d(k+1) - G_d(k) \geq \frac{1}{2}\frac{\log d}{d-k}$.
From the upper bound $\gamma_d \leq (d+4)/4$, we obtain:

$$\log \mathit{\Gamma}_d(k+1) - \log \mathit{\Gamma}_d(k) = \frac{1}{2}\frac{\log \gamma_{d-k}}{d-k-1} \leq \frac{1}{2}\frac{\log(d-k+4)/4}{d-k-1}.$$

Now, since the sequence $\left(\frac{n\log((n+4)/4)}{n-1}\right)_{n\geq 2}$ is increasing, we have:

$$\begin{aligned}\frac{(d-k)\log((d-k+4)/4)}{d-k-1} &\leq \frac{d-1}{d-2}\log((d+3)/4)\\ &= \log d + \frac{(d-1)\log((d+3)/4) - (d-2)\log d}{d-2}\\ &\leq \log d,\end{aligned}$$

since the last term is a decreasing function of d, which is negative for $d = 3$. □

Proof of Lemma 3

We have $-\log\prod_{i\leq t}\delta^{-k_i \log\frac{k_i}{d_i}} = (\log\delta)\cdot\sum_{i\leq t}k_i\log\frac{k_i}{d_i}$. Now, note that the function $x \mapsto x\log x$ is convex on $[0, +\infty)$. This means that for any $t \geq 1$, for any $a_1, \ldots, a_t > 0$, and for any $\lambda_1, \ldots, \lambda_t \in [0,1]$ such that $\sum_{i\leq t}\lambda_i = 1$, we have:

$$\sum_{i\leq t}\lambda_i a_i \log a_i \geq \left(\sum_{i\leq t}\lambda_i a_i\right)\log\left(\sum_{i\leq t}\lambda_i a_i\right).$$

In particular, for $\lambda_i{:=}\frac{d_i}{\sum_{i\leq t}d_i}$ and $a_i{:=}\frac{k_i}{d_i}$, we get (after multiplication by $\sum_{i\leq t}d_i$):

$$-\log\prod_{i\leq t}\delta^{-k_i\log\frac{k_i}{d_i}} \geq (\log\delta)\cdot\left(\sum_{i\leq t}k_i\right)\log\left(\frac{\sum_{i\leq t}k_i}{\sum_{i\leq t}d_i}\right),$$

which is exactly $-\log\delta^{-\left(\sum_{i\leq t}k_i\right)\log\frac{\sum_{i\leq t}k_i}{\sum_{i\leq t}d_i}}$. □

Proof of Lemma 4.

Proof. We start with the first identity. We prove it by induction on k. For $k = 1$, this is Minkowski's bound. Assume it to be true for a given $k \leq d-2$. We are to prove that it holds for $k+1$ instead of k. By applying Minkowski's bound to the $(d-k)$-dimensional HKZ-reduced basis $\boldsymbol{b}_{k+1}^*, \ldots, \boldsymbol{b}_d^*$, we have:

$$\|\boldsymbol{b}_{k+1}^*\| \leq \sqrt{\gamma_{d-k}}^{\frac{d-k}{d-k-1}} \cdot \pi_{[\![k+2,d]\!]}. \tag{3}$$

We can rewrite our induction hypothesis as

$$\pi_{[\![1,k+1]\!]}^{\frac{k+1}{k}} \cdot \|\boldsymbol{b}_{k+1}^*\|^{-\frac{1}{k}} \leq (\Gamma_d(k))^{\frac{d}{k}} \cdot \pi_{[\![k+2,d]\!]}^{\frac{d-k-1}{d-k}} \cdot \|\boldsymbol{b}_{k+1}^*\|^{\frac{1}{d-k}},$$

or, again, as

$$\pi_{[\![1,k+1]\!]}^{\frac{k+1}{k}} \leq (\Gamma_d(k))^{\frac{d}{k}} \cdot \pi_{[\![k+2,d]\!]}^{\frac{d-k-1}{d-k}} \cdot \|\boldsymbol{b}_{k+1}^*\|^{\frac{d}{k(d-k)}}.$$

This gives, by using Equation (3):

$$\pi_{[\![1,k+1]\!]}^{\frac{k+1}{k}} \leq (\Gamma_d(k))^{\frac{d}{k}} \cdot \sqrt{\gamma_{d-k}}^{\frac{d}{k(d-k-1)}} \cdot \pi_{[\![k+2,d]\!]}^{\frac{k+1}{k}} = (\Gamma_d(k+1))^{\frac{d}{k}} \cdot \pi_{[\![k+2,d]\!]}^{(k+1)/k}.$$

By raising this last identity to the power $\frac{k}{k+1}$, we get

$$\pi_{[\![1,k+1]\!]} \leq (\Gamma_d(k+1))^{\frac{d}{k+1}} \cdot \pi_{[\![k+2,d]\!]},$$

which, by induction, yields the first inequality.

The second inequality follows easily from the first one. Indeed, it suffices to raise the first one to the power k/d, multiply both sides by $\left(\pi_{[\![k+1,d]\!]}\right)^{(d-k)/d}$, and use the identity $\det L = \left(\pi_{[\![1,k]\!]}\right)^k \cdot \left(\pi_{[\![k+1,d]\!]}\right)^{d-k}$.

Proof of Lemma 5.

First notice that, as a consequence of Lemma 4, we have, for k, l integers, $1 \leq k \leq l < d$,

$$\pi_{[\![l+1,d]\!]} \geq \Gamma_{d-k}(l-k)^{-1} \cdot \pi_{[\![k+1,d]\!]}. \tag{4}$$

Recall that:

$$\tilde{\pi}_{[x_1,d]} = \left(\pi_{[\![\lfloor x_1 \rfloor,d]\!]}\right)^{\lambda_1} \cdot \left(\pi_{[\![\lfloor x_1 \rfloor+1,d]\!]}\right)^{1-\lambda_1} \quad \text{and} \quad \tilde{\pi}_{[x_2,d]} = \left(\pi_{[\![\lfloor x_2 \rfloor,d]\!]}\right)^{\lambda_2} \cdot \left(\pi_{[\![\lfloor x_2 \rfloor+1,d]\!]}\right)^{1-\lambda_2},$$

with $\lambda_i = \frac{(d-\lfloor x_i \rfloor+1)(1-x_i+\lfloor x_i \rfloor)}{d-x_i+1}$ for $i \in \{1,2\}$. Notice that since $x_1 < x_2$, either $\lfloor x_1 \rfloor + 1 \leq \lfloor x_2 \rfloor$, or $\lfloor x_1 \rfloor = \lfloor x_2 \rfloor$. In the last case, since the function $x \mapsto (u-x)/(v-x)$ is decreasing when $u < v$ and for $x < u$, we must have $\lambda_2 < \lambda_1$.

We split the proof in several cases, depending on the respective values of λ_1 and λ_2.

First case: $\lambda_1 \leq \lambda_2$. In that case, we have $\lfloor x_1 \rfloor + 1 \leq \lfloor x_2 \rfloor$. We define

$$G := \Gamma_{d-\lfloor x_1 \rfloor+1}(\lfloor x_2 \rfloor - \lfloor x_1 \rfloor)^{\lambda_1} \cdot \Gamma_{d-\lfloor x_1 \rfloor}(\lfloor x_2 \rfloor - \lfloor x_1 \rfloor - 1)^{\lambda_2-\lambda_1} \cdot \Gamma_{d-\lfloor x_1 \rfloor}(\lfloor x_2 \rfloor - \lfloor x_1 \rfloor)^{1-\lambda_2}.$$

By using three times Equation (4), we get:

$$\begin{aligned}\tilde{\pi}_{[x_2,d]} &= \left(\pi_{[\![\lfloor x_2\rfloor,d]\!]}\right)^{\lambda_2}\cdot\left(\pi_{[\![\lfloor x_2\rfloor+1,d]\!]}\right)^{1-\lambda_2}\\ &\geq \left(\pi_{[\![\lfloor x_2\rfloor,d]\!]}\right)^{\lambda_1}\cdot\left(\pi_{[\![\lfloor x_2\rfloor,d]\!]}\right)^{\lambda_2-\lambda_1}\cdot\left(\pi_{[\![\lfloor x_2\rfloor+1,d]\!]}\right)^{1-\lambda_2}\\ &\geq G^{-1}\cdot\left(\pi_{[\![\lfloor x_1\rfloor,d]\!]}\right)^{\lambda_1}\cdot\left(\pi_{[\![\lfloor x_1\rfloor+1,d]\!]}\right)^{1-\lambda_1}.\end{aligned}$$

Now, Lemma 4 gives that

$$\frac{\log G}{\log\sqrt{d}} \leq \lambda_1\log\frac{d-\lfloor x_1\rfloor+1}{d-\lfloor x_2\rfloor+1} + (\lambda_2-\lambda_1)\log\frac{d-\lfloor x_1\rfloor}{d-\lfloor x_2\rfloor+1} + (1-\lambda_2)\log\frac{d-\lfloor x_1\rfloor}{d-\lfloor x_2\rfloor},$$

which, by concavity of the function $x\mapsto\log x$, is at most the logarithm of

$$E(x_1,x_2) := \lambda_1\frac{d-\lfloor x_1\rfloor+1}{d-\lfloor x_2\rfloor+1} + (\lambda_2-\lambda_1)\frac{d-\lfloor x_1\rfloor}{d-\lfloor x_2\rfloor+1} + (1-\lambda_2)\frac{d-\lfloor x_1\rfloor}{d-\lfloor x_2\rfloor}$$

To complete the proof of this first case, it suffices to prove that $E(x_1,x_2)\leq fracd-x_1d-x_2$. We have

$$\begin{aligned}E(x_1,x_2) &= \frac{\lambda_1}{d-\lfloor x_2\rfloor+1} + \frac{d-\lfloor x_1\rfloor}{d-x_2+1}\\ &= \frac{d-x_1}{d-x_2} + \frac{\lambda_1}{d-\lfloor x_2\rfloor+1} - \frac{1-\mathrm{frac}(x_1)}{d-x_2+1} - \frac{x_2-x_1}{(d-x_2)(d-x_2+1)},\\ &\leq \frac{d-x_1}{d-x_2} + \frac{1}{d-x_2+1}\left(\lambda_1-(1-\mathrm{frac}(x_1)) - \frac{x_2-x_1}{d-x_2}\right)\\ &= \frac{d-x_1}{d-x_2} + \frac{1}{d-x_2+1}\left(\frac{(1-\mathrm{frac}(x_1))\mathrm{frac}(x_1)}{d-x_1+1} - \frac{x_2-x_1}{d-x_2}\right)\\ &\leq \frac{d-x_1}{d-x_2} + \frac{1}{d-x_2+1}\left(\frac{1-\mathrm{frac}(x_1)}{d-x_2} - \frac{x_2-x_1}{d-x_2}\right),\end{aligned}$$

from which the result follows at once, since $\lfloor x_1\rfloor < \lfloor x_2\rfloor$ implies that $x_2-x_1 = \lfloor x_2\rfloor - \lfloor x_1\rfloor + \mathrm{frac}(x_2)-\mathrm{frac}(x_1) \geq 1-\mathrm{frac}(x_1)$.

Second case: $\lambda_1 > \lambda_2$. Similarly, defining

$$H = \mathit{\Gamma}_{d-\lfloor x_1\rfloor+1}(\lfloor x_2\rfloor-\lfloor x_1\rfloor)^{\lambda_2}\cdot\mathit{\Gamma}_{d-\lfloor x_1\rfloor+1}(\lfloor x_2\rfloor-\lfloor x_1\rfloor+1)^{\lambda_1-\lambda_2}\cdot\mathit{\Gamma}_{d-\lfloor x_1\rfloor}(\lfloor x_2\rfloor-\lfloor x_1\rfloor)^{1-\lambda_1},$$

we obtain

$$\begin{aligned}\tilde{\pi}_{[x_2,d]} &= \left(\pi_{[\![\lfloor x_2\rfloor,d]\!]}\right)^{\lambda_2}\cdot\left(\pi_{[\![\lfloor x_2\rfloor+1,d]\!]}\right)^{1-\lambda_2}\\ &= \left(\pi_{[\![\lfloor x_2\rfloor,d]\!]}\right)^{\lambda_2}\left(\pi_{[\![\lfloor x_2\rfloor+1,d]\!]}\right)^{\lambda_1-\lambda_2}\left(\pi_{[\![\lfloor x_2\rfloor+1,d]\!]}\right)^{1-\lambda_1}\\ &\geq H^{-1}\left(pi_{[\![\lfloor x_1\rfloor,d]\!]}\right)^{\lambda_1}\left(\pi_{[\![\lfloor x_1\rfloor+1,d]\!]}\right)^{1-\lambda_1}.\end{aligned}$$

Lemma 4 gives us that:

$$\frac{\log H}{\log\sqrt{d}} \leq \lambda_2\log\frac{d-\lfloor x_1\rfloor+1}{d-\lfloor x_2\rfloor+1} + (\lambda_1-\lambda_2)\log\frac{d-\lfloor x_1\rfloor+1}{d-\lfloor x_2\rfloor} + (1-\lambda_1)\log\frac{d-\lfloor x_1\rfloor}{d-\lfloor x_2\rfloor}.$$

By concavity of the function $x \mapsto \log x$, the right hand side is at most the logarithm of

$$\lambda_2 \frac{d - \lfloor x_1 \rfloor + 1}{d - \lfloor x_2 \rfloor + 1} + (\lambda_1 - \lambda_2) \frac{d - \lfloor x_1 \rfloor + 1}{d - \lfloor x_2 \rfloor} + (1 - \lambda_1) \frac{d - \lfloor x_1 \rfloor}{d - \lfloor x_2 \rfloor}$$
$$= E(x_1, x_2) + \frac{\lambda_1 - \lambda_2}{(d - \lfloor x_2 \rfloor)(d - \lfloor x_2 \rfloor + 1)}.$$

Hence, we just need to prove that:

$$E'(x_1, x_2) := E(x_1, x_2) + \frac{(\lambda_1 - \lambda_2)}{(d - \lfloor x_2 \rfloor)(d - \lfloor x_2 \rfloor + 1)} \leq \frac{d - x_1}{d - x_2}.$$

Some elementary calculus provides the equalities:

$$\begin{aligned} E'(x_1, x_2) &= \frac{d - x_1}{d - x_2} + \frac{\lambda_1}{d - \lfloor x_2 \rfloor} - \frac{1 - \mathrm{frac}(x_2)}{(d - \lfloor x_2 \rfloor)(d - x_2 + 1)} - \frac{1 - \mathrm{frac}(x_1)}{d - x_2 + 1} - \frac{x_2 - x_1}{(d - x_2)(d - x_2 + 1)} \\ &= \frac{d - x_1}{d - x_2} + \frac{\lambda_1}{d - \lfloor x_2 \rfloor} - \frac{1 - \mathrm{frac}(x_1)}{d - x_2} - \frac{1 - \mathrm{frac}(x_2)}{(d - \lfloor x_2 \rfloor)(d - x_2 + 1)} - \frac{x_2 - \lfloor x_1 \rfloor - 1}{(d - x_2)(d - x_2 + 1)} \end{aligned}$$

Second case, first sub-case: $\lambda_1 > \lambda_2$, $\lfloor x_1 \rfloor < \lfloor x_2 \rfloor$. In that case,

$$\begin{aligned} E'(x_1, x_2) - \frac{d - x_1}{d - x_2} &\leq \frac{\lambda_1 - (1 - \mathrm{frac}(x_1))}{d - x_2} - \frac{1 - \mathrm{frac}(x_2)}{(d - \lfloor x_2 \rfloor)(d - x_2 + 1)} - \frac{1}{(d - x_2)(d - x_2 + 1)} \\ &\leq \frac{1 - \mathrm{frac}(x_1)}{(d - x_2)(d - x_1 + 1)} - \frac{1}{(d - x_2)(d - x_2 + 1)} \\ &\leq 0 \end{aligned}$$

Second case, second sub-case: $\lambda_1 > \lambda_2$, $\lfloor x_1 \rfloor = \lfloor x_2 \rfloor$. In that case, after some rewriting which can be checked with one's favourite computer algebra system, one finds that:

$$\begin{aligned} E'(x_1, x_2) - \frac{d - x_1}{d - x_2} &= \frac{1}{(d - \lfloor x_1 \rfloor)(d - x_2)} \left(\frac{(1 - \mathrm{frac}(x_1))(x_1 - x_2)(d - \lfloor x_2 \rfloor)}{d - x_1 + 1} - \frac{\mathrm{frac}(x_2)(\lambda_1 - \lambda_2)}{d - \lfloor x_1 \rfloor + 1} \right) \\ &\leq 0. \end{aligned}$$

□