A COMPLETE WORST-CASE ANALYSIS OF KANNAN'S SHORTEST LATTICE VECTOR ALGORITHM

GUILLAUME HANROT* AND DAMIEN STEHLÉ†

Abstract. Computing a shortest nonzero vector of a given euclidean lattice and computing a closest lattice vector to a given target are pervasive problems in computer science, computational mathematics and communication theory. The classical algorithms for these tasks were invented by Ravi Kannan in 1983 and, though remarkably simple to establish, their complexity bounds have not been improved for almost thirty years. In the present paper, we provide a complete worst-case analysis of Kannan's algorithm for the shortest vector problem. We obtain a new worst-case complexity upper bound, as well as the first worst-case complexity lower bound, both of the order of $2^{O(d)} \cdot d^{\frac{d}{2e}}$ (up to polynomial factors) bit operations, where d is the rank of the lattice. The lower bound is obtained by the construction of a probabilistic algorithm that returns lattice bases on which Kannan's algorithm requires at least that many operations. We also provide a new complexity upper bound for Kannan's closest vector algorithm, of the order of $2^{O(d)} \cdot d^{\frac{d}{2}}$. To obtain these complexity results, we prove new bounds on the geometry of lattice bases reduced in the sense of Hermite-Korkine-Zolotarev, which may be of independent interest.

Key words. lattice reduction, shortest vector problem, closest vector problem, complexity analysis

AMS subject classifications. 11Y16, 68Q25, 68W40, 11P21

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 $\mathbf{b}_1, \ldots, \mathbf{b}_d$. If these vectors are linearly independent, we say that they are a basis of L, and the integer d is called the rank. The discreteness of L implies the existence of a shortest non-zero lattice vector. Its norm is referred to as the lattice minimum and is denoted by $\lambda(L)$. Similarly, for any given target vector $\mathbf{t}$ in $\mathbb{R}^n$, there exists a lattice vector closest to $\mathbf{t}$. Making the latter effective leads to the most famous computational problems associated with lattices:

SVP— The Shortest Vector Problem is as follows: given a basis of a lattice L, find a shortest non-zero vector of L.

CVP— The Closest Vector Problem is as follows: given a basis of a lattice L and a target vector, find an element of L closest to the target vector.

The decisional variant of CVP (which consists in deciding whether there is a lattice vector within a prescribed distance from the target vector) was proved NP hard by van Emde Boas [23] in 1981. Ajtai later established the NP hardness of SVP under randomized reductions [3]. Later works showed that many relaxations and variants of CVP and SVP also remain NP hard [22, 47, 48, 20, 21, 24, 39, 63, 33].

Fields of application. SVP and CVP are of prime importance in cryptography. They have been the cornerstone of the downfall of knapsack cryptosystems [46, 57, 55], which were an early days alternative to RSA [64]. Their cryptographic relevance was revived by Ajtai and Dwork [6] who proposed a public-key encryption scheme which is provably as hard to break on average as to solve the worst-case instances of a variant of SVP. Other more practical lattice-based cryptosystems were proposed at the same time, but with weaker security guarantees [29, 36]. They paved the way

*École Normale Supérieure de Lyon

†CNRS/University of Sydney/Macquarie University, partly supported by the ARC Discovery Grant DP0880724 "Integral lattices and their theta series"

to the proposals of many different lattice-based cryptographic schemes. We refer the interested reader to the recent survey [50]. For many of them, the best attack known consists in solving instances of close variants of SVP and CVP. It is therefore highly important to precisely assess which complexity is achievable.

Communication theory is another very active field of research where SVP and CVP play a central role. In the linear Multi-Input Multi-Output (MIMO) channel model, a data vector $\mathbf{x} \in \mathbb{Z}^n$ is transformed into a vector $\mathbf{y} = B \cdot \mathbf{x} + \mathbf{e} \in \mathbb{R}^n$, where the channel matrix B is known, and the perturbation $\mathbf{e}$ is unknown. The receiver has to retrieve the data $\mathbf{x}$ from the vector $\mathbf{y}$, which is a general instanciation of CVP. Applying strong pre-processing to B, which essentially consists in solving several SVP instances related to the channel matrix, can help speeding-up the decoding process. We refer to [54, 69, 2, 43] for more details. SVP and CVP also arise in GPS communications [32].

There are many other application domains of SVP and CVP, including discrete optimization, e.g., integer linear programming [42, 38, 1], algorithmic number theory, e.g., to compute the invariants of a number field [17], and combinatorics, e.g., to find t-designs [70].

Known algorithms. Three main categories of algorithms are known for solving SVP and CVP. The first one, which contains deterministic algorithms, relies on the exhaustive enumeration of lattice points within small convex sets. The latter is known as the Fincke-Pohst enumeration algorithm [25] in the algorithmic number theory community, whereas in computer science, it is attributed to Kannan [37] (the CVP variant is also known as sphere decoding in communications). There are two main differences between them: first, in Kannan's algorithm, a long pre-computation is performed on the basis before starting the enumeration process; second, Kannan enumerates points in a hyper-parallelepiped whereas Fincke and Pohst consider a hyper-ellipsoid contained in Kannan's hyper-parallelepiped — though it may be that Kannan chose the hyper-parallelepiped in order to simplify his complexity analysis. At first sight, Kannan's algorithm seems slower than the one of Fincke and Pohst, but it is actually the opposite: the lengthy pre-computation decreases the cost of the hugely expensive enumeration. Kannan obtained a $d^{d+o(d)}$ complexity bound,[1] for both SVP and CVP. Note that the space requirement is polynomial, contratily to the algorithms from the two other categories. In 1985, Helfrich [34] refined Kannan's algorithm (one of the algorithmic improvements is actually attributed to Schnorr) and his analysis, to finally obtain a $d^{\frac{d}{2}+o(d)}$ complexity bound for SVP (the complexity of the CVP algorithm remaining $d^{d+o(d)}$). More recently, Blömer [11] proposed another enumeration-based CVP solver whose complexity is a factor $2^{O(d)}$ less than Helfrich's.

A second type of algorithms was discovered by Ajtai, Kumar and Sivakumar [7]: they introduced in 2001 a probabilistic (Monte Carlo) algorithm for SVP, of complexity $2^{O(d)}$. The best known exponent constant was progressively decreased [62, 56, 52] and is now slightly less than 2.5 (see [61]). The Ajtai-Kumar-Sivakumar algorithm was adapted to CVP in [8], but the CVP adaptation only finds a lattice vector whose distance to the input target is within a factor $1+\epsilon$ from optimal (for any fixed $\epsilon > 0$). The exponent constant in the $2^{O(d)}$ complexity grows to infinity as ϵ tends to 0. The latter algorithm was recently adapted for other lattice problems by Blömer and Naewe [12]. Apart from the possibility of incorrect or non-optimal outputs, these

[1] In all the complexity bounds mentioned in the introduction, we omit an implicit multiplicative factor that is polynomial in the bit-size of the input.

algorithms also have the drawback that they require an exponential amount of space.

Micciancio and Voulgaris [51] very recently introduced yet another family of algorithms for CVP and SVP. The time and space complexities are $2^{O(d)}$ like the Ajtai et al algorithms, but they are deterministic and allow CVP to be solved exactly. Asymptotically, this family seems to supersede the one above, but due to the different space requirements it remains incomparable with the Kannan algorithm.

In practice, the proved and heuristic variants of Kannan's algorithms (see [67, 2, 28]) respectively outperform proved and heuristic variants of the Ajtai et al algorithm (the article [52] contains a description of the currently fastest implementation of a heuristic variant of [7]).

Our contributions. Our main results are to lower the best worst-case complexity upper bound for Kannan's SVP algorithm, from $d^{\frac{d}{2}+o(d)} \approx d^{0.5 \cdot d}$ to $2^{O(d)} \cdot d^{\frac{d}{2e}} \lesssim d^{0.184 \cdot d}$ and to show the existence of inputs for which this bound is essentially reached. This means that the worst-case complexity of Kannan's algorithm is exactly $2^{O(d)} \cdot d^{\frac{d}{2e}}$. We prove the lower bound by exhibiting bases reduced in the Hermite-Korkine-Zolotarev sense (HKZ-reduced for short), which are least reduced possible. This makes them good corner cases for strong lattice reductions. We show the strengthened upper bound by studying the Gram-Schmidt orthogonalisation of HKZ-reduced bases. Finally, we also decrease the best worst-case complexity upper bound for Kannan's CVP algorithm, from $d^{d+o(d)}$ to $2^{O(d)} \cdot d^{\frac{d}{2}}$.

It must be noted that if one follows our analysis step by step, the derived $O(d)$ in the complexity upper bounds may be large when evaluated for some practical d: the constants hidden in the "$O(d)$" may be improved (for some of them it may be easy, for others it is probably much harder). No effort was made in that direction, and we believe that it would have complicated the proof with irrelevant details. In fact, most of our analysis consists in estimating the number of integer points within hyper-ellipsoids, and showing that the approximation by the volume is valid. By replacing this discretization by heuristic volume estimates, one obtains very small heuristic hidden constants.

In his analysis, Kannan [37] bounds the number of integer points in a hyper-ellipsoid by considering the circumbscribed parallelepiped. Our improvement stems from the well-known fact that the latter is much larger than the former: when the dimension increases, the ratio of the two volumes shrinks to 0 very quickly. This had already been experimentally observed by Fincke and Pohst [25], but a theoretical analysis was yet to be obtained. We first relate the number of integer points within ellipsoids to their volumes, and then bound the latter volumes for the situation where the input basis is HKZ-reduced. Some parts of our proof could be of independent interest. For example, we show that for any HKZ-reduced lattice basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$, and any subset I of $\{1, \ldots, d\}$ of cardinality $|I|$, we have:

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

where $(\mathbf{b}_i^*)_{i \leq d}$ is the Gram-Schmidt orthogonalisation of the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$. This inequality generalises the results of [65] on the quality of HKZ-reduced bases.

Studying the tightness of the analysis described above leads to considering HKZ-reduced lattice bases of poorest quality. We prove the existence of such bases (up to lower order factors) by building upon and simplifying a technique introduced by Ajtai in [4, 5] to prove lower bounds on quantities related to Schnorr's hierarchies of

reductions and reduction algorithms [65]. To do so, Ajtai also builds HKZ-reduced bases of bad quality. It could be expected that the function $\log \frac{\|\mathbf{b}_i^*\|}{\|\mathbf{b}_1\|}$ may be linear with respect to i, for such worst-case bases — see [66, 4, 5]. The quality of our bases is even worse, and cannot be reached with a linear function $\log \frac{\|\mathbf{b}_i^*\|}{\|\mathbf{b}_1\|}$: the worst-case HKZ-reduced bases satisfy $\log \frac{\|\mathbf{b}_i^*\|}{\|\mathbf{b}_1\|} \approx \frac{1}{4}\log^2(d-i)$ (see Corollary 5.5), which is a concave function of i. The concavity originates from the greediness of the HKZ-reduction: the ith vector is a shortest non-zero vector in a $(d-i+1)$-dimensional lattice, which is a much stronger requirement for small i than for large i. We prove that given such bases as input, Kannan's SVP algorithm performs at least $2^{O(d)} \cdot d^{\frac{d}{2e}}$ operations.

Practical Implications. Our work was initially motivated by practical applications in cryptanalysis. First of all, our lowered complexity upper bound may explain why Kannan's algorithm remains tractable even in moderate dimensions (higher than 60). Moreover, as mentioned above, our analysis can be interpreted as a formalisation of a heuristic cost estimate based on volume computations. In [30], we precisely describe these estimates and give practical evidence that the practical running times match. They have been implemented in the MAGMA system [14] where they allow the user to estimate the cost of an enumeration before actually running it. These estimates can also be used to efficiently parallelize the enumeration [59] and to provide cost gain and failure probability guesses for the pruned enumeration heuristic of [67] (see [28] for more explanations and further developments in that direction).

When the dimension becomes too large for Kannan's SVP algorithm to terminate in a reasonable amount of time, one uses Schnorr's block-based algorithms [65, 67] (see [26, 27] for state-of-the-art block-based algorithms). These algorithms use either Kannan's algorithm, or the underlying lattice point enumeration procedure. This dominates their running-times. Our complexity improvement on Kannan's SVP algorithm automatically ensures better worst-case efficiency/quality trade-offs for these block-based algorithms.

Our lower-bound analysis can readily be adapted to provide bases that are corner cases for Schnorr's block-based algorithms [65, 67] (we refer to [31] for more details), which could be used to devise experimental corner cases.

Related works. The present article contains the full details of the parts of [30] and [31] that are relevant to Kannan's algorithms. The practical aspects of these earlier works have not been included in the present article, to focus on the theoretical improvements on SVP and CVP. In [60], Pujol and Stehlé investigate the use of floating-point arithmetic within the SVP enumeration algorithms. Their result allows the arithmetic cost of the enumeration (the number of arithmetic steps is unchanged) to be decreased. In [59], Pujol shows how to efficiently parallelize the enumeration algorithms, using the heuristic volume estimates mentioned above. In [28], these volume estimates are used to (heuristically) analyze a modification of the pruning strategy of [67]. These works already led to several implementations, in the Magma computer algebra system [14], and in the stand-alone library `fplll` [16].

Road-Map. In §2, we give some reminders on lattices and on Kannan's algorithms. We then study the underlying enumeration procedure in §3. §4 consists in proving geometrical properties satisfied by HKZ-reduced bases, which is the key for the complexity upper bound to Kannan's SVP algorithm. In §5, we describe the probabilistic sampling of lattice bases satisfying some geometrical properties: this leads to the lower bound on the worst-case complexity of Kannan's SVP algorithm. Finally, in §6,

we draw a list of related open problems.

NOTATION. All logarithms are natural logarithms, i.e., $\log(\mathrm{e}) = 1$. Let $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ be the Euclidean norm and inner product of $\mathbb{R}^n$. Bold variables correspond to vectors. For complexity statements, we use the bit complexity model. The notation $\mathcal{P}(n_1, \ldots, n_i)$ means $O(n_1 \cdot \ldots \cdot n_i)^c$ 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 $x - \lfloor x \rfloor$. If $x \in \mathbb{R}$, then $(x)_+$ denotes $\max(0, x)$. Finally, for any integers a and b, we define $[\![a, b]\!]$ as $[a, b] \cap \mathbb{Z}$.

2. Reminders. We assume the reader is familiar with the algorithmic aspects of the geometry of numbers, and refer to [49] and [62] for introductory exposures.

2.1. Euclidean Lattices. Let $\mathbf{b}_1, \ldots, \mathbf{b}_d$ be (possibly linearly dependent) vectors of $\mathbb{R}^n$. Their *Gram-Schmidt orthogonalisation* (GSO) $\mathbf{b}_1^*, \ldots, \mathbf{b}_d^*$ is the orthogonal family defined as follows: for any i, the vector $\mathbf{b}_i^*$ is the projection of the vector $\mathbf{b}_i$ orthogonally to the linear span of the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$. If the $\mathbf{b}_i$'s are linearly independent, then we have $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*$ where $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\|\mathbf{b}_j^*\|^2}$, for any $j \leq i$. In the case of linearly dependent vectors, the same formula holds if we let $\mu_{i,j}$ be 0 for any $i > j$ such that $\mathbf{b}_j^* = \mathbf{0}$. Notice that the GSO family depends on the order of the vectors $\mathbf{b}_i$. If the $\mathbf{b}_i$'s are rational, then the $\mathbf{b}_i^*$'s and the $\mu_{i,j}$'s are rational, and their bit-size is bounded by a polynomial function of the dimensions and the bit-size of the $\mathbf{b}_i$'s.

Any lattice L with $d \geq 2$ has infinitely many bases, related to one another by unimodular transforms (i.e., elements of $GL_d(\mathbb{Z})$). Some quantities related to L do not depend on the particular choice of basis of L: these are called lattice invariants. For example, the rank d and the minimum $\lambda(L)$ are lattice invariants. The *determinant* of L is another one. It is defined as $\det L = \prod_{i=1}^{d} \|\mathbf{b}_i^*\|$, where $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is any basis of L and can be interpreted as the geometric volume of the parallelepiped spanned by the basis vectors.

The volume and the minimum of a lattice cannot vary completely independently. Hermite [35] was the first to bound the ratio $\frac{\lambda(L)}{(\det L)^{1/d}}$ as a function of the rank only, but his bound was later greatly improved by Minkowski [53]. The *Hermite constant* γ_d is defined as the supremum of the ratio $\frac{\lambda(L)^2}{(\det L)^{2/d}}$ over lattices L of rank d. In particular, we have the bound $\gamma_d \leq \frac{d+4}{4}$ (see [44, Remark 2.7.5]), which we will refer to as *Minkowski's theorem*.

2.2. Lattice Reduction. Unfortunately, there is no known constructive proof of Minkowski's theorem. None provides any insight on how to find a shortest non-zero vector from a given basis. In practice, one often starts with a lattice basis, and tries to improve its quality. This process is called lattice reduction. The most famous ones are probably the LLL [41] and HKZ [35, 40] reductions. Before defining them, we need the concept of size-reduction: a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is *size-reduced* if its GSO family satisfies $|\mu_{i,j}| \leq 1/2$ for all $j < i$.

DEFINITION 2.1 (HKZ-reduction). *A basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice L is said to be* Hermite-Korkine-Zolotarev-reduced *if it is size-reduced, the vector $\mathbf{b}_1$ reaches the lattice minimum (i.e., we have $\|\mathbf{b}_1\| = \lambda(L)$) and the projections of the $\mathbf{b}_i$'s for $i \geq 2$ orthogonally to the vector $\mathbf{b}_1$ (i.e., the vectors $\mathbf{b}_i - \mu_{i,1}\mathbf{b}_1$, for $i \geq 2$) form an HKZ-reduced basis of the lattice they span.*

The following immediately follows from the above definition and Minkowski's theorem. It is the sole property on HKZ-reduced bases that we will use:

LEMMA 2.2. *If $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is HKZ-reduced, then for any $i \leq d$, we have:*

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

HKZ-reduction is very strong, but expensive to compute. Contrarily, LLL-reduction can be achieved in polynomial time, but an LLL-reduced basis is of much lower quality. A basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is *LLL-reduced* if it is size-reduced and if its GSO satisfies the $(d-1)$ Lovász conditions: $\frac{3}{4} \cdot \left\|\mathbf{b}_{i-1}^*\right\|^2 \leq \left\|\mathbf{b}_i^* + \mu_{i,i-1}\mathbf{b}_{i-1}^*\right\|^2$. The definition of LLL-reduction implies that the GSO norms $\|\mathbf{b}_1^*\|, \ldots, \|\mathbf{b}_d^*\|$ do not drop too fast. As a consequence, LLL-reduced bases enjoy useful properties, like providing exponential approximations to SVP and CVP. In particular, their first vector is relatively short.

THEOREM 2.3 ([41]). *Let $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ be an LLL-reduced basis of a lattice L. Then we have $\|\mathbf{b}_1\| \leq 2^{\frac{d-1}{4}} \cdot (\det L)^{1/d}$. We also have $\|\mathbf{b}_i^*\| \geq 2^{\frac{-i+1}{2}} \|\mathbf{b}_1\|$ for any $i \leq d$. Finally, there exists an algorithm, called LLL, that takes as input any set of rational vectors and outputs in deterministic polynomial time an LLL-reduced basis of the lattice they span.*

We will also need the following properties on the LLL algorithm:

(i) If the input set of vectors is an LLL-reduced basis, then LLL returns exactly that basis.

(ii) During the execution of LLL, none of the quantities $\max_{j \leq d} \|\mathbf{b}_j^*\|$ can increase (see [41, p. 523]).

(iii) If the input set of vectors starts with a shortest non-zero lattice vector, then the output basis starts with the same vector.

2.3. Kannan's Algorithms. Kannan's algorithms rely on multiple calls to a lattice points enumeration procedure. The latter aims at computing all vectors of a given lattice that belong to a given hyperball. Let $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ be a basis of a lattice $L \subseteq \mathbb{Q}^n$, let $\mathbf{t} \in \mathbb{Q}^n$ be in the linear span of the $\mathbf{b}_i$'s and let $A \in \mathbb{Q}$. We aim at finding all lattice vectors $\sum_{i=1}^d x_i \mathbf{b}_i$ within squared distance A from the target $\mathbf{t}$. In the case of SVP, we take $\mathbf{t} = \mathbf{0}$. In the case of CVP, the target $\mathbf{t}$ could be outside of the span of the $\mathbf{b}_i$'s: in that case, we decompose $\mathbf{t}$ into two orthogonal components, one that lies in the linear span of the $\mathbf{b}_i$'s and one that is orthogonal to it. Suppose now that $\|\sum_i x_i \mathbf{b}_i - \mathbf{t}\|^2 \leq A$ for some integers x_i's. By considering the change of variable $\mathbf{b}_i \to \mathbf{b}_i^*$, we obtain:

$$\sum_{i \leq d} \left(x_i - t_i + \sum_{j > i} \mu_{j,i} x_j \right)^2 \|\mathbf{b}_i^*\|^2 \leq A, \quad \text{where} \quad \mathbf{t} = \sum_{i \leq d} t_i \mathbf{b}_i^*.$$

The left hand side being a sum of non-negative terms, the following equations hold:

$$\begin{aligned}
(x_d - t_d)^2 \cdot \|\mathbf{b}_d^*\|^2 &\leq A, \\
(x_{d-1} - t_{d-1} + \mu_{d,d-1} x_d)^2 \cdot \|\mathbf{b}_{d-1}^*\|^2 &\leq A - \ell_d, \\
&\cdots \\
\left(x_i - t_i + \sum_{j=i+1}^{d} \mu_{j,i} x_j\right)^2 \cdot \|\mathbf{b}_i^*\|^2 &\leq A - \sum_{j=i+1}^{d} \ell_j, \\
&\cdots \\
\left(x_1 - t_1 + \sum_{j=2}^{d} \mu_{j,i} x_j\right)^2 \cdot \|\mathbf{b}_1\|^2 &\leq A - \sum_{j=2}^{d} \ell_j,
\end{aligned} \tag{2.1}$$

where $\ell_i = (x_i - t_i + \sum_{j>i} x_j \mu_{j,i})^2 \cdot \|\mathbf{b}_i^*\|^2$. The enumeration algorithm considers all the solutions $x_d \in \mathbb{Z}$ to the first equation, then for each x_d it considers all the solutions $x_{d-1} \in \mathbb{Z}$ to the second equation, etc. It proceeds in a depth first tree search manner. The ith layer of the tree contains nodes labelled $(x_i, \ldots, x_d)$ corresponding to solutions of the $(d-i+1)$th equation above. Its sons are the solutions $(x'_{i-1}, x'_i, \ldots, x'_d)$ to the $(d-i+2)$th equation such that $x'_j = x_j$ for all $j \geq i$. The enumeration algorithm is given in Figure 2.1. For other variants, see [2]. The bit-cost of this algorithm is bounded by the number of loop iterations, up to a multiplicative factor that is polynomial in the bit-size of the input. Note that as described, the space complexity of the enumeration procedure is at least the bit-size of the output set of vectors, which may not be polynomially bounded with respect to the bit-size of the input. We discuss later how to avoid this issue when the enumeration algorithm is used within Kannan's algorithms.

Inputs: Basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d \in \mathbb{Q}^n$, a target $\mathbf{t} \in \mathbb{Q}^n$ in the span of the $\mathbf{b}_i$'s and a bound $A \in \mathbb{Q}$.
Output: All vectors $\mathbf{b} \in L(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ such that $\|\mathbf{b} - \mathbf{t}\|^2 \leq A$.

1. Compute (over $\mathbb{Q}$) the $\mathbf{b}_i^*$'s, the t_i's and the $\mu_{i,j}$'s for all $i \geq j$.
2. $\mathbf{x}{:=}\mathbf{0}, \mathbf{l}{:=}\mathbf{0}, S{:=}\emptyset$.
3. $i{:=}d$ and $x_d{:=}\left\lceil t_d - \frac{\sqrt{A}}{\|\mathbf{b}_d^*\|} \right\rceil$. While $i \leq d$, do
4. $\quad \ell_i{:=}(x_i - t_i + \sum_{j>i} x_j \mu_{j,i})^2 \|\mathbf{b}_i^*\|^2$.
5. $\quad$ If $i = 1$ and $\sum_{j=1}^{d} \ell_j \leq A$, then $S{:=}S \cup \{\mathbf{x}\}$, $x_1{:=}x_1 + 1$.
6. $\quad$ If $i \neq 1$ and $\sum_{j \geq i} \ell_j \leq A$, then
7. $\quad\quad i{:=}i-1$, $x_i{:=}\left\lceil t_i - \sum_{j>i}(x_j \mu_{j,i}) - \sqrt{\frac{A - \sum_{j>i} \ell_j}{\|\mathbf{b}_i^*\|^2}} \right\rceil$.
8. $\quad$ If $\sum_{j \geq i} \ell_j > A$, then $i{:=}i+1$, $x_i{:=}x_i + 1$.
9. Return S.

FIG. 2.1. *The Enumeration Algorithm.*

To solve SVP, Kannan provides an algorithm that computes HKZ-reduced bases, see Figure 2.2 (which actually describes Helfrich's variant [34] of Kannan's algorithm). The cost of the enumeration procedure dominates the overall cost and mostly depends on the quality (i.e., the slow decrease of the $\|\mathbf{b}_i^*\|$'s) of the input basis. The main idea behind Kannan's algorithm consists in spending an important amount of time pre-

computing a basis of excellent quality before calling the enumeration procedure. More precisely, it pre-computes a basis which is almost HKZ-reduced.

DEFINITION 2.4 (Quasi-HKZ-Reduction). *A basis* $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ *is called quasi-HKZ-reduced if it is size-reduced, if* $(\mathbf{b}_1, \mathbf{b}_2)$ *is LLL-reduced and the projections of the* $\mathbf{b}_i$*'s for* $i \geq 2$ *orthogonally to the vector* $\mathbf{b}_1$ *are an HKZ-reduced basis.*

Note that any quasi-HKZ-reduced basis is LLL-reduced.

Input: A rational basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice L.
Output: An HKZ-reduced basis of L.
1. If $d \leq 1$, return $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$.
2. LLL-reduce the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$.
3. Compute $\mathbf{b}'_i = \mathbf{b}_i - \mu_{i,1}\mathbf{b}_1$, the projection of $\mathbf{b}_i$ orthogonally to $\mathbf{b}_1$, for all $i \geq 2$.
4. HKZ-reduce the $(d-1)$-dimensional basis $(\mathbf{b}'_2, \ldots, \mathbf{b}'_d)$.
5. Extend the obtained $(\mathbf{b}'_i)_{i\geq 2}$'s into vectors of L by adding to them rational multiples of $\mathbf{b}_1$, in such a way that we have $|\widetilde{\mu}_{i,1}| \leq 1/2$ for any $i > 1$, providing a new basis $(\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_d)$.
6. If $(\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_d)$ is not quasi-HKZ-reduced, HKZ-reduce $(\widetilde{\mathbf{b}}_1, \widetilde{\mathbf{b}}_2)$ and go to Step 3.
7. Call the algorithm of Figure 2.1 with $\mathbf{t} = \mathbf{0}$ and $A = \|\widetilde{\mathbf{b}}_1\|^2$. Let $\widetilde{\mathbf{b}}_0$ be a shortest non-zero vector among the solutions.
8. $(\mathbf{c}_1, \ldots, \mathbf{c}_d)$:=LLL$(\widetilde{\mathbf{b}}_0, \ldots, \widetilde{\mathbf{b}}_d)$.
9. Compute $\mathbf{c}'_i = \mathbf{c}_i - \mu_{i,1}\mathbf{c}_1$, the projection of $\mathbf{c}_i$ orthogonally to $\mathbf{c}_1$, for all $i \geq 2$.
10. HKZ-reduce the $(d-1)$-dimensional basis $(\mathbf{c}'_2, \ldots, \mathbf{c}'_d)$.
11. Extend the obtained $(\mathbf{c}'_i)_{i\geq 2}$'s into vectors of L by adding to them rational multiples of $\mathbf{c}_1$, in such a way that we have $|\widetilde{\mu}_{i,1}| \leq 1/2$ for any $i > 1$, providing a new basis $(\widetilde{\mathbf{c}}_1, \ldots, \widetilde{\mathbf{c}}_d)$.
12. Return $(\widetilde{\mathbf{c}}_1, \ldots, \widetilde{\mathbf{c}}_d)$.

FIG. 2.2. *Kannan's HKZ-reduction Algorithm.*

Several comments need to be made on the algorithm of Figure 2.2. First, the algorithm aims at HKZ-reducing the input lattice. An SVP algorithm is easily obtained by first running the algorithm of Figure 2.2, and then returning the first vector of the output. Step 6 contains a recursive call in dimension 2: Theorem 2.3 may be used to show correctness in dimension 2. Steps 4 and 10 are recursive calls in dimension $d-1$. 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 vectors of L having appropriate values once projected orthogonally to $\mathbf{b}_1$, and subtracting a correct multiple of the vector $\mathbf{b}_1$ to ensure that $|\widetilde{\mu}_{i,1}| \leq 1/2$ for any i. The way it is written, Step 7 may require an amount of space that is not polynomially bounded in the bit-size of the input. The algorithm of Figure 2.1 can easily be modified to avoid this issue: instead of storing all vectors whose norms are below the prescribed bound, it suffices to keep (and update) the shortest non-zero vector found so far during the execution.

Kannan's CVP algorithm is given in Figure 2.3. It consists in HKZ-reducing the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ and then applying the enumeration algorithm possibly several times.

At first sight, the algorithm of Figure 2.3 may require a huge amount of space. This can be avoided by starting Step 5 each time a vector is found at Step 4, and going back to Step 4 at the end of any such recursive call. Step 5 is a recursive call in dimension $i - 1 < d$. Note that the vector $\mathbf{t}' - \sum_{j\geq i} x_j \mathbf{b}'_j$ belongs to the linear span of $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$. The correctness of the algorithm follows from the following facts:

Inputs: A rational basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice L and a rational target $\mathbf{t}$ in the span of the $\mathbf{b}_i$'s.
Output: A vector of L that is closest to $\mathbf{t}$.
1. HKZ-reduce the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$, by using the algorithm of Figure 2.2.
2. Let $i \leq d$ such that $\|\mathbf{b}_i^*\|$ is maximal.
3. For each $j \geq i$, decompose $\mathbf{b}_j$ as $\mathbf{b}_j = \mathbf{b}'_j + \mathbf{b}''_j$, with $\mathbf{b}'_j$ belonging to the span of $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$ and $\mathbf{b}''_j$ orthogonal to it. Proceed similarly with $\mathbf{t}$.
4. Call the algorithm of Figure 2.1 to find all vectors $\mathbf{c}'' = \sum_{j \geq i} x_j \mathbf{b}''_j$ within squared distance $\frac{d}{4}\|\mathbf{b}_i^*\|^2$ of $\mathbf{t}''$.
5. For any $\mathbf{c}''$, find a vector $\mathbf{c}' \in L[\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}]$ closest to $\mathbf{t}' - \sum_{j \geq i} x_j \mathbf{b}'_j$.
6. Among the vectors $\mathbf{c}' + \mathbf{c}''$, output one which is closest to $\mathbf{t}$.

FIG. 2.3. *Kannan's CVP Algorithm.*

(i) If $\sum_{j=1}^d x_j \mathbf{b}_j$ is a closest vector to $\mathbf{t}$, then we have $\|\sum_{j \geq i} x_j \mathbf{b}''_j - \mathbf{t}''\|^2 \leq \|\sum_{j=1}^d x_j \mathbf{b}_j - \mathbf{t}\|^2 \leq \frac{1}{4}\sum_{j=1}^d \|\mathbf{b}_j^*\|^2 \leq \frac{d}{4}\|\mathbf{b}_i^*\|^2$ (see for example [9]).

(ii) Under the same assumption, the vector $\sum_{j<i} x_j \mathbf{b}_j$ is closest to $\mathbf{t} - \sum_{j \geq i} x_j \mathbf{b}_j$ in the lattice $L[\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}]$.

(iii) The vector $\mathbf{t}' - \sum_{j \geq i} x_j \mathbf{b}'_j$ is the orthogonal projection of the vector $\mathbf{t} - \sum_{j \geq i} x_j \mathbf{b}_j$ onto the span of the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$.

2.4. Worst-case Complexities of Kannan's Algorithms. The main result of the present paper is to exhibit the exact worst-case complexity of Kannan's HKZ-reduction algorithm (and thus SVP algorithm), by lowering Helfrich's complexity upper bound [34], and by providing the first worst-case complexity lower bound.

THEOREM 2.5. *Given as inputs any quasi-HKZ-reduced basis* $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$, $\mathbf{t} = \mathbf{0}$ *and* $A = \|\mathbf{b}_1\|^2$, *the number of loop iterations occurring during the execution of the algorithm of Figure 2.1 is* $2^{O(d)} \cdot d^{\frac{d}{2e}}$. *Furthermore, there exist HKZ-reduced bases* $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ *such that given* $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$, $\mathbf{t} = \mathbf{0}$ *and* $A = \|\mathbf{b}_1\|^2$ *as inputs, the number of loop iterations occurring during the execution of the algorithm of Figure 2.1 is* $2^{O(d)} \cdot d^{\frac{d}{2e}}$.

As a consequence, given as input any d-dimensional basis of n-dimensional rational vectors with entries of bit-sizes $\leq \beta$, *the algorithm of Figure 2.2 returns an HKZ-reduced basis of the input lattice, in deterministic time* $\mathcal{P}(\beta, n, 2^d) \cdot d^{\frac{d}{2e}}$. *Finally, there exist input bases for which the running-time of the algorithm of Figure 2.2 is* $\geq 2^{O(d)} \cdot d^{\frac{d}{2e}}$.

We now prove the second part of Theorem 2.5. The rest of the paper will be devoted to the cost analysis of the enumeration algorithm. What follows is classical (e.g., see [34]), but we provide it for the sake of completeness.

Proof. We prove the second part, assuming that the first part holds.

We first consider the bit-sizes of the GSO of a rational basis. Wlog we consider an integer basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ (otherwise, one may scale by the product of the denominators of the coefficients of the basis vectors). Let β be the maximum of the bit-sizes of the entries of the $\mathbf{b}_i$'s. Let $i \leq d$. The vector $\mathbf{b}_i^*$ may be written as $\mathbf{b}_i^* = \mathbf{b}_i + \sum_{j<i} y_j \mathbf{b}_j$, with $y_j \in \mathbb{Q}$ for $j < i$. We have $\langle \mathbf{b}_j, \mathbf{b}_i^* \rangle = 0$ for $j < i$, which implies the matrix identity $C^T C \cdot \mathbf{y} = -C^T \cdot \mathbf{b}_i$, where C is the matrix whose columns are $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$ and the entries of $\mathbf{y}$ are the y_j's. Cramer's rule implies that $y_j = \frac{a_j}{\det(C^T C)}$, for some integer a_j. This proves that y_j is a rational number. Moreover, Hadamard's inequality provides $\det(C^T C) \leq 2^{\mathcal{P}(n,\beta)}$. As a consequence,

the denominator of y_j may be written on $\mathcal{P}(n,\beta)$ bits. The same argument holds for its numerator. We thus obtain that the bit-size of the rational vector $\mathbf{b}_i^*$ is $\mathcal{P}(n,\beta)$. Since $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\|\mathbf{b}_j^*\|^2}$, this also holds for the $\mu_{i,j}$'s.

We now show that the bit-size of any vector occurring during the execution of the algorithm of Figure 2.2 is $\mathcal{P}(n,\beta)$. If we unroll Steps 4 and 10, we see that the only operations performed on the set of d or $d+1$ vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots$ generating the lattice L are of the following type:

1. consider the lattice $L = L[\mathbf{b}_i^{(i)}, \ldots, \mathbf{b}_d^{(i)}]$, where $\mathbf{b}_j^{(i)} = \mathbf{b}_j - \sum_{k<i} \mu_{j,k} \mathbf{b}_k^*$ is the projection of $\mathbf{b}_j$ orthogonally to the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$; find a shortest non-zero vector $\mathbf{b}^{(i)}$ in L, by using the algorithm of Figure 2.1.

2. LLL-reduce $\mathbf{b}^{(i)}, \mathbf{b}_i^{(i)}, \ldots, \mathbf{b}_d^{(i)}$, where $\mathbf{b}^{(i)}$ is a shortest non-zero vector in L.

We prove that the quantity $\max_{j \leq d} \|\mathbf{b}_j^*\|$ cannot increase under the action of either type of step. We already know it for type (2) (see the discussion after Theorem 2.3). We now consider type (1). Let $\mathbf{c}_1, \ldots, \mathbf{c}_{d+1}$ be the vectors after the enumeration: the vector $\mathbf{c}_i$ is the new one, and we have $\mathbf{c}_j = \mathbf{b}_j$ for $j < i$ and $\mathbf{c}_j = \mathbf{b}_{j-1}$ for $j > i$. For $j < i$, as $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$ are not modified, the quantity $\|\mathbf{b}_j^*\|$ remains constant. If $j > i$, the vector $\mathbf{c}_j^*$ is the vector $\mathbf{b}_{j-1}^*$ after projection orthogonally to the new vector $\mathbf{c}_i^*$, and thus $\|\mathbf{c}_j^*\| \leq \|\mathbf{b}_{j-1}^*\|$. Finally, for $j = i$, the correctness of the algorithm of Figure 2.1 implies that $\|\mathbf{c}_i^*\| \leq \|\mathbf{b}_i^*\|$.

From the discussion above, we conclude that for any j and at any moment during the execution of the algorithm of Figure 2.2, we have $\|\mathbf{b}_j^*\| \leq B$, where $B = 2^{\mathcal{P}(n,\beta)}$ is the maximum of the norms of the input vectors. Since all considered bases are size-reduced, for any considered vector $\mathbf{b}_j^{(i)}$, we have $\|\mathbf{b}_j^{(i)}\| \leq \sqrt{d} \max_{k \leq j} \|\mathbf{b}_k^*\| \leq \sqrt{d} B$. As a consequence, any vector occurring during the execution of the algorithm is of bit-size $\mathcal{P}(n,\beta)$. It is also the case for any GSO coefficient that occurs.

We now show that the x_i's computed during the calls to the enumeration algorithm of Figure 2.1 are of bit-sizes $\mathcal{P}(n,\beta)$. Any considered x_i is such that $|x_i| \leq \sum_{j>i} |x_j||\mu_{j,i}| + \frac{\sqrt{A}}{\|\mathbf{b}_i^*\|} + 1$ (see the list of equations (2.1)). Since the enumeration is called on an LLL-reduced basis $(\mathbf{c}_1, \ldots, \mathbf{c}_k)$ with $A = \|\mathbf{c}_1\|^2$ and $k \leq d$, Theorem 2.3 implies that $|x_i| \leq \sum_{j>i} |x_j| + 2^d + 1$. The $|x_i|$'s are thus no greater than the y_i's defined by $y_i = \sum_{j>i} y_j + 2^d + 1$ and $y_k = 2^d + 1$. We have $y_i = 2y_{i+1}$ when $i < k$, which allows us to conclude that any occurring x_i has magnitude $\leq 2^{\mathcal{P}(d)}$. This completes the proof that all the rationals considered during the execution of the algorithm are of bit-sizes $\mathcal{P}(n,\beta)$.

We now recall Helfrich's proof [34] that the number of iterations of the loop made of Steps 3–6 of the algorithm of Figure 2.2 is $O(\log d)$. If the test of Step 6 fails, then the vector $\mathbf{b}_1$ is replaced by a vector $\mathbf{b}_1'$ such that $\|\mathbf{b}_1'\| \leq \sqrt{2}\sqrt{\|\mathbf{b}_1\| \cdot \|\mathbf{b}_2^*\|}$ (see Theorem 2.3). Since the test fails, the vector $\mathbf{b}_1$ cannot be a shortest non-zero vector of the lattice $L = L[\mathbf{b}_1, \ldots, \mathbf{b}_d]$. A shortest non-zero vector must therefore make use of a non-zero integer multiple of the other $\mathbf{b}_i$'s. Since the projections of the other $\mathbf{b}_i$'s orthogonally to $\mathbf{b}_1$ form an HKZ-reduced basis, we must have $\lambda(L) \geq \|\mathbf{b}_2^*\|$. Overall, we obtain $\|\mathbf{b}_1'\| \leq \sqrt{2}\sqrt{\|\mathbf{b}_1\|\lambda(L)}$, which can be rewritten as $\frac{\|\mathbf{b}_1'\|}{\lambda(L)} \leq \sqrt{2}\sqrt{\frac{\|\mathbf{b}_1\|}{\lambda(L)}}$. Initially, the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is LLL-reduced, which implies that $\|\mathbf{b}_1\| \leq 2^d \lambda(L)$ (see Theorem 2.3). We thus obtain that within $O(\log d)$ iterations, we have $\|\mathbf{b}_1\| \leq 8\lambda(L)$. Each further iteration decreases $\|\mathbf{b}_1\|$ by a factor $\geq \frac{\sqrt{3}}{2}$ (since $\|\mathbf{b}_1'\| \leq \|\mathbf{b}_2\| \leq \frac{\sqrt{3}}{2}\|\mathbf{b}_1\|$), which yields the result.

Let $\mathcal{C}(d,n,\beta)$ be the worst-case cost of the algorithm of Figure 2.2. So far,

we have that $\mathcal{C}(d,n,\beta) \leq \mathcal{P}(n,\beta)\cdot\mathcal{C}'_d$, where the sequence $(\mathcal{C}'_d)$ satisfies the equation $\mathcal{C}'_d \leq (K_1\log d)\cdot\mathcal{C}'_{d-1} + d^{\frac{d}{2e}}\cdot 2^{K_2 d}$, for some constants K_1 and K_2 Note that we just used the first part of the theorem. This implies that for any d, we have $\mathcal{C}'_d \leq \sum_{i\leq d}(K_1\log d)^{d-i} i^{\frac{i}{2e}} 2^{K_2 d}\cdot\mathcal{C}'_1 = 2^{O(d)}\cdot d^{\frac{d}{2e}}$.

We now prove the last statement of the theorem. Suppose that an HKZ-reduced basis satisfying the second claim is given as input to Algorithm 2.2. As an HKZ-reduced basis is LLL-reduced, Step 2 does not modify the input basis. Definition 2.1 implies that Steps 3–5 do not modify it either. Also, an HKZ-reduced is always quasi-HKZ-reduced, and therefore the condition of Step 6 is not satisfied, and the execution proceeds with Step 7. Finally, the second claim of the theorem provides us that Step 7 costs $2^{O(d)}\cdot d^{\frac{d}{2e}}$ bit operations. □

We also obtain the following results on Kannan's CVP algorithm.

THEOREM 2.6. *Given as inputs a basis $(\mathbf{b}_1,\ldots,\mathbf{b}_d)$ of a lattice L with $\|\mathbf{b}_1\| = \max_{i\leq d}\|\mathbf{b}_i^*\|$, a target vector $\mathbf{t}$ in the linear span of the $\mathbf{b}_i$'s and $A \geq \frac{d}{4}\|\mathbf{b}_1\|^2$, the number of loop iterations occurring during the execution of the algorithm of Figure 2.1 is $\leq 2^{O(d)}\cdot\frac{\sqrt{A/d}^d}{\det L}$. As a consequence, given as inputs a d-dimensional basis made of n-dimensional rational vectors with entries of bit-sizes $\leq\beta$ and a target rational vector $\mathbf{t}$ with entries of bit-sizes $\leq\beta$, the algorithm of Figure 2.3 returns a closest vector to $\mathbf{t}$ in the lattice spanned by the $\mathbf{b}_i$'s, in deterministic time $\mathcal{P}(\beta,n,2^d)\cdot d^{\frac{d}{2}}$.*

Proof. We will show the first part of the result at the end of §3. The costs of the rational arithmetic operations involved in the execution of the algorithm of Figure 2.3 and its calls to the algorithm of Figure 2.1 can be bounded in a fashion similar to what we did in the proof of Theorem 2.5. Also, Theorem 2.5 implies that Step 1 of the algorithm of Figure 2.3 is performed within the prescribed amount of time.

As a consequence of the first statement of the theorem, of Minkowski's bound and of the HKZ-reducedness of the basis after Step 1, if $\mathcal{C}(d,n,\beta)$ denotes the worst-case cost of the algorithm of Figure 2.3, then we have $\mathcal{C}(d,n,\beta)\leq\mathcal{P}(n,\beta)\mathcal{C}'(d)$ with $\mathcal{C}'(d)\leq 2^{K(d-i+1)}d^{\frac{d-i+1}{2}}\cdot\mathcal{C}'(i)$, for some constant K and with i as in Step 2 of the algorithm of Figure 2.3. This provides the result. □

3. Complexity of the Enumeration Procedure. The present section is devoted to providing complexity (upper and lower) bounds for the enumeration algorithm (described in Figure 2.1). The latter dominates the costs of both CVP and HKZ algorithms. The bounds involve geometric data related to the input basis $\mathbf{b}_1,\ldots,\mathbf{b}_d$, namely the GSO norms $\|\mathbf{b}_i^*\|$. We shall obtain an upper and a lower bound of very similar shapes: the main term of those bounds actually match in the case where the GSO norms form a non-increasing sequence. In this section, the input basis $(\mathbf{b}_1,\ldots,\mathbf{b}_d)$, the input target $\mathbf{t}$ and the input bound A are arbitrary.

The complexity of the enumeration procedure of Figure 2.1 is its number of loop iterations, up to some polynomial in n and the maximum of the bit-sizes of the entries of the input vectors.

LEMMA 3.1. *Let $(\mathbf{b}_1,\ldots,\mathbf{b}_d)$, $\mathbf{t}$ and A be valid inputs to the algorithm of Figure 2.1. We define*

$$\mathcal{E}_i = \left\{ \mathbf{y}\in\mathbb{R}^{d-i+1} : \left\| \sum_{j=i}^{d} y_j \mathbf{b}_j^{(i)} - \mathbf{t}^{(i)} \right\|^2 \leq A \right\},$$

and $\mathbf{b}_j^{(i)} = \mathbf{b}_j - \sum_{k<i}\mu_{j,k}\mathbf{b}_k^*$ is the vector $\mathbf{b}_j$ once projected orthogonally to the linear

span of the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$*, and* $\mathbf{t}^{(i)}$ *is defined similarly. The number of loop iterations performed during the execution of the algorithm of Figure 2.1 is contained in the interval* $[\sum_{i\leq d} N_i, 3\sum_{i\leq d} N_i]$*, where* $N_i = |\mathcal{E}_i \cap \mathbb{Z}^{d-i+1}|$.

Proof. The lower bound derives from the fact that the enumeration algorithm finds all integer solutions to (2.1). To prove the upper bound, we define a valid truncated coordinate as an integer tuple $(x_i, \ldots, x_d)$ satisfying $\| \sum_{j=i}^{d} x_j \mathbf{b}_j^{(i)} - \mathbf{t}^{(i)} \|^2 \leq A$. Every loop iteration corresponds to a different truncated coordinate. The upper bound holds because any loop iteration corresponding to a valid truncated coordinate $(x_i, \ldots, x_d)$ is followed by at most two loop iterations corresponding to truncated coordinates that are not valid: these can only be $(x_i + 1, \ldots, x_d)$ and $(x_{i-1}, x_i \ldots, x_d)$ for at most one integer x_{i-1} (defined at Step 7). □

The set $\mathcal{E}_i$ is a skewed hyper-ellipsoid. In the following lemma, we show that we can instead count integer points contained in hyper-ellipsoids with orthogonal axes.

LEMMA 3.2. *We keep the notations of Lemma 3.1. For* $i \leq d$ *and* $B \geq 0$*, we define* $\mathcal{F}_i(B) = \{\mathbf{y} \in \mathbb{R}^{d-i+1} : \sum_{j\geq i} y_j^2 \|\mathbf{b}_j^*\|^2 \leq B\}$*. We have:*

$$\left| \mathcal{F}_i\left(\frac{4A}{9}\right) \cap (\mathbb{Z} \setminus 0)^{d-i+1} \right| \leq N_i \leq \left| \mathcal{F}_i(4A) \cap \mathbb{Z}^{d-i+1} \right|.$$

Proof. We let $\mathbf{t} = \sum_{j\leq d} t_j \mathbf{b}_j^*$ be the expression of $\mathbf{t}$ with respect to the GSO of the $\mathbf{b}_i$'s. Let $\phi : \mathbb{R}^{d-i+1} \to \mathbb{R}^{d-i+1}$ be defined by $\phi(\mathbf{y}) = \mathbf{z}$ with $z_j = y_j + \left\lfloor t_j - \sum_{k>j} \mu_{k,j} z_k \right\rceil$ for any $j \geq i$. The function ϕ is a bijection. Furthermore,

$$\sum_{j\geq i} z_j \mathbf{b}_j^{(i)} - \mathbf{t}^{(i)} = \sum_{j\geq i} \left(z_j - t_j + \sum_{k>j} \mu_{k,j} z_k \right) \mathbf{b}_j^* = \sum_{j\geq i} (y_j + \delta_j) \mathbf{b}_j^*,$$

for some $\delta_j \in [-1/2, 1/2]$ (for all $j \geq i$).

For any non-zero integer y and any $\delta \in [-1/2, 1/2]$, we have $(y + \delta)^2 \leq \frac{9}{4} y^2$. Hence, for $\mathbf{y} \in \mathcal{F}_i(\frac{4A}{9}) \cap (\mathbb{Z} \setminus 0)^{d-i+1}$, the z_j's are integers and

$$\| \sum_{j\geq i} z_j \mathbf{b}_j^{(i)} - \mathbf{t}^{(i)} \|^2 = \sum_{j\geq i} (y_j + \delta_j)^2 \|\mathbf{b}_j^*\|^2 \leq \frac{9}{4} \sum_{j\geq i} y_j^2 \|\mathbf{b}_j^*\|^2 \leq A.$$

This implies that $\phi(\mathcal{F}_i(\frac{4A}{9}) \cap (\mathbb{Z} \setminus 0)^{d-i+1}) \subseteq \mathcal{E}_i \cap \mathbb{Z}^{d-i+1}$, which means that the lower bound holds.

For any integer y and any $\delta \in [-1/2, 1/2]$, we have $(y + \delta)^2 \geq y^2/4$. This implies that $\phi^{-1}(\mathcal{E}_i \cap \mathbb{Z}^{d-i+1}) \subseteq \mathcal{F}_i(4A) \cap \mathbb{Z}^{d-i+1}$, which provides the upper bound. □

We now consider the quantity $\left| \mathcal{F}_i(B) \cap \mathbb{Z}^{d-i+1} \right|$, for $B \geq 0$. The proof of the first inequality below is inspired from [45, Lemma 1].

LEMMA 3.3. *We keep the same notations as above. We have the following, for all* i*:*

$$\left| \mathcal{F}_i(B) \cap \mathbb{Z}^{d-i+1} \right| \leq ((1 + \sqrt{\pi})\mathrm{e})^d \cdot \prod_{j\geq i} \max\left(1, \frac{\sqrt{B}}{\sqrt{d}\|\mathbf{b}_j^*\|}\right).$$

Furthermore, if i *is such that* $\|\mathbf{b}_j^*\| \leq \sqrt{\frac{B}{d}}$ *for all* $j \geq i$*, then:*

$$\left| \mathcal{F}_i(B) \cap (\mathbb{Z} \setminus 0)^{d-i+1} \right| \geq \prod_{j\geq i} \frac{\sqrt{B}}{\sqrt{d}\|\mathbf{b}_j^*\|}.$$

Proof. We start with the first inequality. Let $\mathbf{1}_{\mathcal{F}_i(B)}$ denote the indicator function of the set $\mathcal{F}_i(B)$. We have the following sequence of relations:

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

where $\Theta(t)=\sum_{x\in\mathbb{Z}}\exp(-tx^2)$ is defined for $t>0$. Notice that

$$\Theta(t)=1+2\sum_{x\geq 1}\exp(-tx^2)\leq 1+2\int_0^\infty \exp(-tx^2)\mathrm{d}x = 1+\sqrt{\frac{\pi}{t}}.$$

Hence $\Theta(t)\leq\frac{1+\sqrt{\pi}}{\sqrt{t}}$ for $t\leq 1$ and $\Theta(t)\leq 1+\sqrt{\pi}$ for $t\geq 1$. This provides the first assertion of the lemma.

We now prove the second inequality. The set $\mathcal{F}_i(B)\cap(\mathbb{R}\setminus 0)^{d-i+1}$ contains the subset

$$\prod_{j=i}^{d}\left(\left[-\frac{\sqrt{B}}{\sqrt{d}\|\mathbf{b}_j^*\|},\frac{\sqrt{B}}{\sqrt{d}\|\mathbf{b}_j^*\|}\right]\setminus 0\right).$$

This implies that

$$\left|\mathcal{F}_i(B)\cap(\mathbb{Z}\setminus 0)^{d-i+1}\right|\geq\prod_{j=i}^{d}\left(2\left\lfloor\frac{\sqrt{B}}{\sqrt{d}\|\mathbf{b}_j^*\|}\right\rfloor\right)\geq\prod_{j=i}^{d}\frac{\sqrt{B}}{\sqrt{d}\|\mathbf{b}_j^*\|}.$$

The definition of i provides the result. □

We now prove the first statement of Theorem 2.6. Thanks to the assumptions $A\geq\frac{d}{4}\|\mathbf{b}_1\|^2$ and $\|\mathbf{b}_1\|=\max_{i\leq d}\|\mathbf{b}_i^*\|$, we have $B:=4A\geq d\|\mathbf{b}_i^*\|^2$ for all $i\leq d$. The upper bounds of Lemmas 3.1, 3.2 and 3.3 then give that the number of loop iterations performed during the execution of the enumeration algorithm is upper bounded by

$$2^{O(d)}\sum_{i\leq d}\frac{\sqrt{A/d}^{d-i+1}}{\prod_{j\geq i}\|\mathbf{b}_j^*\|}\leq 2^{O(d)}\frac{\sqrt{A/d}^{d}}{\det(L[\mathbf{b}_1,\ldots,\mathbf{b}_d])}.$$

The lemmas above also provide the following upper bound for the number of loop iterations of the enumeration routine, which holds in a more general context.

THEOREM 3.4. *Let $(\mathbf{b}_1,\ldots,\mathbf{b}_d)$, $\mathbf{t}$ and A be valid inputs to the algorithm of Figure 2.1. The number of loop iterations performed during the execution of the latter is upper bounded by*

$$2^{O(d)}\cdot\max_{I\subseteq[\![1,d]\!]}\left(\frac{(\sqrt{A})^{|I|}}{(\sqrt{d})^{|I|}\prod_{i\in I}\|\mathbf{b}_i^*\|}\right).$$

In §4, we will bound the quantity above for quasi-HKZ-reduced bases, and thus derive the first statement of Theorem 2.5. Finally, note that the lower bounds of Lemmas 3.1, 3.2 and 3.3 imply that to prove the second statement of Theorem 2.5, it suffices to construct an HKZ-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ such that $\frac{(\|\mathbf{b}_1\|/\sqrt{d})^{d-i+1}}{\prod_{j\geq i}\|\mathbf{b}_j^*\|}$ is large, for an i such that for any $j \geq i$ we have $\|\mathbf{b}_j^*\| \leq \frac{2}{3}\frac{\|\mathbf{b}_1\|}{\sqrt{d}}$. We will fulfill this task in §5.

4. On the Geometry of HKZ-Reduced Bases. In this section, we assume that $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is quasi-HKZ-reduced, that $d \geq 2$ and that $A = \|\mathbf{b}_1\|^2$. We aim at bounding the quantity $\max_{I\subseteq[\![1,d]\!]}\left(\frac{(\sqrt{A})^{|I|}}{(\sqrt{d})^{|I|}\prod_{i\in I}\|\mathbf{b}_i^*\|}\right)$ from Theorem 3.4. Our first step consists in strengthening the quasi-HKZ-reducedness hypothesis into an HKZ-reducedness hypothesis. Let $I \subseteq [\![1, d]\!]$. If $1 \notin I$, then, because of the quasi-HKZ-reducedness assumption:

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

If $1 \in I$, then we have, by removing the term $\|\mathbf{b}_1^*\|$ from the product:

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

As a consequence of Theorem 3.4, the following provides the first assertion of Theorem 2.5. Note that the second inequality simply derives from the inequality $\frac{\log x}{x} \leq \frac{1}{\mathrm{e}}$ for $x \geq 1$.

THEOREM 4.1. *Let $\mathbf{b}_1, \ldots, \mathbf{b}_d$ be an HKZ-reduced basis. Let $I \subseteq [\![1, d]\!]$. Then*

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

The technicality of the proof of Theorem 4.1 increases with the non-connexity of the set I. In a first step, we will consider the case where I is an interval. Note that if the sequence of the $\|\mathbf{b}_i^*\|$'s were non-increasing, then all the sets I that derive from the upper bound of Lemma 3.3 would be intervals, and thus the study of intervals would suffice to prove the first statement of Theorem 2.5. The difficulties arise when the shape of the set I under study becomes more complicated. The strategy 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. A block is a group of consecutive vectors $\mathbf{b}_i, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_{j-1}$ such that $i, \ldots, k-1 \notin I$ and $k, \ldots, j-1 \in I$, for some k. Over each block, Lemma 4.2 relates the average norm of the last vectors to the average norm of the block. We consider the blocks by decreasing indices (in Lemma 4.6), and use an amortised analysis to combine the local behaviours on blocks and eventually obtain a global bound. This recombination is very tight, and in order to get the desired bound we use "parts of vectors", or, to be more specific, non-integral powers of their norms. This is why we need to introduce the $\widetilde{\pi}$'s (in Definition 4.4). A final convexity argument provided by Lemma 4.7 gives the result.

4.1. Handling Intervals. For any $I \subseteq [\![1, d]\!]$, we define the average norm over I as $\pi_I = \left(\prod_{i\in I} \|\mathbf{b}_i^*\|\right)^{\frac{1}{|I|}}$. The following lemma allows us to handle sets I that are intervals. It generalizes Minkowski's theorem (consider $k = 1$) and can be interpreted as an "averaged" version of [65, Lemma 4].

LEMMA 4.2. *For all $1 \leq k < d$, we have the two following relations*

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

where $\Gamma_d(k) = \prod_{i=d-k}^{d-1} (\gamma_{i+1})^{\frac{1}{2i}}$. For later use, we define $\Gamma_d(0) = 1$.

Proof. We start with the first identity. We prove it by induction on k. For $k = 1$, this directly comes from the definition of γ_d. Assume that the identity holds for a given $k \geq 1$. We are to prove that it also holds for $k+1$. We can rewrite the induction hypothesis as

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

which is itself equivalent to

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

As the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is HKZ-reduced, the vector $\mathbf{b}_k^*$ is a shortest non-zero vector of the projection of the lattice orthogonally to the linear span of $\mathbf{b}_1, \ldots, \mathbf{b}_{k-1}$. The definition of Hermite's constant gives us $\|\mathbf{b}_{k+1}^*\| \leq \sqrt{\gamma_{d-k}}^{\frac{d-k}{d-k-1}} \cdot \pi_{[\![k+2,d]\!]}$. Combined with the equation above, this gives:

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

Raising the last identity to the power $\frac{k}{k+1}$ yields the result.

To obtain the second inequality, it suffices to raise the first one to the power $\frac{k}{d}$, multiply both sides by $\pi_{[\![k+1,d]\!]}^{\frac{d-k}{d}}$ and use the identity $\det L = \pi_{[\![1,k]\!]}^k \cdot \pi_{[\![k+1,d]\!]}^{d-k}$. □

The following result provides a bound on the quantity $\Gamma_d(k)$ from Lemma 4.2. The proof is a rigorous version of the sequence of identities:

$$\log \Gamma_d(k) \approx \int_{x=d-k}^{d} \frac{1}{2x} \log x \, \mathrm{d}x \approx \frac{\log^2(d) - \log^2(d-k)}{4} \lesssim \frac{\log d}{2} \log \frac{d}{d-k}.$$

LEMMA 4.3. *For all $1 \leq k < d$, we have $\Gamma_d(k) \leq \sqrt{d}^{\log \frac{d+1}{d+1-k}}$.*

Proof. For $d \leq 3$, the result follows by explicit computations, using $\gamma_2^2 = 4/3$, $\gamma_3^3 = 3$. In what follows, we thus assume that $d \geq 4$.

We now prove the result by induction on k. For $k = 1$, the bound follows from Minkowski's theorem (i.e., from the identity $\gamma_d \leq \frac{d+4}{4}$): it suffices to show that $\frac{d+4}{4} \leq d^{-(d-1)\log(1-\frac{1}{d+1})}$; to get the latter, note that $\log(1 - \frac{1}{d+1}) \leq -\frac{1}{d+1}$, so that $d^{-(d-1)\log(1-\frac{1}{d+1})} \geq d^{\frac{d-1}{d+1}}$. We now prove that $d^{\frac{d-1}{d+1}} \geq \frac{d+4}{4}$, which we will re-use later on. Since $d^{-\frac{2}{d+1}} \geq 1/2$ for $d \geq 4$, we have $d \cdot d^{-\frac{2}{d+1}} \geq \frac{d}{2} \geq \frac{d+4}{4}$.

Suppose now that the result holds for some $k \geq 1$. We are to prove that it also holds for $k+1$. We can now suppose that $d \geq 3$. Define $G_d(k) = \frac{1}{2} \log d \log \frac{d+1}{d+1-k}$. To

obtain the result, it suffices to prove that $\log \Gamma_d(k+1) - \log \Gamma_d(k) \leq G_d(k+1) - G_d(k)$. We have

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

From the upper bound $\gamma_d \leq \frac{d+4}{4}$, we obtain:

$$\log \Gamma_d(k+1) - \log \Gamma_d(k) = \frac{1}{2} \frac{\log \gamma_{d-k}}{d-k-1} \leq \frac{1}{2} \frac{\log \frac{d-k+4}{4}}{d-k-1}.$$

Now, since the sequence $\left(\frac{(n+1) \log \frac{n+4}{4}}{n-1} \right)_{n \geq 3}$ is increasing, we have:

$$\frac{(d+1-k) \log \frac{d-k+4}{4}}{d-k-1} \leq \max \left(3 \log \frac{3}{2}, \frac{d}{d+2} \log \frac{d+3}{4} \right) \leq \log d,$$

where the last inequality follows from the case $k = 1$. □

We now extend the study of the $\pi_{[\![1,k]\!]}$'s to non-integer intervals. This is needed to study the π_I's for general sets I, because we will use "extended intervals" for which the extension is a "fractional part" of the vector at the left of the boundary of the interval.

DEFINITION 4.4. *If $1 \leq x_1 \leq x_2 \leq d$, with $x_1 \in \mathbb{R}$ and $x_2 \in \mathbb{Z}$, we define:*

$$\begin{aligned} \widetilde{\pi}_{[x_1,x_2]} &= \left(\|\mathbf{b}^*_{\lfloor x_1 \rfloor}\|^{1-x_1+\lfloor x_1 \rfloor} \cdot \prod_{i=\lfloor x_1 \rfloor+1}^{x_2} \|\mathbf{b}^*_i\| \right)^{\frac{1}{x_2-x_1+1}} \\ &= \left(\pi_{[\![\lfloor x_1 \rfloor, x_2]\!]}\right)^{\frac{(x_2-\lfloor x_1 \rfloor+1)(1-x_1+\lfloor x_1 \rfloor)}{x_2-x_1+1}} \cdot \left(\pi_{[\![\lfloor x_1 \rfloor+1, x_2]\!]}\right)^{\frac{(x_2-\lfloor x_1 \rfloor)(x_1-\lfloor x_1 \rfloor)}{x_2-x_1+1}}. \end{aligned}$$

Note that Definition 4.4 is a sound extension of the definition of the π_I's where I is an integral interval, since $\widetilde{\pi}_{[x_1,x_2]} = \pi_{[\![x_1,x_2]\!]}$ when $x_1 \in \mathbb{Z}$. The following lemma extends Lemma 4.2 to the case where k is not necessarily an integer.

LEMMA 4.5. *If $1 \leq x_1 \leq x_2 < d$ are real, then $\widetilde{\pi}_{[x_2,d]} \geq \sqrt{d}^{\log \frac{d+1-x_2}{d+1-x_1}} \cdot \widetilde{\pi}_{[x_1,d]}$.*

Proof. First note that, as a consequence of the second inequality of Lemma 4.2 (applied the sublattice spanned by the projections of the last vectors orthogonally to $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$), we have, for $i, j \in \mathbb{Z}$ with $1 \leq i \leq j \leq d$,

$$\pi_{[\![j,d]\!]} \geq \Gamma_{d-i+1}(j-i)^{-1} \cdot \pi_{[\![i,d]\!]}. \tag{4.1}$$

We define $\lambda_i = \frac{(d-\lfloor x_i \rfloor+1)(1-x_i+\lfloor x_i \rfloor)}{d-x_i+1} \in [0,1]$, for $i \in \{1,2\}$. Then we have:

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

Note that since $x_1 \leq x_2$, either $\lfloor x_1 \rfloor < \lfloor x_2 \rfloor$ or $\lfloor x_1 \rfloor = \lfloor x_2 \rfloor$. The lemma easily holds when $x_1 = x_2$, so we now assume that $x_1 < x_2$. In the second case, since the function $x \mapsto \frac{u-x}{v-x}$ is decreasing when $u < v$ and for $x < v$, 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: $\lambda_1 \leq \lambda_2$. Then we must have $\lfloor x_1 \rfloor < \lfloor x_2 \rfloor$. We define G as

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

By using Equation (4.1) three times, we get:

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

Now, Lemma 4.3 gives that

$$\frac{\log G}{\log \sqrt{d}} \leq \lambda_1 \log \frac{d-\lfloor x_1 \rfloor+2}{d-\lfloor x_2 \rfloor+2} + (\lambda_2-\lambda_1) \log \frac{d-\lfloor x_1 \rfloor+1}{d-\lfloor x_2 \rfloor+2} + (1-\lambda_2) \log \frac{d-\lfloor x_1 \rfloor+1}{d-\lfloor x_2 \rfloor+1},$$

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

$$H := \lambda_1 \frac{d-\lfloor x_1 \rfloor+2}{d-\lfloor x_2 \rfloor+2} + (\lambda_2-\lambda_1) \frac{d-\lfloor x_1 \rfloor+1}{d-\lfloor x_2 \rfloor+2} + (1-\lambda_2) \frac{d-\lfloor x_1 \rfloor+1}{d-\lfloor x_2 \rfloor+1}.$$

To complete the proof of this case, it suffices to prove that $H \leq \frac{d-x_1+1}{d-x_2+1}$. Let $n_i = d - \lfloor x_i \rfloor$ and $y_i = 1 - x_i + \lfloor x_i \rfloor$ for $i \in \{1,2\}$. It suffices to prove that for any integers $n_1 > n_2 \geq 0$ and any reals $y_1, y_2 \in [0,1)$ (after regrouping the λ_1's):

$$\frac{(n_1+1)y_1}{n_1+y_1} \frac{1}{n_2+2} + \frac{(n_2+1)y_2}{n_2+y_2} \frac{n_1+1}{n_2+2} + \frac{n_2(1-y_2)}{n_2+y_2} \frac{n_1+1}{n_2+1} - \frac{n_1+y_1}{n_2+y_2} \leq 0.$$

By differentiating with respect to y_1, we obtain $\frac{n_1+1}{n_2+2} \frac{n_1}{(n_1+y_1)^2} - \frac{1}{n_2+y_2}$, which is always ≤ 0. It is therefore sufficient to prove the above for $y_1 = 0$, i.e., after multiplication by $n_2 + y_2$:

$$(n_2+1)y_2 \frac{n_1+1}{n_2+2} + n_2(1-y_2) \frac{n_1+1}{n_2+1} - n_1 \leq 0.$$

As the above increases with y_2, it suffices to prove it for $y_2 = 1$, which is easy.

Second case: $\lambda_1 > \lambda_2$. We define G' by

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

By using Equation (4.1) three times, we get:

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

Now, Lemma 4.3 gives us that:

$$\frac{\log G'}{\log \sqrt{d}} \leq \lambda_2 \log \frac{d-\lfloor x_1 \rfloor+2}{d-\lfloor x_2 \rfloor+2} + (\lambda_1-\lambda_2) \log \frac{d-\lfloor x_1 \rfloor+2}{d-\lfloor x_2 \rfloor+1} + (1-\lambda_1) \log \frac{d-\lfloor x_1 \rfloor+1}{d-\lfloor x_2 \rfloor+1},$$

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

$$H' := \lambda_2 \frac{d-\lfloor x_1 \rfloor+2}{d-\lfloor x_2 \rfloor+2} + (\lambda_1-\lambda_2) \frac{d-\lfloor x_1 \rfloor+2}{d-\lfloor x_2 \rfloor+1} + (1-\lambda_1) \frac{d-\lfloor x_1 \rfloor+1}{d-\lfloor x_2 \rfloor+1}.$$

To conclude, it suffices to obtain $H' \leq \frac{d-x_1+1}{d-x_2+1}$.

With the same change of variables as above, it suffices to prove that for any integers $n_1 \geq n_2 \geq 0$ and any reals $y_1, y_2 \in [0,1)$ (after regrouping the λ_2's):

$$-\frac{y_2}{n_2+y_2}\frac{n_1+2}{n_2+2} + \frac{(n_1+1)y_1}{n_1+y_1}\frac{n_1+2}{n_2+1} + \frac{n_1(1-y_1)}{n_1+y_1}\frac{n_1+1}{n_2+1} - \frac{n_1+y_1}{n_2+y_2} \leq 0. \quad (4.2)$$

By differentiating the left hand side of Equation (4.2) with respect to y_2, we obtain $-\frac{n_1+2}{n_2+2}\frac{n_2}{(n_2+y_2)^2} + \frac{n_1+y_1}{(n_2+y_2)^2}$, which is always ≥ 0. It is therefore sufficient to prove the above for the largest possible value of y_2. We consider two sub-cases.

First sub-case: $\lambda_1 > \lambda_2$ and $\lfloor x_1 \rfloor < \lfloor x_2 \rfloor$. In that situation the largest possible value for y_2 is ≤ 1. It therefore suffices to prove that

$$-\frac{n_1+2}{(n_2+1)(n_2+2)} + \frac{(n_1+1)y_1}{n_1+y_1}\frac{n_1+2}{n_2+1} + \frac{n_1(1-y_1)}{n_1+y_1}\frac{n_1+1}{n_2+1} - \frac{n_1+y_1}{n_2+1} \leq 0,$$

which is equivalent to (after simplification and multiplication by $\frac{n_2+1}{n_1+2}$):

$$-\frac{1}{n_2+2} + \frac{n_1+1}{n_1+2}\frac{n_1+2y_1}{n_1+y_1} - \frac{n_1+y_1}{n_1+2} \leq 0.$$

The latter increases with respect to n_2, so it suffices to prove it for $n_2 = n_1 - 1$. The numerator is $-(n_1+1)(y_1-\frac{1}{2})^2 + \frac{1-3n_1}{4} - y_1$, which is indeed ≤ 0.

Second sub-case: $\lambda_1 > \lambda_2$ and $\lfloor x_1 \rfloor = \lfloor x_2 \rfloor$. In that situation, we have $y_1 = 1 - x_1 + \lfloor x_1 \rfloor \geq 1 - x_2 + \lfloor x_2 \rfloor = y_2$. As the left hand side of Equation (4.2) increases with y_2, it suffices to prove it for $y_2 = y_1$, which means $x_2 = x_1$. In that situation, the result trivially holds. This completes the proof. □

4.2. Handling General Subsets of $[\![1,d]\!]$. We prove Theorem 4.1 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: we already have the result for some set $I \subseteq [\![v+1,d]\!]$ and add some vectors $\mathbf{b}_{u+1}, \ldots, \mathbf{b}_v$ to I. We make use of the local densities δ_i of the set I over small intervals $[\![\alpha_i+1, \alpha_{i+1}]\!]$. Note that in the lemma, the local densities δ_i are decreasing: this is required for applying Minkowski's theorem in the proof of Theorem 4.1.

LEMMA 4.6. *Let $v \in [\![2,d]\!]$, $I \subseteq [\![v+1,d]\!]$. Assume that there exist an integer $t \geq 1$ and some integers $v = \alpha_1 < \alpha_2 < \ldots < \alpha_t \leq d$ such that:*

$$\pi_I^{|I|} \geq \prod_{i=1}^{t-1} \left(\pi_{[\![\alpha_i+1,\alpha_{i+1}]\!]}^{|I_i|} \cdot \sqrt{d}^{|I_i| \log \delta_i} \right) \quad \text{and} \quad 1 \geq \delta_1 > \ldots > \delta_{t-1} > 0,$$

where $I_i = I \cap [\![\alpha_i+1, \alpha_{i+1}]\!]$ and $\delta_i = \frac{|I_i|}{\alpha_{i+1}-\alpha_i}$, for $i < t$.

Let $u \in [\![1, v-1]\!]$ and $I' = [\![u+1, v]\!] \cup I$. Then there exist an integer $t' \geq 1$ and some integers $0 = \alpha'_1 < \alpha'_2 < \ldots < \alpha'_{t'} \leq d$ such that:

$$\pi_{I'}^{|I'|} \geq \prod_{i=1}^{t'-1} \left(\pi_{[\![\alpha'_i+1,\alpha'_{i+1}]\!]}^{|I'_i|} \cdot \sqrt{d}^{|I'_i| \log \delta'_i} \right) \quad \text{and} \quad 1 \geq \delta'_1 > \ldots > \delta'_{t'-1} > 0,$$

where $I'_i = I' \cap [\![\alpha'_i+1, \alpha'_{i+1}]\!]$ and $\delta'_i = \frac{|I'_i|}{\alpha'_{i+1}-\alpha'_i}$, for $i < t'$.

Proof. Assume first that $\frac{v-u}{v} > \delta_1$, Then Lemmas 4.2 and 4.3 give

$$\pi_{I'}^{|I'|} = \pi_{[\![u+1,v]\!]}^{v-u} \cdot \pi_I^{|I|} \geq \pi_{[\![1,v]\!]}^{v-u} \cdot \sqrt{d}^{(v-u)\log\frac{v-u+1}{v+1}} \cdot \pi_I^{|I|} \geq \pi_{[\![1,v]\!]}^{v-u} \cdot \sqrt{d}^{(v-u)\log\frac{v-u}{v}} \cdot \pi_I^{|I|}.$$

It suffices to take $t' = t+1$, $\alpha'_1 = 0$, $\delta'_1 = \frac{v-u}{v}$, $\alpha'_k = \alpha_{k-1}$ and $\delta'_k = \delta_{k-1}$ for $k \geq 1$.

Otherwise, we let $\lambda_1 \in (0,u]$ 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:

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

Then we have, by using Lemma 4.5,

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

If $\frac{v-u+|I_1|}{\alpha_2} > \frac{|I_2|}{\alpha_3-\alpha_2}$, we use Lemma 4.5 to lower bound the above by

$$\left(\widetilde{\pi}_{[1,\alpha_2]}^{v-u+|I_1|} \cdot \sqrt{d}^{(v-u+|I_1|)\log\frac{v-u+|I_1|}{\alpha_2}}\right) \cdot \prod_{i=2}^{t-1} \left(\pi_{[\![\alpha_i+1,\alpha_{i+1}]\!]}^{|I_i|} \cdot \sqrt{d}^{|I_i|\log\delta_i}\right)$$

and we conclude as in the first step, putting $t' = t$, $\alpha'_1 = 0$, $\alpha'_k = \alpha_k$ 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:

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

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 4.5 to relate $\widetilde{\pi}_{[\lambda_1+1,\alpha_2]}$ to $\widetilde{\pi}_{[\lambda_2+1,\alpha_2]}$, leads to:

$$\begin{aligned}
\pi_{I'}^{|I'|} &\geq \left(\widetilde{\pi}_{[\lambda_2+1,\alpha_3]}^{v-u+|I\cap[\![\alpha_1+1,\alpha_3]\!]|} \cdot \sqrt{d}^{(v-u+|I\cap[\![\alpha_1+1,\alpha_3]\!]|)\log\frac{v-u+|I\cap[\![\alpha_1+1,\alpha_3]\!]|}{\alpha_3-\lambda_2}}\right) \\
&\quad \cdot \prod_{i=3}^{t-1} \left(\pi_{[\![\alpha_i+1,\alpha_{i+1}]\!]}^{|I_i|} \cdot \sqrt{d}^{|I_i|\log\delta_i}\right).
\end{aligned}$$

We can proceed in the same way, constructing $\lambda_2 > \lambda_3 > \ldots$. Suppose first that the construction stops at some point. After application of Lemma 4.5 to $\widetilde{\pi}_{[\lambda_k+1,\alpha_{k+1}]}$,

we have:

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

We can then conclude, by putting $t' = t-k+1$, $\alpha'_1 = 0, \alpha'_j = \alpha_{j+k-1}$ 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:

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

to which we can apply Lemma 4.5 to obtain $\pi_{I'}^{|I'|} \geq \pi_{[\![1,\alpha_t]\!]}^{|I'|} \cdot \sqrt{d}^{|I'| \log \frac{|I' \cap [\![1,\alpha_t]\!]|}{\alpha_t}}$, which is again in the desired form, with $t' = 2$, $\alpha'_1 = 0$, $\alpha'_2 = \alpha_t$, $\delta'_1 = \frac{|I' \cap [\![1,\alpha_t]\!]|}{\alpha_t}$. □

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

LEMMA 4.7. *Let $\Delta \geq 1$, and define $F_\Delta(k,d) = \Delta^{-k \log \frac{k}{d}}$. 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 $i \leq t$,*

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

Proof. Since the function $x \mapsto x \log x$ is convex on $[0, +\infty)$, 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 $\sum_{i \leq t} d_i$):

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

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

Theorem 4.1 now follows from successive applications of Lemma 4.6, as follows:

Proof of Theorem 4.1. Lemma 4.6 gives us, by induction on the size of the considered set I, that for all $I \subseteq [\![1,d]\!]$, 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 4.7 with $\Delta := \sqrt{d}, k_i := |I_i|$ and $d_i := \alpha_{i+1} - \alpha_i$, we 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 δ_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:

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

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

5. Worst-case HKZ-Reduced Bases. We now turn to the construction of worst-case inputs for Kannan's SVP algorithm, i.e., to the proof of the last assertion of Theorem 2.5. In view of the results of §3, it suffices to build HKZ-reduced bases $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ such that $\frac{(\|\mathbf{b}_1\|/\sqrt{d})^{d-i+1}}{\prod_{j \geq i} \|\mathbf{b}_j^*\|}$ is large, for an i such that for any $j \geq i$ we have $\|\mathbf{b}_j^*\| \leq \frac{2}{3} \frac{\|\mathbf{b}_1\|}{\sqrt{d}}$. To achieve that goal, we will build HKZ-reduced bases for which a certain number of Minkowski inequalities are simultaneously tight. More precisely, we will essentially have $\|\mathbf{b}_i^*\| \approx \sqrt{d-i+1} \left(\prod_{j=i}^{d} \|\mathbf{b}_j^*\|\right)^{\frac{1}{d-i+1}}$ for all i, where the $\approx$ symbol hides a constant. Our HKZ-reduced bases are arguably the least reduced possible, as their $\|\mathbf{b}_i^*\|$'s decrease as fast as allowed by the HKZ-reducedness assumption. Note that one can easily build bases with pre-determined values for the GSO quantities $\|\mathbf{b}_i^*\|$ and $\mu_{i,j}$ for $j < i \leq d$: consider the columns of the upper triangular matrix $B = (B)_{i,j}$ with $B_{i,i} = \|\mathbf{b}_i^*\|$ and $B_{i,j} = \mu_{j,i} B_{i,i}$. However, we also need the corresponding basis to be HKZ-reduced.

In this section, we first provide a sufficient condition on the sequence $(f(i))_{i \leq d}$ for an HKZ-reduced basis with $\|\mathbf{b}_i^*\| = f(i)$ to exist. This is a refinement of a probabilistic technique due to Ajtai [4, 5]. We then explicit a function f that satisfies that condition and which leads to HKZ-reduced bases of worst possible quality. For these bases, we will have $\frac{(\|\mathbf{b}_1\|/\sqrt{d})^{d-i+1}}{\prod_{j \geq i} \|\mathbf{b}_j^*\|} \geq 2^{O(d)} \cdot d^{\frac{d}{2\mathrm{e}}}$, for a valid i.

5.1. Ajtai's Sampling Revisited. Below is a general condition for an HKZ-reduced basis with prescribed $\|\mathbf{b}_i^*\|$'s to exist.

THEOREM 5.1. *Let $d > 0$ and $f : [\![1, d]\!] \to (0, +\infty)$. Assume that*

$$\forall j \leq d, \ \sum_{i=1}^{j-1} \left(\frac{j-i}{2\pi\mathrm{e}}\right)^{-\frac{j-i}{2}} \left(1 - \left(\frac{f(j)}{f(i)}\right)^2\right)_+^{\frac{j-i}{2}} \left(\prod_{k=i}^{j} \frac{f(i)}{f(k)}\right) < 1,$$

where $(x)_+$ *denotes* $\max(0,x)$. *Then there exists an HKZ-reduced basis* $(\mathbf{b}_1,\ldots,\mathbf{b}_d)$ *with* $\|\mathbf{b}_i^*\| = f(i)$.

The condition above might seem intricate at first glance, though it is in fact fairly natural. The term $(j-i)^{-\frac{j-i}{2}}\prod_{k=i}^{j}\frac{f(i)}{f(k)}$ resembles Minkowski's inequality. It is natural that it should occur for all (i,j), since for an HKZ-reduced basis Minkowski's inequality is satisfied for all bases $(\mathbf{b}_i^{(i)},\ldots,\mathbf{b}_j^{(i)})$, where $\mathbf{b}_l^{(k)}$ denotes the projection of the vector $\mathbf{b}_l$ orthogonally to the linear span of the vectors $\mathbf{b}_1,\ldots,\mathbf{b}_{k-1}$. Said differently, the following is a necessary condition for a basis to be HKZ-reduced:

$$\forall j \le d,\ \sum_{i=1}^{j-1}(4\gamma_{j-i+1})^{-\frac{j-i}{2}}\left(1-\left(\frac{f(j)}{f(i)}\right)^2\right)_+^{\frac{j-i}{2}}\left(\prod_{k=i}^{j}\frac{f(i)}{f(k)}\right) < 1.$$

This is merely a restatement of the fact that, since Minkowski's inequality is verified for any pair (i,j), the i-th term is at most $2^{-(j-i)}$, so that the sum is < 1. Since asymptotically we have $\gamma_d \le d(\frac{1.744}{2\pi\mathrm{e}}+o(1))$ (see [18, Ch. 1]), we see that the condition of Theorem 5.1 is not far from optimal.

Lemma 5.2 is the core of the proof of Theorem 5.1. It bounds the probability that when a random basis $(\mathbf{b}_1,\ldots,\mathbf{b}_d)$ is built appropriately, any lattice vector $\sum_i x_i\mathbf{b}_i$ with $x_d \neq 0$ will be longer than $\mathbf{b}_1$.

LEMMA 5.2. *Let* $(\mathbf{b}_1,\ldots,\mathbf{b}_{d-1})$ *be a lattice basis and let* $\mathbf{b}_d$ *be a random vector. We suppose that:*

(i) *For any* $i \le d$, *we have* $\|\mathbf{b}_i^*\| = f(i)$.

(ii) *The* $\mu_{d,i}$*'s for* $i < d$ *are independently and uniformly distributed in* $[-\frac{1}{2},\frac{1}{2}]$.

The probability that there exists $\mathbf{x}\in\mathbb{Z}^d$ *with* $x_d \neq 0$ *and* $\|\sum_i x_i\mathbf{b}_i\| \le \|\mathbf{b}_1\|$ *is upper bounded by*

$$\left(\frac{d-1}{2\pi\mathrm{e}}\right)^{-\frac{d-1}{2}}\left(1-\left(\frac{f(d)}{f(1)}\right)^2\right)_+^{\frac{d-1}{2}}\left(\prod_{i\le d}\frac{f(1)}{f(i)}\right).$$

Proof. We write $\sum_{i\le d}x_i\mathbf{b}_i$ as $\sum_{i\le d}\left(x_i+\sum_{j=i+1}^{d}\mu_{j,i}x_j\right)\mathbf{b}_i^*$ and define $u_i = x_i + \left\lfloor\sum_{j=i+1}^{d}\mu_{j,i}x_j\right\rceil$ and $\delta_i = \left\{\sum_{j=i+1}^{d}\mu_{j,i}x_j\right\}$, for $i \le d$. If $i < d$, then $\delta_i = \left\{\mu_{d,i}x_d + \sum_{j=i+1}^{d-1}\mu_{j,i}x_j\right\}$ contains a random term $(\mu_{d,i}x_d)$ and a constant term $(\sum_{j=i+1}^{d-1}\mu_{j,i}x_j)$. Since $x_d \neq 0$ and since the $\mu_{d,i}$'s are distributed independently and uniformly in $[-1/2,1/2]$, the same holds for the δ_i's (for each fixed choice of $\mathbf{x}$).

The event under scope can be rewritten as

$$\exists u_d\in\mathbb{Z}\setminus 0,\ \exists(u_1,\ldots,u_{d-1})\in\mathbb{Z}^{d-1},\ \sum_{i<d}(u_i+\delta_i)^2f(i)^2 \le f(1)^2 - u_d^2f(d)^2.$$

Let p be its probability. If $f(1)^2 - u_d^2f(d)^2 < 0$, then $p = 0$. We thus have:

$$p \le \sum_{u_d\in\mathbb{Z}\cap\left[1,\frac{f(1)}{f(d)}\right]}\ \sum_{(u_1,\ldots,u_{d-1})\in\mathbb{Z}^{d-1}}\Pr\left(\sum_{i<d}(u_i+\delta_i)^2f(i)^2 \le f(1)^2 - u_d^2f(d)^2\right).$$

Let $c > 0$ be an arbitrary constant. We can bound the summand by

$$\int_{\delta\in\left[-\frac{1}{2},\frac{1}{2}\right]^{d-1}}\exp\left(c - c\frac{\sum_{i<d}(u_i+\delta_i)^2f(i)^2}{f(1)^2-u_d^2f(d)^2}\right)\mathrm{d}\delta.$$

By summing over the u_i's for $i < d$, we obtain

$$\begin{aligned}
\sum_{\mathbf{u}\in\mathbb{Z}^{d-1}} \int_{\delta\in\left[-\frac{1}{2},\frac{1}{2}\right]^{d-1}} \exp\left(c - c\frac{\sum_{i<d}(u_i+\delta_i)^2 f(i)^2}{f(1)^2 - u_d^2 f(d)^2}\right) \mathrm{d}\delta \\
&= \int_{\mathbb{R}^{d-1}} \exp\left(c - c\frac{\sum_{i<d}\delta_i^2 f(i)^2}{f(1)^2 - u_d^2 f(d)^2}\right) \mathrm{d}\delta \\
&= \mathrm{e}^c \prod_{i<d} \int_{\mathbb{R}} \exp\left(-c\frac{\delta_i^2 f(i)^2}{f(1)^2 - u_d^2 f(d)^2}\right) \mathrm{d}\delta_i \\
&= \mathrm{e}^c \left(\frac{\pi}{c}\right)^{\frac{d-1}{2}} \left(1 - \left(\frac{u_d f(d)}{f(1)}\right)^2\right)^{\frac{d-1}{2}} \prod_{i<d} \frac{f(1)}{f(i)}.
\end{aligned}$$

By taking $c = \frac{d-1}{2}$ and considering all possible u_d's, we obtain:

$$p \leq \frac{f(1)}{f(d)} \cdot \left(\frac{2\pi\mathrm{e}}{d-1}\right)^{\frac{d-1}{2}} \left(1 - \left(\frac{f(d)}{f(1)}\right)^2\right)_+^{\frac{d-1}{2}} \prod_{i<d} \frac{f(1)}{f(i)}.$$

□

We now prove Theorem 5.1. We build the basis iteratively, starting with $\mathbf{b}_1$, chosen arbitrarily with $\|\mathbf{b}_1\| = f(1)$. Assume now that $\mathbf{b}_1, \ldots, \mathbf{b}_{j-1}$ have already been chosen with $\|\mathbf{b}_i^*\| = f(i)$ for $i < j$ and that they are HKZ-reduced. We choose $\mathbf{b}_j$ as $\mathbf{b}_j^* + \sum_{k<j} \mu_{j,k}\mathbf{b}_k^*$ such that $\|\mathbf{b}_j^*\| = f(j)$ and the random variables $(\mu_{j,k})_{k<j}$ are chosen uniformly and independently in $[-\frac{1}{2}, \frac{1}{2}]$. Let $p_{i,j}$ be the probability that the vector $\mathbf{b}_i^*$ is not a shortest non-zero vector in $L(\mathbf{b}_i^{(i)}, \ldots, \mathbf{b}_j^{(i)})$, for $i < j$. This means that there exists an integral vector $\mathbf{x}$ such that $\|\sum_{k=i}^{j} x_k \mathbf{b}_k^{(i)}\| < \|\mathbf{b}_i^*\|$. Since $(\mathbf{b}_1, \ldots, \mathbf{b}_{j-1})$ is HKZ-reduced, so is the basis $(\mathbf{b}_i^{(i)}, \ldots, \mathbf{b}_{j-1}^{(i)})$, and thus we must have $x_j \neq 0$. Lemma 5.2 gives us

$$p_{i,j} \leq \left(\frac{j-i}{2\pi\mathrm{e}}\right)^{-\frac{j-i}{2}} \left(1 - \left(\frac{f(j)}{f(i)}\right)^2\right)_+^{\frac{j-i}{2}} \left(\prod_{k=i}^{j} \frac{f(i)}{f(k)}\right).$$

We conclude the proof by observing that the probability of non-HKZ-reducedness of $(\mathbf{b}_1, \ldots, \mathbf{b}_j)$ is at most $\sum_{i<j} p_{i,j}$. By hypothesis, this quantity is < 1. Overall, this means that there exist $\mu_{i,j}$'s such that $(\mathbf{b}_1, \ldots, \mathbf{b}_j)$ is HKZ-reduced. □

5.2. The GSO of Worst-Case HKZ-reduced Bases. This section is devoted to the construction of a function f satisfying the conditions of Theorem 5.1 as tightly as possible. In order to make explicit the fact that f depends on the dimension d, we shall write f_d instead of f. Note that although $f(i)$ will depend on d, this will not be the case for $f(d-i)$. Suppose that the basis $(\mathbf{b}_i)_i$ is HKZ-reduced. Then f_d must satisfy the Minkowski inequalities:

$$\forall i < j,\ f_d(i) \leq \sqrt{\gamma_{j-i+1}} \cdot \left(\prod_{k=i}^{j} f_d(k)\right)^{\frac{1}{j-i+1}}.$$

We choose f_d according to the strongest of those conditions, i.e., with $j = d$. It is known (see [58] for example) that this set of conditions does not suffice for an

HKZ-reduced basis to exist. We thus expect to have to relax these constraints. We will also replace the Hermite constant by a more explicit term. For these reasons, we introduce

$$f_{\psi,d}(i) = \sqrt{\psi(d-i+1)} \cdot \left(\prod_{k=i}^{d} f_{\psi,d}(k) \right)^{\frac{1}{d-i+1}},$$

where ψ is to be chosen in the sequel. This equation uniquely defines $f_{\psi,d}(i)$ for all i once we set $f_{\psi,d}(d) = 1$, as implied by the following result.

LEMMA 5.3. *The following holds for any $i \leq j \leq d$:*

$$\frac{f_{\psi,d}(i)}{f_{\psi,d}(j)} = \sqrt{\frac{\psi(d-i+1)}{\psi(d-j+1)}} \cdot \prod_{k=i}^{j-1} \psi(d-k+1)^{\frac{1}{2(d-k)}}.$$

Proof. By taking the quotient between $f_{\psi,d}(i)^{d-i+1}$ and $f_{\psi,d}(i+1)^{d-i}$, we get

$$\frac{f_{\psi,d}(i)}{f_{\psi,d}(i+1)} = \sqrt{\frac{\psi(d-i+1)}{\psi(d-i)}} \cdot \psi(d-i+1)^{\frac{1}{2(d-i)}}.$$

The lemma follows. □

In the next subsection, we will prove the following theorem.

THEOREM 5.4. *Let $\psi(x) = Cx$ with $C = \exp(-6)$. Then, for all $i < j$, we have*

$$(j-i+1)^{-\frac{j-i}{2}} \left(1 - \left(\frac{f_{\psi,d}(j)}{f_{\psi,d}(i)}\right)^2\right)_+^{\frac{j-i}{2}} \left(\prod_{k=i}^{j} \frac{f_{\psi,d}(i)}{f_{\psi,d}(k)}\right) \leq \left(2\pi\mathrm{e}(\sqrt{\mathrm{e}}+1)^2\right)^{-\frac{j-i}{2}}.$$

Thanks to Theorem 5.1, we obtain the following.

COROLLARY 5.5. *Let ψ be as in Theorem 5.4. There exist HKZ-reduced bases $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ with*

$$\|\mathbf{b}_i^*\| = f_{\psi,d}(i) = \sqrt{d-i+1} \cdot \prod_{k=i}^{d-1} \left(C(d-k+1)\right)^{\frac{1}{2(d-k)}}.$$

Moreover, when $d-i$ grows to infinity, we have

$$\|\mathbf{b}_i^*\| = \exp\left(\frac{\log^2(d-i+1)}{4} + \frac{1+\log C}{2}\log(d-i+1) + O(1)\right).$$

Proof. Let $j \leq d$. Thanks to Theorem 5.4, we have

$$\begin{aligned}\sum_{i=1}^{j-1} \left(\frac{j-i}{2\pi\mathrm{e}}\right)^{-\frac{j-i}{2}} & \left(1 - \left(\frac{f(j)}{f(i)}\right)^2\right)_+^{\frac{j-i}{2}} \left(\prod_{k=i}^{j} \frac{f(i)}{f(k)}\right) \\ &\leq \sum_{i=1}^{j-1} \left(\frac{j-i+1}{j-i}\right)^{\frac{j-i}{2}} \left(\sqrt{\mathrm{e}}+1\right)^{-(j-i)} \\ &< \sqrt{\mathrm{e}} \cdot \sum_{i \geq 1} (\sqrt{\mathrm{e}}+1)^{-i} = 1.\end{aligned}$$

The first part of the result follows from Theorem 5.1 and Lemma 5.3. For the second part, note that our choice of ψ gives

$$2\log f_{\psi,d}(i) = \log(d-i+1) + \sum_{k=i}^{d-1} \frac{\log C + \log(d-k+1)}{d-k}.$$

Suppose that $d-i \to +\infty$. The quantity $\left|\sum_{k=i}^{d-1} \frac{\log(d-k+1)}{d-k} - \int_i^d \frac{\log(d-x+1)}{d-x+1}\,\mathrm{d}x\right|$ is

$$\begin{aligned}
&\leq \left|\sum_{k=i}^{d-1} \frac{\log(d-k+1)}{d-k+1} - \int_i^d \frac{\log(d-x+1)}{d-x+1}\,\mathrm{d}x\right| + \sum_{k=i}^{d-1} \frac{\log(d-k+1)}{(d-k)^2} \\
&\leq O(1) + \sum_{k=i}^{d-1} \int_k^{k+1} \left|\frac{\log(d-k+1)}{d-k+1} - \frac{\log(d-x+1)}{d-x+1}\right| \mathrm{d}x \\
&\leq O(1) + \sum_{k=i}^{d-1} \max_{x\in[k,k+1]} \frac{|1-\log(d-x+1)|}{(d-x+1)^2} \;=\; O(1).
\end{aligned}$$

Classically, we also have $\left|\sum_{k=i}^{d-1} \frac{1}{d-k} - \log(d-i)\right| = O(1)$. The result follows from the fact that $\int_i^d \frac{\log(d-x+1)}{d-x+1}\,\mathrm{d}x = \frac{\log^2(d-i+1)}{2}$. □

We can now prove the remaining assertion of Theorem 2.5. To do that, we consider an HKZ-reduced basis as in Corollary 5.5, and try to apply Lemmas 3.1, 3.2 and 3.3. For Lemma 3.3, we let $B = \frac{4}{9}\|\mathbf{b}_1\|^2$ and $i = \left\lfloor d\left(1-\frac{1}{\mathrm{e}}\right) + \alpha\frac{d}{\log d}\right\rceil$, for some constant α to be fixed later.

LEMMA 5.6. *There exists an α such that when d is large enough, we have $\|\mathbf{b}_j^*\| \leq \frac{2}{3}\frac{\|\mathbf{b}_1\|}{\sqrt{d}}$, for all $j \geq i$. Furthermore, we have $\prod_{j\geq i} \frac{2}{3}\frac{\|\mathbf{b}_1\|}{\sqrt{d}\|\mathbf{b}_j^*\|} \geq 2^{O(d)} \cdot d^{\frac{d}{2\mathrm{e}}}$.*

Proof. Since $d-i \to +\infty$, Corollary 5.5 implies that:

$$\begin{aligned}
2\log\frac{\|\mathbf{b}_i^*\|}{\|\mathbf{b}_1\|} &= \frac{\log^2(d-i+1) - \log^2 d}{2} + (1+\log C)\left(\log(d-i+1) - \log d\right) + O(1) \\
&\leq \log\left(\frac{d-i+1}{d}\right)(\log d + 1 + \log C) + O(1) \\
&\leq \log\left(\frac{1}{\mathrm{e}} - \frac{\alpha}{\log d} + O\left(\frac{1}{d}\right)\right)(\log d + 1 + \log C) + O(1) \\
&\leq -\log d - \alpha\mathrm{e}\left(1 + \frac{1+\log C}{\log d}\right) + O(1),
\end{aligned}$$

where the $O(1)$ constant does not depend on α. We choose α so that the result holds for $j = i$.

Furthermore, Lemma 5.3 provides $\frac{\|\mathbf{b}_{j-1}^*\|}{\|\mathbf{b}_j^*\|} = \sqrt{1+\frac{1}{d-j+1}}\,(C(d-j+2))^{\frac{1}{2(d-j+1)}}$. This implies that for any $j \leq d+2-\frac{1}{C}$, we have $\|\mathbf{b}_j^*\| \leq \|\mathbf{b}_{j-1}^*\| \leq \ldots \leq \|\mathbf{b}_i^*\| \leq \frac{2}{3}\frac{\|\mathbf{b}_1\|}{\sqrt{d}}$. Assume now that $j \in [d+2-\frac{1}{C}, d]$. Then the explicit formula for $\|\mathbf{b}_j^*\|$ given in Corollary 5.5 implies that $\|\mathbf{b}_j^*\| = O(1)$. As $\|\mathbf{b}_1\| = \exp\left(\frac{\log^2 d}{4}(1+o(1))\right)$, when d is large enough we have $\|\mathbf{b}_j^*\| \leq \frac{2}{3}\frac{\|\mathbf{b}_1\|}{\sqrt{d}}$ for all $j \geq i$.

We now prove the second assertion of the result. By definition of $f_{\psi,d}$, we have $\prod_{j\geq i}\frac{2}{3}\frac{\|\mathbf{b}_1\|}{\sqrt{d}\|\mathbf{b}_j^*\|} = 2^{O(d)}\left(\sqrt{\frac{d-i+1}{d}}\frac{\|\mathbf{b}_1\|}{\|\mathbf{b}_i^*\|}\right)^{d-i+1}$. Finally, for our value of i, we have $\left(\frac{\sqrt{d-i+1}}{\sqrt{d}}\right)^{d-i+1} = 2^{O(d)}$ and $\left(\frac{\|\mathbf{b}_1\|}{\|\mathbf{b}_i^*\|}\right)^{d-i+1} \geq 2^{O(d)}\cdot d^{\frac{d}{2\mathrm{e}}}$. □

5.3. Sketch of the Proof of Theorem 5.4. As $\exp(5) > 2\pi\mathrm{e}(\sqrt{\mathrm{e}}+1)^2$, it suffices to prove that for all $i<j$, we have:

$$(j-i+1)^{-\frac{j-i}{2}}\cdot T_1\cdot T_2 \leq \exp\left(-\frac{5}{2}(j-i)\right),$$

where $T_1 = \left(1-\left(\frac{f_{\psi,d}(j)}{f_{\psi,d}(i)}\right)^2\right)_+^{\frac{j-i}{2}}$ and $T_2 = \prod_{k=i}^{j}\frac{f_{\psi,d}(i)}{f_{\psi,d}(k)}$.

The proof follows from elementary (though technical) analytical considerations. Let us write $a = d-i+1$ and $b = d-j+1$. This change of variables makes the problem independent of d. The domain of valid pairs (a,b) is $1\leq b<a\leq d$. Note that if $b=1$, then we can bound T_1 by 1 and use the definition of $f_{\psi,d}$ to obtain the sufficient condition:

$$\sqrt{a}\cdot\exp(-3a) \leq \exp\left(-\frac{5}{2}(a-1)\right),$$

which is valid for all a. We now assume that $a>b>1$. Our proof is made of three main steps. In the first step, we try to obtain the result without the first term, i.e., while bounding T_1 by 1. We reach this goal for $a\geq 158000$ and $b\leq a-\frac{1.65}{\log^3 a}$. In the second step, we use T_1 to obtain the result for $a\geq 158000$ and $b\geq a-\frac{1.65}{\log^3 a}$. Finally, we prove the result for $1<b<a\leq 158000$ by exhaustively checking the inequality.

We start by simplifying T_1 and T_2. Lemma 5.3 implies that

$$T_1 = \left(1-\frac{\psi(b)}{\psi(a)}\prod_{k=b}^{a-1}\psi(k+1)^{-\frac{1}{k}}\right)_+^{\frac{a-b}{2}}.$$

The following lemma allows us to simplify the expression of the term T_2.

LEMMA 5.7. *We have* $T_2 = \left(\prod_{k=b}^{a-1}\frac{\psi(a)\psi(k+1)}{\psi(k)\psi(k+1)^{\frac{b-1}{k}}}\right)^{\frac{1}{2}}$.

Proof. We have

$$T_2 = \prod_{k=i}^{j}\frac{f_{\psi,d}(i)}{f_{\psi,d}(k)} = \left(\prod_{k=i}^{d}\frac{f_{\psi,d}(i)}{f_{\psi,d}(k)}\right)\cdot\left(\prod_{k=j}^{d}\frac{f_{\psi,d}(j)}{f_{\psi,d}(k)}\right)^{-1}\cdot\left(\frac{f_{\psi,d}(i)}{f_{\psi,d}(j)}\right)^{d-j}.$$

The first two terms can be simplified by using the definition of $f_{\psi,d}$, and the last one has been studied in Lemma 5.3. We get:

$$\begin{aligned}T_2 &= \frac{\psi(d-i+1)^{\frac{d-i+1}{2}}}{\psi(d-j+1)^{\frac{d-j+1}{2}}}\cdot\left(\frac{\psi(d-i+1)}{\psi(d-j+1)}\right)^{\frac{j-d}{2}}\cdot\left(\prod_{k=i}^{j-1}\psi(d-k+1)^{\frac{j-d}{2(d-k)}}\right)\\ &= \left(\prod_{k=i}^{j-1}\frac{\psi(d-i+1)\psi(d-k+1)}{\psi(d-k)\psi(d-k+1)^{\frac{d-j}{(d-k)}}}\right)^{\frac{1}{2}},\end{aligned}$$

as claimed. □

Now that both T_1 and T_2 have been expressed with the new variables a and b, the proof of Theorem 5.4 reduces to a study of a function of two variables. It is given in appendix.

6. Concluding Remarks. We have presented a complete worst-case analysis of Kannan's algorithm for the shortest vector problem. This analysis however leaves a few questions unanswered and raises several other significant questions.

On the complexity upper bound. Our analysis gives a complexity upper bound of $2^{O(d)} \cdot d^{\frac{d}{2}}$ for CVP, whereas we are unable to get a lower bound better than $2^{O(d)} \cdot d^{\frac{d}{2e}}$. This is related to the fact that we have only a poor estimate on the covering radius of the lattice (the value of A which we have to use to guarantee that we shall find a vector during the enumeration), hence the upper bound. Obtaining matching upper and lower bounds seems to require a deep understanding of the relationship between the geometry of the HKZ-reduced bases of the lattice and the covering radius: one would have to prove that the larger the covering radius, the better the basis. Besides, it should be noted that Banaszczyk's transference bound [10] $\mu(L)\lambda_1(L^*) \leq d$ implies that as soon as $\lambda_1(L^*) \approx \sqrt{d}(\det L)^{-1/d}$ (i.e., Minkowski's bound is essentially sharp for L^*), we have $\mu(L) \approx \lambda_1(L)$ and CVP can be solved in time $2^{O(d)} \cdot d^{\frac{d}{2e}}$, up to polynomial factors. Here $\mu(L)$ denotes the covering radius of L and L^* denotes the dual of L, i.e., the set of points in the span of L that have integral inner product with all vectors of L. Overal, this suggests that for almost all lattices as $d \to \infty$ (with the measure defined in [68]), Kannan's algorithm solves CVP in time at most $2^{O(d)} \cdot d^{\frac{d}{2e}}$. However, this leaves open the question of its worst-case complexity.

On the complexity lower bound. Though proved for real lattices, the complexity lower bound can most likely be extended to rational lattices (sublattices of $\mathbb{Q}^n$), by replacing integrals with discrete sums in our derivation of Lemma 5.2, thus leading a very similar criterion.

Average-case analysis. Our analysis leaves open the question of the average geometry of an HKZ-reduced basis, i.e., of the geometry of the almost always well-defined HKZ-reduced bases of random lattices. It is our belief that this geometry matches the worst case, i.e., that the norms of the Gram-Schmidt vectors still behave like $\|\mathbf{b}_i^*\| \approx \exp\left(-\left(\frac{1}{4} + o(1)\right)\log(d+1-i)^2\right)\|\mathbf{b}_1\|$. Such a result would allow one to prove that the average complexity of Kannan's algorithm is $2^{O(d)} \cdot d^{\frac{d}{2e}}$. Some authors favor the hypothesis that the average behaviour of an HKZ-reduced basis is rather a geometric decrease of the $\|\mathbf{b}_i^*\|$'s, i.e., roughly $\|\mathbf{b}_i^*\| \approx d^{-\frac{i}{d}}\|\mathbf{b}_1\|$. With such a basis, solving SVP by Kannan's algorithm would have a $2^{O(d)} \cdot d^{\frac{d}{8}}$ complexity.

Preprocessing the basis. Even if it turns out that HKZ bases do not behave that nicely, the question of whether such a basis exists for all lattices is of equal interest: this is related to the question of the optimal preprocessing for enumeration algorithm. Kannan chose HKZ, the main feature being that this is a strong reduction which can be embedded within an SVP computation at negligible cost, but it is not clear whether HKZ-reduction is the best choice with respect to enumeration. Geometrically decreasing bases indeed appear better. A plausible way to build them would be to consider bases that minimize the $d/2$-dimensional volume of the sublattice $(\mathbf{b}_1, \ldots, \mathbf{b}_{d/2})$ (and so on recursively). However, computing such a basis seems to require a huge amount of time, which makes its use limited for enumeration algorithms. Note finally that lower bounds on generalized Hermite's constants [13] strongly suggest that $d^{\frac{d}{8}+o(d)}$ is the limit for enumeration techniques, at least for a subset of lattices

of asymptotic probability 1 (as $d \to \infty$). In short, it seems that the enumeration techniques are bound to remain of superexponential complexity.

Acknowledgments. This work was initiated during the July 2007 seminar "Explicit methods in Number Theory" at Mathematisches Forschungsinstitut Oberwolfach. The authors are grateful to the MFO for the great working conditions provided on this occasion. The authors would also like to thank Jacques Martinet for the interest he showed for a preliminary version of those results and for pointing [58], Claus-Peter Schnorr for having pointed out several errors in earlier versions of this paper, and Cong Ling, Phong Nguyen, Xavier Pujol and Antonio Vera for several discussions.

Part of this work was undergone while the first author was employed by the INRIA and was working in the LORIA laboratory, and while the second author was working at the LIP laboratory of the École Normale Supérieure of Lyon.

REFERENCES

[1] K. Aardal and F. Eisenbrand, *The LLL algorithm and integer programming*. In *The LLL algorithm*, P. Q. Nguyen and B. Vallée (eds), 2009.

[2] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, *Closest point search in lattices*, IEEE Transactions on Information Theory, 48 (2002), pp. 2201–2214.

[3] 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), ACM Press, 1998, pp. 284–293.

[4] ———, *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), ACM Press, 2003, pp. 396–406.

[5] ———, *Optimal lower bounds for the Korkine-Zolotareff parameters of a lattice and for Schnorr's algorithm for the shortest vector problem*, Theory of Computing, 4 (2008), pp. 21–51.

[6] 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), ACM Press, 1997, pp. 284–293.

[7] 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), ACM Press, 2001, pp. 601–610.

[8] ———, *Sampling short lattice vectors and the closest lattice vector problem*, in Proceedings of the 17th Annual IEEE Conference on Computational Complexity (CCC 17), 2002, pp. 53–57.

[9] L. Babai, *On Lovász lattice reduction and the nearest lattice point problem*, Combinatorica, 6 (1986), pp. 1–13.

[10] W. Banaszczyk, *New bounds in some transference theorems in the geometry of numbers*, Mathematische Annalen, 296 (1993), pp. 625–635.

[11] J. Blömer, *Closest vectors, successive minima and dual-HKZ bases of lattices*, in Proceedings of the 2000 International Colloquium on Automata, Languages and Programming (ICALP 2000), vol. 1853 of Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 248–259.

[12] J. Blömer and S. Naewe, *Sampling methods for shortest vectors, closest vectors and successive minima*, Theoretical Computer Science, 410 (2009), pp. 1648–1665.

[13] M. I. Boguslavsky, *Radon transforms and packings*, Discrete Applied Mathematics, 111 (2001), pp. 3–22.

[14] W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system. I. The user language.*, Journal of Symbolic Computation, 24 (1997), pp. 235–265.

[15] H. Brönnimann, G. Melquiond, and S. Pion, *The design of the Boost interval arithmetic library*, Theoretical Computer Science, 351 (2006), pp. 111–118.

[16] D. Cadé, X. Pujol, and D. Stehlé, *fplll-3.0, a floating-point LLL implementation*. Available at `http://perso.ens-lyon.fr/damien.stehle#software`.

[17] H. Cohen, *A Course in Computational Algebraic Number Theory, 2nd edition*, Springer-Verlag, 1995.

[18] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, Springer-Verlag, 1988.

[19] *CRLibm, a library of correctly rounded elementary functions in double-precision.* `http://lipforge.ens-lyon.fr/www/crlibm/`.

[20] I. Dinur, *Approximating SVP_∞ to within almost-polynomial factors is NP-hard*, Theoretical Computer Science, 285 (2002), pp. 55–71.

[21] I. Dinur, G. Kindler, R. Raz, and S. Safra, *Approximating CVP to within almost-polynomial factors is NP-hard*, Combinatorica, 23 (2003), pp. 205–243.

[22] I. Dinur, G. Kindler, and S. Safra, *Approximating CVP to within almost polynomial factors is NP-hard*, in Proceedings of the 1998 Symposium on Foundations of Computer Science (FOCS 1998), IEEE Computer Society Press, 1998, pp. 99–109.

[23] P. van Emde Boas, *Another NP-complete partition problem and the complexity of computing short vectors in a lattice.* Technical report 81-04, Mathematisch Instituut, Universiteit van Amsterdam, 1981.

[24] U. Feige and D. Micciancio, *The inapproximability of lattice and coding problems with preprocessing*, Journal of Computer and System Sciences, 69 (2004), pp. 45–67.

[25] U. Fincke and M. Pohst, *A procedure for determining algebraic integers of given norm*, in Proceedings of EUROCAL, vol. 162 of Lecture Notes in Computer Science, Springer-Verlag, 1983, pp. 194–202.

[26] N. Gama, N. Howgrave-Graham, H. Koy, and P. Nguyen, *Rankin's constant and blockwise lattice reduction*, in Proceedings of Crypto 2006, no. 4117 in Lecture Notes in Computer Science, Springer-Verlag, 2006, pp. 112–130.

[27] N. Gama and P. Q. Nguyen, *Finding short lattice vectors within Mordell's inequality*, in Proceedings of the 40th Symposium on the Theory of Computing (STOC 2008), ACM Press, 2008.

[28] N. Gama, P. Q. Nguyen, and O. Regev, *Lattice enumeration using extreme pruning*, 2010. To appear in the proceedings of Eurocrypt 2010.

[29] O. Goldreich, S. Goldwasser, and S. Halevi, *Public-key cryptosystems from lattice reduction problems*, in Proceedings of Crypto 1997, vol. 1294 of Lecture Notes in Computer Science, Springer-Verlag, 1997, pp. 112–131.

[30] G. Hanrot and D. Stehlé, *Improved analysis of Kannan's shortest lattice vector algorithm (extended abstract)*, in Proceedings of Crypto 2007, vol. 4622 of Lecture Notes in Computer Science, Springer-Verlag, 2007, pp. 170–186.

[31] ———, *Worst-Case Hermite-Korkine-Zolotarev Reduced Lattice Bases*, Research Report RR-6422, INRIA, 2008.

[32] A. Hassibi and S. Boyd, *Integer parameter estimation in linear models with applications to GPS*, IEEE Transactions on Signal Processing, 46 (1998), pp. 2938–2952.

[33] I. Haviv and O. Regev, *Tensor-based hardness of the shortest vector problem to within almost polynomial factors*, in Proceedings of the 39th Symposium on the Theory of Computing (STOC 2007), ACM Press, 2007, pp. 469–477.

[34] B. Helfrich, *Algorithms to construct Minkowski reduced and Hermite reduced lattice bases*, Theoretical Computer Science, 41 (1985), pp. 125–139.

[35] C. Hermite, *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 (1850), pp. 279–290. Also available in *Œuvres de Charles Hermite*, by É. Picard, Gauthiers-Villars, Paris, 1905.

[36] 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), vol. 1423 of Lecture Notes in Computer Science, Springer-Verlag, 1998, pp. 267–288.

[37] R. Kannan, *Improved algorithms for integer programming and related lattice problems*, in Proceedings of the 15th Symposium on the Theory of Computing (STOC 1983), ACM Press, 1983, pp. 99–108.

[38] ———, *Algorithmic geometry of numbers*, Annual Review of Computer Science, 2 (1987), pp. 231–267.

[39] S. Khot, *Hardness of approximating the shortest vector problem in lattices*, in Proceedings of the 2004 Symposium on Foundations of Computer Science (FOCS 2004), IEEE Computer Society Press, 2004, pp. 126–135.

[40] A. Korkine and G. Zolotarev, *Sur les formes quadratiques*, Mathematische Annalen, 6 (1873), pp. 336–389.

[41] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, *Factoring polynomials with rational coefficients*, Mathematische Annalen, 261 (1982), pp. 513–534.

[42] H. W. Lenstra, Jr., *Integer programming with a fixed number of variables*, Mathematics of Operations Research, 8 (1983), pp. 538–548.

[43] C. Ling, *On the proximity factors of lattice reduction-aided decoding*. Submitted, available at `http://www.commsp.ee.ic.ac.uk/~cling`, 2008.
[44] J. Martinet, *Perfect Lattices in Euclidean Spaces*, Springer-Verlag, 2002.
[45] J. Mazo and A. Odlyzko, *Lattice points in high-dimensional spheres*, Monatshefte für Mathematik, 110 (1990), pp. 47–61.
[46] R. Merkle and M. Hellman, *Hiding information and signatures in trapdoor knapsacks*, IEEE Transactions on Information Theory, 24 (1978), pp. 525–530.
[47] D. Micciancio, *The hardness of the closest vector problem with preprocessing*, IEEE Transactions on Information Theory, 47 (2001), pp. 1212–1215.
[48] ———, *The shortest vector problem is NP-hard to approximate to within some constant*, SIAM Journal on Computing, 30 (2001), pp. 2008–2035.
[49] D. Micciancio and S. Goldwasser, *Complexity of lattice problems: a cryptographic perspective*, Kluwer Academic Press, 2002.
[50] D. Micciancio and O. Regev, *Lattice-based cryptography*, in Proceedings of Post-quantum Cryptography (PQC'08), Springer-Verlag, ed., 2008.
[51] D. Micciancio and P. Voulgaris, *A deterministic single exponential time algorithm for most lattice p roblems based on Voronoi cell computations*, 2010. To appear in the proceedings of STOC 2010.
[52] ———, *Faster exponential time algorithms for the shortest vector problem*, in Proc. of SODA, SIAM Publications, 2010, pp. 1468–1480.
[53] H. Minkowski, *Geometrie der Zahlen*, Teubner-Verlag, 1896.
[54] W. H. Mow, *Maximum likelihood sequence estimation from the lattice viewpoint*, IEEE Transactions on Information Theory, 40 (1994), pp. 1591–1600.
[55] P. Q. Nguyen and J. Stern, *Adapting density attacks to low-weight knapsacks*, in Proceedings of Asiacrypt 2005, vol. 3788 of Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 41–58.
[56] P. Q. Nguyen and T. Vidick, *Sieve algorithms for the shortest vector problem are practical*, Journal of Mathematical Cryptology, 2 (2008).
[57] A. M. Odlyzko, *The rise and fall of knapsack cryptosystems*, in Proceedings of Cryptology and Computational Number Theory, vol. 42 of Proceedings of Symposia in Applied Mathematics, American Mathematical Society, 1989, pp. 75–88.
[58] R. A. Pendavingh and S. H. M. van Zwam, *New Korkin-Zolotarev inequalities*, SIAM Journal on Optimization, 18 (2007), pp. 364–378.
[59] X. Pujol, *Recherche efficace de vecteur court dans un réseau euclidien*. Master degree thesis, ENS Lyon, 2008.
[60] X. Pujol and D. Stehlé, *Rigorous and efficient short lattice vectors enumeration*, in Proceedings of Asiacrypt 2008, vol. 5350 of Lecture Notes in Computer Science, Springer-Verlag, 2008, pp. 390–405.
[61] X. Pujol and D. Stehlé, *Solving the shortest lattice vector problem in time* $2^{2.465n}$. Cryptology ePrint Archive, Report 2009/605, 2009. `http://eprint.iacr.org/`.
[62] 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`, 2004.
[63] O. Regev and R. Rosen, *Lattice problems and norm embeddings*, in Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC 2006), ACM, 2006, pp. 447–456.
[64] R. L. Rivest, A. Shamir, and L. M. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Commun. ACM, 21 (1978), pp. 120–126.
[65] C. P. Schnorr, *A hierarchy of polynomial lattice basis reduction algorithms*, Theoretical Computer Science, 53 (1987), pp. 201–224.
[66] ———, *Lattice reduction by random sampling and birthday methods*, in Proceedings of the annual symposium on theoretical aspects of computer science (STACS 2003), vol. 2607 of Lecture Notes in Computer Science, Springer-Verlag, 2003, pp. 145–156.
[67] C. P. Schnorr and M. Euchner, *Lattice basis reduction: improved practical algorithms and solving subset sum problems*, Mathematics of Programming, 66 (1994), pp. 181–199.
[68] C. L. Siegel, *A mean value theorem in geometry of numbers*, The Annals of Mathematics, 46 (1945), pp. 340–347.
[69] E. Viterbi and J. Boutros, *A universal lattice code decoder for fading channels*, IEEE Transactions on Information Theory, 45 (1999), pp. 1639–1642.
[70] A. Wassermann, *Lattice point enumeration and applications*, Bayreuther Mathematische Schriften, 73 (2006).

Appendix A. End of proof of Theorem 5.4.

In order to prove Theorem 5.4, it suffices to show that for any $1 < b < a$, we have:

$$(a-b+1)^{-\frac{a-b}{2}} \cdot T_1 \cdot T_2 \leq \exp\left(-\frac{5}{2}(a-b)\right), \tag{A.1}$$

with $T_1 = \left[1 - \frac{\psi(b)}{\psi(a)} \prod_{k=b}^{a-1} \psi(k+1)^{-\frac{1}{k}}\right]_+^{\frac{a-b}{2}}$, $T_2 = \left[\prod_{k=b}^{a-1} \frac{\psi(a)\psi(k+1)}{\psi(k)\psi(k+1)^{\frac{b-1}{k}}}\right]^{\frac{1}{2}}$ and $\psi(k) = \mathrm{e}^{-6}k$.

A.1. First attempt, without T_1. We consider the logarithm of $(a-b+1)^{-\frac{a-b}{2}} T_2$ and try to show that it is $\leq \frac{5}{2}(b-a)$. Thanks to Lemma 5.7, this is equivalent to:

$$(b-a)\log(a-b+1) + \sum_{k=b}^{a-1} \left[\log\psi(a) - \log\psi(k) + \log\psi(k+1)\left[1 - \frac{b-1}{k}\right]\right] \leq 5(b-a).$$

We first try to simplify the summand.

LEMMA A.1. *The function $x \mapsto -\log x + \log(x+1)\left(1 - \frac{b-1}{x}\right)$ is increasing for $x \geq b$ if $b \geq 3$ and for $x \geq 4$ if $b = 2$.*

Proof. The derivative is $\frac{\log(x+1)(b-1)(x+1)-bx}{x^2(x+1)}$. The function under study is increasing as soon as $\left(1+\frac{1}{x}\right)\log(x+1) \geq \frac{b}{b-1}$. The result follows. □

LEMMA A.2. *The following holds for $a \geq 8$:*

$$\sum_{k=b}^{a-1}\left[\log a - \log k + \log(k+1)\left[1 - \frac{b-1}{k}\right]\right]$$
$$\leq (a-b)\log(a-b+1) + (a-b)\left[\log\frac{a^2}{(a-1)(a-b+1)} - \frac{b-1}{a-1}\log a\right].$$

Proof. When $b \geq 3$, the result follows from Lemma A.1, by using the fact that for all $k \in [b, a-1]$ we have

$$-\log k + \log(k+1)\left[1 - \frac{b-1}{k}\right] \leq -\log(a-1) + \log(a)\left[1 - \frac{b-1}{a-1}\right].$$

Suppose that $b = 2$. The inequality can be checked numerically for $a = 8$. Suppose now that $a > 8$. Then:

$$\sum_{k=b}^{a-1}\left[\log a - \log k + \log(k+1)\left[1 - \frac{1}{k}\right]\right] \leq 6\log 7 + 6\left[\log\frac{64}{49} - \frac{1}{7}\log 8\right]$$
$$+ \sum_{k=8}^{a-1}\left[\log a - \log(a-1) + \log(a)\frac{a-b}{a-1}\right]$$
$$= \sum_{k=2}^{a-1}\left[\log a - \log(a-1) + \log(a)\frac{a-b}{a-1}\right],$$

which gives the result. □

LEMMA A.3. *Let* $\alpha(a,b) = \log\frac{a}{a-b} - \frac{b-1}{a-1}\log a$ *and* $\beta(a,b) = 1 - \frac{b}{a-b}\log\frac{a}{b}$. *For* $a \geq 8$, *we have:*

$$(b-a)\log(a-b+1) + \sum_{k=b}^{a-1}\left[\log\psi(a) - \log\psi(k) + \log\psi(k+1)\left[1 - \frac{b-1}{k}\right]\right] \leq (a-b)\left[\alpha(a,b) + \beta(a,b)\log C\right].$$

Proof. Lemma A.2 and the fact that $(a-1)(a-b+1) \geq a(a-b)$ give:

$$(b-a)\log(a-b+1) + \sum_{k=b}^{a-1}\left[\log a - \log k + \log(k+1)\left[1 - \frac{b-1}{k}\right]\right] \leq (a-b)\alpha(a,b).$$

We now consider the terms depending on C. Since $\sum_{x=b}^{a-1}\frac{1}{x} \leq \log\frac{a-1}{b-1}$ and $\log C < 0$, we have:

$$\sum_{k=b}^{a-1}\left(\log(C)\left(1 - \frac{b-1}{k}\right)\right) \leq \log(C)\left(a - b - (b-1)\log\frac{a-1}{b-1}\right).$$

The fact that $(b-1)\log\frac{a-1}{b-1} \leq b\log\frac{a}{b}$ completes the proof. □

In the following, we study the function $(a,b) \mapsto \alpha(a,b) + \beta(a,b)\log C$. We would like to bound it by -5, but we will only be able to do this for a subset of all possible values for the pair (a,b).

LEMMA A.4. *Let* $0 < \kappa < 1$ *be a real constant and suppose that* $a \geq 8$. *The function* $a \mapsto \alpha(a,\kappa a) + \beta(a,\kappa a)\log C$ *decreases with respect to* a.

Proof. We have

$$\alpha(a,\kappa a) + \beta(a,\kappa a)\log C = -\log(1-\kappa) + \log C\left(1 + \frac{\kappa\log\kappa}{1-\kappa}\right) - \frac{(\kappa a - 1)\log a}{a-1}.$$

Hence,

$$\frac{\partial}{\partial a}\left(\alpha(a,\kappa a) + \log C\beta(a,\kappa a)\right) = \frac{-\kappa a^2 + (\kappa-1)a\log a + (\kappa+1)a - 1}{a(a-1)^2}.$$

For the numerator to be negative, it suffices that $a \geq 1 + \frac{1}{\kappa}$ (then the term in a^2 is larger than the term in a) or that $a \geq \exp\left(\frac{\kappa+1}{1-\kappa}\right)$ (then the term in $a\log a$ is larger than the term in a). Since

$$\max_{\kappa\in[0,1]}\min\left(1 + \frac{1}{\kappa}, \exp\left(\frac{\kappa+1}{1-\kappa}\right)\right) \leq 8,$$

the result follows. □

In the results above, we did not need $C = \exp(-6)$. The only property we used about C was $\log C < 0$. In the sequel, we define $\tau(a,\kappa) = \alpha(a,\kappa a) - 6\beta(a,\kappa a)$. We are to prove that $\tau(a,\kappa) \leq -5$ as soon as κ is not very close to 1.

LEMMA A.5. *For any* $a \geq 756$, *the function* $\kappa \mapsto \tau(a,\kappa)$ *increases to a local maximum in* $\left[0, \frac{1}{2}\right]$, *then decreases to a local minimum in* $\left[\frac{1}{2}, 1 - \frac{1}{2\log a}\right]$ *and then increases.*

Proof. We first study

$$\frac{\partial^3}{\partial\kappa^3}\tau(a,\kappa) = \frac{20\kappa^2+10\kappa^3+6-36\kappa-36\kappa^2\log\kappa}{(1-\kappa)^4\kappa^2}.$$

Using the fact that $\log\kappa \le (\kappa-1)-(\kappa-1)^2/2+(\kappa-1)^3/3$ for $\kappa\in(0,1]$, we find that the numerator can be lower bounded by a polynomial which is non-negative for $\kappa\in(0,1]$. As a consequence, $\tau'_\kappa(a,\kappa)=\frac{\partial}{\partial\kappa}\tau(a,\kappa)$ is a convex function with respect to $\kappa\in(0,1)$.

Notice now that $\tau'_\kappa(a,\kappa) = -6\log\kappa+o(\log\kappa) > 0$ for κ close to 0, that $\tau'_\kappa(a,1/2) = -10+24\log 2-\frac{a\log a}{a-1}\le 0$ for $a\ge 756$, and finally that

$$\begin{aligned}\tau'_\kappa\left(a,1-\frac{1}{2\log a}\right) &= -10\log a-24\log\left(1-\frac{1}{2\log a}\right)\log^2 a-\frac{a}{a-1}\log a\\ &\ge 2\log a-\frac{a}{a-1}\log a,\end{aligned}$$

which is clearly positive for $a\ge 2$. □

The following lemma provides the result claimed in Theorem 5.4 for $a\ge 158000$ and $b\le a-1.65\frac{a}{\log^3 a}$.

LEMMA A.6. *Suppose that $a\ge 158000$. Then, for all $\kappa\le 1-1.65\frac{1}{\log^3 a}$, we have $\alpha(a,\kappa a)-6\beta(a,\kappa a)\le -5$.*

Proof. Let $a_0=158000$. We have $\tau'_\kappa(a_0,0.08962)>0>\tau'_\kappa(a_0,0.08963)$. Furthermore, for $\kappa\in[0.08962,0.08963]$, we have

$$|\tau'_\kappa(a_0,\kappa)|\le\max\left(|\tau'_\kappa(a_0,0.08962)|,|\tau'_\kappa(a_0,0.08963)|\right)\le 3\cdot 10^{-4}.$$

Hence,

$$\max_{\kappa\in[0.08962,0.08963]}\tau(a_0,\kappa)\le\tau(a_0,0.08962)+3\cdot 10^{-9}\le -5.$$

Lemma A.5 implies that $\max_{\kappa\in[0,1/2]}\tau(a_0,\kappa)\le -5$. Thanks to Lemma A.4, we have, for $a\ge 158000$:

$$\max_{\kappa\in[0,1/2]}\left(\alpha(a,\kappa a)-6\beta(a,\kappa a)\right)\le -5.$$

Furthermore, since $\frac{1}{2\log a}\ge\frac{1.65}{\log^3 a}$ and thanks to Lemma A.5, we have, for any $a\ge 158000$:

$$\max_{\kappa\in\left[\frac12,1-\frac{1.65}{\log^3 a}\right]}\tau(a,\kappa)=\max\left(\tau\left(a,\frac12\right),\tau\left(a,1-\frac{1.65}{\log^3 a}\right)\right).$$

Notice that

$$\tau\left(a,1-\frac{1.65}{\log^3 a}\right)\le\alpha\left(a,a-\frac{1.65a}{\log^3 a}\right)=-\log 1.65+3\log\log a-\log a+\frac{a}{a-1}\frac{1.65}{(\log a)^2},$$

which is decreasing with respect to $a\ge 158000$. Moreover, for $a=158000$, its value is below -5. As a consequence,

$$\max_{\kappa\in\left[\frac12,1-\frac{1.65}{\log^3 a}\right]}\tau(a,\kappa)\le\max\left(\tau\left(a,\frac12\right),-5\right)\le -5.$$

□

A.2. Using T_1 when $b > a - \frac{1.65a}{\log^3 a}$. This section ends the proof of Theorem 5.4 for $a \geq 158000$.

LEMMA A.7. *Assume that $\psi(x) = e^{-6} \cdot x$. Then, for $a > b \geq a - 1.65\frac{a}{\log^3 a}$, we have*

$$T_1^{\frac{2}{a-b}} = 1 - \left(\frac{f_{\psi,d}(d-b+1)}{f_{\psi,d}(d-a+1)}\right)^2 \leq 1.65\frac{\log a - 5}{\log^3 a - 1.65}.$$

Proof. According to Lemma 5.3, we have

$$\begin{aligned} -2\log\frac{f_{\psi,d}(d-b+1)}{f_{\psi,d}(d-a+1)} &= \log\left(\frac{a}{b}\right) + \sum_{l=b}^{a-1}\frac{-6+\log(l+1)}{l} \\ &\leq \frac{a-b}{b}\left(1 + (-6+\log a)\right), \\ &\leq 1.65\frac{\log a - 5}{\log^3 a - 1.65}. \end{aligned}$$

□

By using Lemmas A.3 and A.5 and the fact that $\beta(a,b)\log(C) \leq 0$, we have, for $b \geq a - 1.65\frac{a}{(\log a)^3}$ and $a \geq a_1 \geq 158000$:

$$\begin{aligned} (b-1)\log(a-b+1) + 2\log T_1T_2 &\leq (a-b)[\alpha(a,b) + \beta(a,b)\log(C)] + 2\log T_1 \\ &\leq (a-b)\left[\alpha(a,a-1) + \frac{2}{a-b}\log T_1\right] \\ &\leq (a-b)\left[\frac{\log a}{a-1} + \log\left[1.65\frac{\log a - 5}{\log^3 a - 1.65}\right]\right]. \end{aligned}$$

The term $\frac{\log a}{a-1} + \log\left(1.65\frac{\log a-5}{\log^3 a-1.65}\right)$ decreases for $a \geq 1782$ and becomes ≤ -5 for $a \leq 158000$, thus completing the proof.

A.3. Small Values of a. It only remains to prove Theorem 5.4 for small values of a. The following lemma was obtained numerically. In order to provide a reliable proof, we used the Boost interval arithmetic library [15] and CRlibm [19] as underlying floating-point libraries.

LEMMA A.8. *Equation* (A.1) *holds for any $1 < b < a \leq 158000$.*