Improved Analysis of Kannan's Shortest Lattice Vector Algorithm (Extended Abstract)

Guillaume Hanrot[1]* and Damien Stehlé[2]

[1] LORIA/INRIA Lorraine, Technopôle de Nancy-Brabois,
615 rue du jardin botanique, F-54602 Villers-lès-Nancy Cedex, France.
hanrot@loria.fr – http://www.loria.fr/~hanrot
[2] CNRS and ÉNS Lyon/ LIP, 46 allée d'Italie, 69364 Lyon Cedex 07, France.
damien.stehle@ens-lyon.fr – http://perso.ens-lyon.fr/damien.stehle

Abstract. The security of lattice-based cryptosystems such as NTRU, GGH and Ajtai-Dwork essentially relies upon the intractability of computing a shortest non-zero lattice vector and a closest lattice vector to a given target vector in high dimensions. The best algorithms for these tasks are due to Kannan, and, though remarkably simple, their complexity estimates have not been improved since over twenty years. Kannan's algorithm for solving the shortest vector problem (SVP) is in particular crucial in Schnorr's celebrated block reduction algorithm, on which rely the best known generic attacks against the lattice-based encryption schemes mentioned above. In this paper we improve the complexity upper-bounds of Kannan's algorithms. The analysis provides new insight on the practical cost of solving SVP, and helps progressing towards providing meaningful key-sizes.

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 at most n vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$. These vectors can be chosen linearly independent, and in that case, we say that they are a basis of the lattice L. The most famous algorithmic problem associated with lattices is the so-called shortest vector problem (SVP). Its computational variant is to find a non-zero lattice vector of smallest Euclidean length — this length being the minimum $\lambda(L)$ of the lattice — given a basis of the lattice. Its decisional variant is known to be NP-hard under randomised reductions [2], even if one only asks for a vector whose length is no more than $2^{(\log d)^{1-\epsilon}}$ times the length of a shortest vector [12] (for any $\epsilon > 0$).

SVP is of prime importance in cryptography since a now quite large family of public-key cryptosystems relies more or less on it. The Ajtai-Dwork cryptosystem [4] relies on d^c-SVP for some $c > 0$, where $f(d)$-SVP is the problem of finding the shortest non-zero vector in the lattice L,

* Work partially supported by CNRS GDR 2251 "Réseau de théorie des nombres".

under the promise that any vector of length less than $f(d) \cdot \lambda(L)$ is parallel to it. The GGH cryptosystem [11] relies on special instances of the Closest Vector Problem (CVP), a non-homogeneous version of SVP. Both the Ajtai-Dwork and GGH cryptosystems have been shown impractical for real-life parameters [25, 23] (the initial GGH containing a major theoretical flaw as well). Finally, one strongly suspects that in NTRU [15] the private key can be read on the coordinates of a shortest vector of the Coppersmith-Shamir lattice [8]. The best known generic attacks against these encryption schemes are based on solving SVP. It is therefore highly important to know precisely what complexity is achievable, both in theory and practice, in particular to select meaningful key-sizes. Most often, for cryptanalysing lattice-based cryptosystems, one considers Schnorr's block-based algorithms [28, 30], such as BKZ. These algorithms internally solve instances of SVP in much lower dimensions (related to the size of the block). They help solving relaxed variants of SVP in high dimensions. Increasing the dimensions up to which one can solve SVP helps decreasing the relaxation factors that are achievable in higher dimensions. Solving the instances of SVP is the computationally expensive part of the block-based reduction algorithms.

Two main algorithms are known for solving SVP. The first one is based on the deterministic exhaustive enumeration of lattice points within a small convex body. It is known as Fincke-Pohst's enumeration algorithm [9] in the algorithmic number theory community. Cryptographers know it as Kannan's algorithm [16]. There are two main differences between both: firstly, in Kannan's algorithm, a long pre-computation on the basis is performed before starting the enumeration process; secondly, Kannan enumerates integer points in a hyper-parallelepiped whereas Fincke and Pohst consider an hyper-ellipsoid which is strictly contained in Kannan's hyper-parallelepiped – though Kannan may have chosen the hyper-parallelepiped in order to simplify the complexity analysis. Kannan obtained a $d^{d+o(d)}$ complexity bound (in the complexity bounds mentioned in the introduction, there is an implicit factor that is polynomial in the bit-size of the input). In 1985, Helfrich [13] refined Kannan's analysis, and obtained a $d^{d/2+o(d)}$ complexity bound. On the other hand, Ajtai, Kumar and Sivakumar [5] designed a probabilistic algorithm of complexity $2^{O(d)}$. The best exponent constant is likely to be small, as suggested by some recent progress [26]. A major drawback of this algorithm is that it requires an exponential space, whereas Kannan's requires a polynomial space.

Our main result is to lower Helfrich's complexity bound on Kannan's algorithm, from $d^{\frac{d}{2}+o(d)} \approx d^{0.5 \cdot d}$ to $d^{\frac{d}{2e}+o(d)} \approx d^{0.184 \cdot d+o(d)}$. This may explain why Kannan's algorithm is tractable even in moderate dimensions. Our analysis can also be adapted to Kannan's algorithm for CVP: it decreases Helfrich's complexity bound from $d^{d+o(d)}$ to $d^{d/2+o(d)}$. The complexity improvement for SVP provides better worst-case efficiency/quality trade-offs for Schnorr's block-based algorithms [28, 30, 10].

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

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

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

where $(\boldsymbol{b}_i^*)_{i \leq d}$ is the Gram-Schmidt orthogonalisation of the $\boldsymbol{b}_i$'s. This generalises the results of [28] on the quality of HKZ-reduced bases.

Practical Implications. We do not change Kannan's algorithm, but only improve its complexity upper-bound. As a consequence, the running-time of Kannan's algorithm remains the same. Nevertheless, our work may still have some important practical impact. First of all, it revives the interest on Kannan's algorithm. Surprisingly, although it has the best complexity upper-bound, it is not the one implemented in the usual number theory libraries (e.g., NTL [32] and Magma [18] implement Schnorr-Euchner's variant [30]): we show that by using Kannan's principle (i.e., pre-processing the basis before starting the enumeration), one can solve SVP in larger dimensions. This might point a problem in NTRU's security estimates, since they are derived from experimentations with NTL. Secondly, our analysis helps providing a heuristic measure of the (practical) cost of solving SVP for a particular instance, which is both efficiently computable and reliable: given a lattice basis, it provides very quickly a heuristic upper bound on the cost of finding a shortest vector.

Road-Map of the Paper. In Section 2, we recall some basic definitions and properties on lattice reduction. Section 3 is devoted to the description of Kannan's algorithm and Section 4 to its complexity analysis. In Section 5, we give without much detail our sibling result on CVP, as well as direct consequences of our result for block-based algorithms. In Section 6, we discuss the practical implications of our work.

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

2 Background on Lattice Reduction

We assume that the reader is familiar with the geometry of numbers and its algorithmic aspects. Introductions may be found in [21] and [27].

Lattice invariants. Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ be linearly independent vectors. Their *Gram-Schmidt orthogonalisation* (GSO) $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_d^*$ is the orthogonal family defined recursively as follows: the vector $\boldsymbol{b}_i^*$ is the component of $\boldsymbol{b}_i$ which is orthogonal to the span of the vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}$. We have $\boldsymbol{b}_i^* = \boldsymbol{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \boldsymbol{b}_j^*$ where $\mu_{i,j} = \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^* \rangle}{\|\boldsymbol{b}_j^*\|^2}$. For $i \leq d$ we let $\mu_{i,i} = 1$. Notice that the GSO family depends on the order of the vectors. If the $\boldsymbol{b}_i$'s are integer vectors, the $\boldsymbol{b}_i^*$'s and the $\mu_{i,j}$'s are rational. The *volume* of a lattice L is defined as $\det(L) = \prod_{i=1}^{d} \|\boldsymbol{b}_i^*\|$, where the $\boldsymbol{b}_i$'s are any basis of L. It does not depend on the choice of the basis of L and can be interpreted as the geometric volume of the parallelepiped naturally spanned by the basis vectors. Another important lattice invariant is the minimum. The *minimum* $\lambda(L)$ is the length of a shortest non-zero lattice vector.

The most famous lattice problem is the *shortest vector problem* (SVP). Here is its computational variant: given a basis of a lattice L, find a lattice vector whose norm is exactly $\lambda(L)$. The *closest vector problem* (CVP) is a non-homogeneous variant of SVP. We give here its computational variant: given a basis of a lattice L and a target vector in the real span of L, find a vector of L which is closest to the target vector.

The volume and the minimum of a lattice cannot behave independently. Hermite [14] was the first to bound the ratio $\frac{\lambda(L)}{(\det L)^{1/d}}$ as a function of the dimension only. His bound was later on greatly improved by Minkowski in his *Geometrie der Zahlen* [22]. *Hermite's constant* γ_d is defined as the supremum over d-dimensional lattices L of $\frac{\lambda(L)^2}{(\det L)^{2/d}}$. We have $\gamma_d \leq \frac{d+4}{4}$ (see [19]), which we will refer to as *Minkowski's theorem.*

Lattice reduction. In order to solve lattice problems, a classical strategy consists in considering a lattice basis and trying to improve its quality (e.g., the slow decrease of the $\|\boldsymbol{b}_i^*\|$'s). This is called *lattice reduction.* The most usual notions of reduction are probably L^3 and HKZ. HKZ-reduction is very strong, but expensive to compute. On the contrary, L^3-reduction is fairly cheap, but an L^3-reduced basis is of much lower quality.

A basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ is *size-reduced* if its GSO family satisfies $|\mu_{i,j}| \leq 1/2$ for all $1 \leq j < i \leq d$. A basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ is said to be *Hermite-Korkine-Zolotarev-reduced* if it is size-reduced, the vector $\boldsymbol{b}_1$ reaches the lattice minimum, and the projections of the $(\boldsymbol{b}_i)_{i\geq 2}$'s orthogonally to the vector $\boldsymbol{b}_1$ are themselves an HKZ-reduced basis. Lemma 1 immediately follows from this definition and Minkowski's theorem. It is the sole property on HKZ-reduced bases that we will use.

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

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

A basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ is *L^3-reduced* [17] if it is size-reduced and if its GSO satisfies the $(d-1)$ Lovász conditions: $\frac{3}{4} \cdot \left\|\boldsymbol{b}_{\kappa-1}^*\right\|^2 \leq \left\|\boldsymbol{b}_\kappa^* + \mu_{\kappa,\kappa-1}\boldsymbol{b}_{\kappa-1}^*\right\|^2$. The L^3-reduction implies that the norms of the GSO vectors never drop too fast: intuitively, the vectors are not far from being orthogonal. Such bases have useful properties, like providing exponential approximations to SVP and CVP. In particular, their first vector is relatively short.

Theorem 1 ([17]). *Let $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ be an L^3-reduced basis of a lattice L. Then we have $\|\boldsymbol{b}_1\| \leq 2^{\frac{d-1}{4}} \cdot (\det L)^{1/d}$. Moreover, there exists an algorithm that takes as input any set of integer vectors and outputs in deterministic polynomial time an L^3-reduced basis of the lattice they span.*

In the following, we will also need the fact that if the set of vectors given as input to the L^3 algorithm starts with a shortest non-zero lattice vector, then this vector is not changed during the execution of the algorithm: the output basis starts with the same vector.

3 Kannan's SVP Algorithm

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

3.1 Short Lattice Points Enumeration

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

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

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

3.2 Solving SVP

To solve SVP, Kannan provides an algorithm that computes HKZ-reduced bases, see Figure 2. The cost of the enumeration procedure dominates the overall cost and mostly depends on the quality of the input basis. The main idea of Kannan's algorithm is to spend a lot of time pre-computing a basis of excellent quality before calling the enumeration procedure. More precisely, it pre-computes a so-called quasi-HKZ-reduced basis.

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

Fig. 1. The enumeration algorithm.

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

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

Fig. 2. Kannan's SVP algorithm.

A few comments need to be made on the algorithm of Figure 2. Steps 3 and 9 are recursive calls. However, the $\boldsymbol{b}_i'$'s may be rational vectors, whereas the input of the algorithm must be integral. These vectors may be scaled by a common factor. Steps 4 and 10 may be performed by expressing the reduced basis vectors as integer linear combinations of the initial ones, using these coefficients to recover lattice vectors and subtracting a correct multiple of the vector $\boldsymbol{b}_1$. In Step 6, it is possible to choose such a vector $\boldsymbol{b}_0$, since this enumeration always provides non-zero solutions (the vector $\boldsymbol{b}_1$ is one of them).

3.3 Cost of Kannan's SVP Solver

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

- Kannan's algorithm computes an HKZ-reduced basis of the lattice spanned by the input vectors.
- All arithmetic operations performed during the execution are of cost $\mathcal{P}(d,n,\log B)$. This implies that $C(d,n,B)$ can be bounded by $C(d) \cdot \mathcal{P}(\log B, n)$ for some function $C(d)$.
- There are fewer than $O(1) + \log d$ iterations of the loop of Steps 2–5.
- The cost of the call to the enumeration procedure at Step 6 is bounded by $\mathcal{P}(\log B, n) \cdot d^{d/2+o(d)}$.

From these properties and those of the L^3 algorithm as recalled in the previous section, it is easy to obtain the following equation:

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

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

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

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

4 Complexity of the Enumeration Procedure

This section is devoted to proving Theorem 2. The previous section has shown that the cost of Kannan's algorithm is dominated by the time for enumerating the integer points in the hyper-ellipsoids $(\mathcal{E}_i)_{1\leq i\leq d}$ defined by $\mathcal{E}_i = \left\{(y_i, \ldots, y_d) \in \mathbb{R}^{d-i+1}, \|\sum_{j\geq i} y_j \boldsymbol{b}_j^{(i)}\| \leq \|\boldsymbol{b}_1\|\right\}$, where $\boldsymbol{b}_j^{(i)} = \boldsymbol{b}_j - \sum_{k<i} \mu_{j,k} \boldsymbol{b}_k^*$ is the vector $\boldsymbol{b}_j$ once projected orthogonally to $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_{i-1}^*$.

Classically, the number of integer points in a body of some $\mathbb{R}^n$ is heuristically estimated by the n-dimensional volume of the body. This yields the following heuristic complexity upper-bound for Kannan's algorithm:

$$\max_{i \le d} \frac{V_i \|\boldsymbol{b}_1\|^i}{\prod_{j \ge d-i+1} \|\boldsymbol{b}_j^*\|} \lesssim \max_{i \le d} \frac{\|\boldsymbol{b}_1\|^i}{(\sqrt{i})^i \cdot \prod_{j \ge d-i+1} \|\boldsymbol{b}_j^*\|}, \tag{1}$$

where V_i is the volume of the i-dimensional unit ball.

Here, such an estimate may not be too optmistic since the hyper-ellipsoids might be too flat for the approximation by the volume to be valid. The first step of our analysis is to prove a slight modification of this heuristic estimate. This is essentially an adaptation of a method due to Mazo and Odlyzko [20] to bound the number of integer points in hyperspheres. We prove the weaker upper bound $\max_{I \subset [\![1,d]\!]} \frac{\|\boldsymbol{b}_1\|^{|I|}}{\sqrt{d}^{|I|} \prod_{i \in I} \|\boldsymbol{b}_i^*\|}$, for quasi-HKZ-reduced bases (Subsections 4.1 and 4.2).

In the second step of our analysis (Subsection 4.3), we bound the above quantity. This involves a rather precise study of the geometry of HKZ-reduced bases. The only available tool is Minkowski's inequality, which is used numerous times. For the intuition, the reader should consider the typical case where $(\boldsymbol{b}_i)_{1 \le i \le d}$ is an HKZ-reduced basis for which $(\|\boldsymbol{b}_i^*\|)_i$ is a non-increasing sequence. In that case, the first part of the analysis shows that one has to consider a set I of much simpler shape: it is an interval $[\![i, d]\!]$ starting at some index i. Lemmata 2 and 3 (which should thus be considered as the core of the proof) and the fact that $x \log x \ge -1/e$ for $x \in [0, 1]$ are sufficient to deal with such sets.

Non-connex sets I are harder to handle. We split the HKZ-reduced basis into *blocks* (defined by the expression of I as a union of intervals), i.e., groups of consecutive vectors $\boldsymbol{b}_i, \ldots, \boldsymbol{b}_{j-1}$ such that $i, \ldots, k-1 \notin I$ and $k, \ldots, j-1 \in I$. The former vectors will be the "large ones" and the latter the "small ones". Over each block, Lemma 3 relates the average size of the small vectors to the average size of the whole block. We consider the blocks by decreasing indices and use an amortised analysis to combine the local behaviours on blocks to obtain a global bound (Lemma 4). A final convexity argument gives the result (Lemma 5).

4.1 Integer Points in Hyper-ellipsoids

In this subsection, we do not assume anything on the input basis vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ and on the input bound A. Up to some polynomial in d and $\log B$, the complexity of the enumeration procedure of Figure 1 is the number of loop iterations. This number of iterations is itself bounded by $3\sum_{i=1}^{d}|\mathcal{E}_i|$. Indeed, the truncated coordinate $(x_i,\dots,x_d)$ is either a valid one, i.e., we have $\|\sum_{j=i}^{d} x_j \boldsymbol{b}_j^{(i)}\|^2 \leq A$, or $(x_i-1,\dots,x_d)$ is a valid one, or $(x_{i+1},\dots,x_d)$ is a valid one. In fact, if $(x_i,\dots,x_d)$ is a valid truncated coordinate, at most two non-valid ones related to that one may be considered during the execution of the algorithm: $(x_i+1,\dots,x_d)$ and $(x_{i-1},x_i\dots,x_d)$ for at most one integer x_{i-1}. We now fix some $i \leq d$. By applying the change of variable $x_j \leftarrow x_j - \left\lfloor \sum_{k>j}\mu_{k,j}x_k \right\rceil$, we obtain:

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

If x is an integer and $\epsilon \in [-1/2,1/2]$, then we have $(x+\epsilon)^2 \geq x^2/4$ (it suffices to use the inequality $|\epsilon| \leq 1/2 \leq |x|/2$, which is valid for a non-zero x). As a consequence, up to a polynomial factor, the complexity of the enumeration is bounded by $\sum_{i\leq d} N_i$, where $N_i = \left|\mathcal{E}_i' \cap \mathbb{Z}^{d-i+1}\right|$ and $\mathcal{E}_i' = \left\{(y_i,\dots,y_d) \in \mathbb{R}^{d-i+1}, \sum_{j\geq i} y_j^2 \|\boldsymbol{b}_j^*\|^2 \leq 4A\right\}$, for any $i \leq d$.

We again fix some index i. The following sequence of relations is inspired from [20, Lemma 1].

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

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

$$
N_i \leq (4e(1+\sqrt{\pi}))^d \cdot \prod_{j\geq i} \max\left(1, \frac{\sqrt{A}}{\sqrt{d}\|\boldsymbol{b}_j^*\|}\right). \qquad (2)
$$

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

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

4.2 The Case of Quasi-HKZ-Reduced Bases

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

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

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

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

As a consequence, Theorem 2 follows from the following:

Theorem 3. *Let $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ be HKZ-reduced and $I \subset [\![1, d]\!]$. Then*

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

By applying Theorem 3 the HKZ-reduced basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_i)$ and $I = \{i\}$, we recover the result of [28]: $\|\boldsymbol{b}_i^*\| \geq (\sqrt{i})^{-\log i - 1} \cdot \|\boldsymbol{b}_1\|$.

4.3 A Property on the Geometry of HKZ-Reduced Bases

In this section, we prove Theorem 3, which is the last missing part to obtain the claimed result. The proofs of the following lemmata will be contained in the full version of this paper. In the sequel, $(\boldsymbol{b}_i)_{i \leq d}$ is an HKZ-reduced basis of a lattice L of dimension $d \geq 2$.

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

We need upper bounds on $\Gamma_d(k)$ and a technical lemma allowing us to finely recombine such bounds. Intuitively, the following lemma is a rigorous version of the identity:

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

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

We now give an "averaged" version of [28, Lemma 4], deriving from Lemma 2. This provides the result claimed in Theorem 3 for any set I of the shape $[\![i,j]\!]$, for any $i \leq j \leq d$.

Lemma 3. *For all* $k \in [\![0, d-1]\!]$, *we have* $\pi_{[\![1,k]\!]} \leq (\Gamma_d(k))^{d/k} \cdot \pi_{[\![k+1,d]\!]}$ *and* $\pi_{[\![k+1,d]\!]} \geq (\Gamma_d(k))^{-1} \cdot (\det L)^{1/d} \geq \sqrt{d}^{\log \frac{d-k}{d}} (\det L)^{1/d}$.

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

Lemma 4. *Let* $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ *be an HKZ-reduced basis. Let* $v \in [\![2, d]\!]$, $I \subset [\![v+1, d]\!]$ *and* $u \in [\![1, v]\!]$. *Assume that:*

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

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

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

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

The last ingredient to the proof of Theorem 3 is the following, which derives from the convexity of the function $x \mapsto x \log x$.

Lemma 5. *Let* $\Delta \geq 1$, *and define* $F_\Delta(k, d) = \Delta^{-k \log \frac{k}{d}}$. *We have, for any* $t \in \mathbb{Z}$, *for any* $k_1, \ldots, k_t \in \mathbb{Z}$ *and* $d_1, \ldots, d_t \in \mathbb{Z}$ *such that* $1 \leq k_i < d_i$ *for all* $i \leq t$,

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

Finally, Theorem 3 follows from Lemmata 4 and 5.

Proof of Theorem 3. Lemma 4 gives us, by induction on the size of the considered set I, that for all $I \subset [\![1,d]\!]$:

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

where $I_i = I \cap [\![\alpha_i+1,\alpha_{i+1}]\!]$, and t, the α_i's, and the densities $\delta_i = \frac{|I_i|}{\alpha_{i+1}-\alpha_i}$ satisfy $t \geq 1$, $0 = \alpha_1 < \ldots < \alpha_t \leq d$ and $1 \geq \delta_1 > \ldots > \delta_{t-1} > 0$. By using Lemma 5 with $\Delta{:=}\sqrt{d}$, $k_i{:=}|I_i|$ and $d_i{:=}\alpha_{i+1}-\alpha_i$, we obtain:

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

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

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

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

$$\frac{\pi_I^{|I|}}{\sqrt{d}^{|I| \log \frac{|I|}{d}}} \geq \left(\frac{\|\boldsymbol{b}_1\|}{\sqrt{d}} \right)^{\sum_{j<t} \alpha_{j+1}(\delta_j - \delta_{j+1})} \geq \left(\frac{\|\boldsymbol{b}_1\|}{\sqrt{d}} \right)^{|I|}.$$

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

5 CVP and Other Related Problems

Our improved analysis of Kannan's algorithm can be adapted to the Closest Vector Problem and other problems related to strong lattice reduction.

In CVP, we are given a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ and a target vector $\boldsymbol{t}$, and we look for a lattice vector that is closest to $\boldsymbol{t}$. Kannan's CVP algorithm starts by HKZ-reducing the $\boldsymbol{b}_i$'s. Then it runs a slight modification of the enumeration algorithm of Figure 1. For the sake of simplicity, we assume that $\|\boldsymbol{b}_1^*\|$ is the largest of the $\|\boldsymbol{b}_i^*\|$'s (we refer to Kannan's proof [16] for the general case). By using Babai's nearest hyperplane strategy [6], we see that there is a lattice vector $\boldsymbol{b}$ at distance less than $\sqrt{d} \cdot \|\boldsymbol{b}_1\|$ of the target vector $\boldsymbol{t}$. As a consequence, if we take $A = d \cdot \|\boldsymbol{b}_1\|^2$ in the modified enumeration procedure, we will find all solutions. The analysis then reduces (at the level of Equation (2)) to bound the ratio $\frac{\|\boldsymbol{b}_1\|^d}{\prod_{i \leq d} \|\boldsymbol{b}_i^*\|}$, which can be done with Minkowski's theorem.

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

The best deterministic complexity upper bound previously known for this problem was $\mathcal{P}(\log B, n) \cdot d^{d+o(d)}$ (see [13, 7]).

Our result can also be adapted to the enumeration of all vectors of a given lattice that are of length below a prescribed bound, which is in particular useful in the context of computing lattice theta series. Another important consequence of our analysis is a significant worst-case bound improvement of Schnorr's block-based strategy [28] to compute relatively short vectors in high-dimensional lattices. More precisely, if we take the bounds given in [10] for the quality of Schnorr's semi-$2k$ reduction and for the transference reduction, we obtain the table of Figure 3. Each entry of the table gives the upper bound of the quantity $\frac{\|\boldsymbol{b}_1\|}{(\det L)^{1/d}}$ which is reachable for a computational effort of 2^t, for t growing to infinity. To sum up, the exponent constant is divided by $e \approx 2.7$. The table upper bounds may be adapted to the quantity $\frac{\|\boldsymbol{b}_1\|}{\lambda_1(L)}$ by squaring them.

	Semi-$2k$ reduction	Transference reduction
Using [13]	$\lesssim 2^{\frac{\log 2}{2} \frac{d \log^2 t}{t}} \approx 2^{0.347 \frac{d \log^2 t}{t}}$	$\lesssim 2^{\frac{1}{4} \frac{d \log^2 t}{t}} \approx 2^{0.250 \frac{d \log^2 t}{t}}$
Using Theorem 2	$\lesssim 2^{\frac{\log 2}{2e} \frac{d \log^2 t}{t}} \approx 2^{0.128 \frac{d \log^2 t}{t}}$	$\lesssim 2^{\frac{1}{4e} \frac{d \log^2 t}{t}} \approx 2^{0.092 \frac{d \log^2 t}{t}}$

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

6 Practical Implications

As mentioned in the introduction, the main contribution of the present paper is to improve the worst-case complexity analysis of an already known algorithm, namely, Kannan's HKZ-reduction algorithm. Our improvement has no direct impact on the practical capabilities of lattice reduction algorithms. However, our work may have two indirect consequences: popularising Kannan's principle and providing easily computable cost estimates for SVP instances.

6.1 Pre-processing Before Enumerating

In the main libraries containing lattice reduction routines, the shortest vector problem is solved with the enumeration routine, but starting from only L^3-reduced bases. This is the case for the `BKZ` routines of Victor Shoup's NTL [32], which, depending on a parameter k, compute strongly reduced bases in high dimensions (the quality being quantified by k). This is also the case in Magma's `ShortestVectors` routine [18], which computes the shortest vectors of a given lattice. Both rely on the enumeration of Schnorr and Euchner [30]. On the theoretical side, this strategy is worse than using Kannan's algorithm, the worst-case complexity being $2^{O(d^2)}$ instead of $d^{O(d)}$. To justify this choice, one might argue that L^3 computes much better bases in practice than guaranteed by the worst-case bounds, in particular in low dimensions (see [24] for more details), and that the asymptotically superior algorithm of Kannan may overtake the L^3-based enumeration only for large dimensions (in particular too large to be tractable).

It may be that the genuine Kannan algorithm is expensive. However, the general principle of enumerating from a more than L^3-reduced basis works, as the following experiments tend to show. For a given dimension d, we consider the lattice spanned by the columns of the following matrix:

$$\begin{pmatrix} x_1 & x_2 & \dots & x_d \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix},$$

where the x_i's are chosen uniformly and independently in $[\![0, 2^{100 \cdot d}]\!]$. The basis is then L^3-reduced with a close to optimal parameter ($\delta = 0.99$). For the same lattice, we compute more reduced bases, namely BKZ$_k$-reduced for different parameters k, using NTL's `BKZ_FP` routine without pruning and close to optimal factor ($\delta = 0.99$). We run the same enumeration routine starting from these different bases and compare the timings. The results of the experiments are given in Figure 4. The enumeration is a non-optimised C-code, which updates the norm upper bound during the enumeration [30]. All timings are given in seconds and include the BKZ-reduction (unless we start from the L^3-reduced basis). Each point corresponds to the average over at least 10 samples. The experiments were performed on 2.4 GHz AMD Opterons. The enumeration from an

L^3-reduced basis is clearly outperformed. BKZ-reducing the basis with larger block-sizes becomes more interesting when the dimension increases: it seems that in moderate dimension, a BKZ_k reduced basis is close to being HKZ-reduced, even when k is small with respect to the dimension.

pre-processing	$d = 40$	$d = 43$	$d = 46$	$d = 49$	$d = 52$	$d = 55$	$d = 58$
L^3	1.8	15	110	990	$5.0 \cdot 10^3$	–	–
BKZ_{10}	0.36	1.6	6.7	36	160	–	–
BKZ_{20}	0.40	1.3	4.7	21	96	800	$2.5 \cdot 10^3$
BKZ_{30}	0.57	1.7	5.2	19	68	660	$1.6 \cdot 10^3$

Fig. 4. Comparison between various pre-processings.

6.2 Estimating the Cost of Solving SVP

The cost of solving SVP on a particular instance with the enumeration routine is essentially dominated by the cost of the highest-dimensional enumeration. Up to a polynomial factor, the cost of the enumeration as described in Figure 1 can be estimated with Equation (1):

$$E(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d) := \max_{i \leq d} \frac{\pi^{i/2} \cdot \|\boldsymbol{b}_1\|^i}{\Gamma(i/2+1) \cdot \prod_{j \geq d-i+1} \|\boldsymbol{b}_j^*\|}.$$

This estimate is simply the application of the Gaussian heuristic, stating that the number of integer points within a body is essentially the volume of the body. It can be computed in polynomial time from the basis from which the enumeration will be started. We computed $E(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ for random bases generated as above and obtained the table of Figure 5. It confirms that a strong pre-processing should help increasing the dimension up to which SVP may be solved completely.

pre-processing	$d = 40$	$d = 45$	$d = 50$	$d = 55$	$d = 60$	$d = 65$	$d = 70$	$d = 75$
L^3	$1.0 \cdot 10^8$	$4.4 \cdot 10^9$	$1.5 \cdot 10^{14}$	$9.6 \cdot 10^{16}$	$3.0 \cdot 10^{18}$	$6.1 \cdot 10^{21}$	$2.8 \cdot 10^{27}$	$1.6 \cdot 10^{30}$
BKZ_{10}	$4.6 \cdot 10^5$	$1.2 \cdot 10^7$	$1.1 \cdot 10^8$	$1.3 \cdot 10^{10}$	$7.6 \cdot 10^{11}$	$1.7 \cdot 10^{14}$	$4.3 \cdot 10^{16}$	$1.9 \cdot 10^{19}$
BKZ_{20}	$2.4 \cdot 10^5$	$2.7 \cdot 10^6$	$3.1 \cdot 10^7$	$1.3 \cdot 10^9$	$4.1 \cdot 10^{10}$	$3.7 \cdot 10^{12}$	$6.4 \cdot 10^{13}$	$2.1 \cdot 10^{16}$
BKZ_{30}	$1.9 \cdot 10^5$	$1.6 \cdot 10^6$	$1.8 \cdot 10^7$	$3.0 \cdot 10^8$	$4.3 \cdot 10^9$	$1.1 \cdot 10^{11}$	$3.7 \cdot 10^{12}$	$1.9 \cdot 10^{14}$

Fig. 5. Value of $E(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ for randomly generated $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$.

If one is looking for vectors smaller than some prescribed B (for example if the existence of an unusually short vector is promised), then $\|\boldsymbol{b}_1\|$ may be replaced by B in the estimate. Overall, these estimates are rather crude since factors that are polynomial in the dimension should be considered as well. Furthermore, it does not take into account more elaborate techniques such as updating the norm during the enumeration, pruning [30, 31] and random sampling [29].

Open problem. One may wonder if the complexity upper bound for Kannan's SVP algorithm can be decreased further. Work under progress seems to show, by using a technique due to Ajtai [3], that it is sharp, in the sense that for all $\epsilon > 0$, we can build HKZ-reduced bases for which the number of steps of Kannan's algorithm would be at least $d^{d\left(\frac{1}{2e}-\epsilon\right)}$.

Acknowledgements. We thank Frederik Vercauteren for helpful discussions, as well as John Cannon and the University of Sydney for having hosted the second author while a large part of this work was completed.

References

1. E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Trans. Inform. Theory*, 48(8):2201–2214, 2002.
2. M. Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions (extended abstract). In *Proc. of STOC 1998*, pages 284–293. ACM, 1998.
3. M. Ajtai. The worst-case behavior of Schnorr's algorithm approximating the shortest nonzero vector in a lattice. In *Proc. of STOC 2003*, pages 396–406. ACM, 2003.
4. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. of STOC 1997*, pages 284–293. ACM, 1997.
5. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proc. STOC 2001*, pages 601–610. ACM, 2001.
6. L. Babai. On Lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.
7. J. Blömer. Closest vectors, successive minima and dual-HKZ bases of lattices. In *Proc. of ICALP 2000*, volume 1853 of *LNCS*, pages 248–259. Springer-V., 2000.
8. D. Coppersmith and A. Shamir. Lattice attacks on NTRU. In *Proc. of Eurocrypt 1997*, volume 1233 of *LNCS*, pages 52–61. Springer-V., 1997.
9. U. Fincke and M. Pohst. A procedure for determining algebraic integers of given norm. In *Proc. of EUROCAL*, volume 162 of *LNCS*, pages 194–202, 1983.
10. N. Gama, N. Howgrave-Graham, H. Koy, and P. Nguyen. Rankin's constant and blockwise lattice reduction. In *Proc. of Crypto 2006*, number 4117 in LNCS, pages 112–130. Springer-V., 2006.
11. O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *Proceedings of Crypto 1997*, volume 1294 of *LNCS*, pages 112–131. Springer-V., 1997.
12. I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *Proc. of STOC 2007*, 2007.
13. B. Helfrich. Algorithms to construct Minkowski reduced and Hermite reduced lattice bases. *Theoret. Comput. Sci.*, 41:125–139, 1985.
14. C. Hermite. Extraits de lettres de M. Hermite à M. Jacobi sur différents objets de la théorie des nombres, deuxième lettre. *J. Reine Angew. Math.*, 40:279–290, 1850.

15. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: a ring based public key cryptosystem. In *Proceedings of the 3rd Algorithmic Number Theory Symposium (ANTS III)*, volume 1423 of *LNCS*, pages 267–288. Springer-V., 1998.
16. R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proc. of STOC 1983*, pages 99–108. ACM, 1983.
17. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:513–534, 1982.
18. Magma. The Magma computational algebra system for algebra, number theory and geometry. Available at `http://magma.maths.usyd.edu.au/magma/`.
19. J. Martinet. *Perfect Lattices in Euclidean Spaces.* Springer-V., 2002.
20. J. Mazo and A. Odlyzko. Lattice points in high-dimensional spheres. *Monatsh. Math.*, 110:47–61, 1990.
21. D. Micciancio and S. Goldwasser. *Complexity of lattice problems: a cryptographic perspective.* Kluwer Academic Press, 2002.
22. H. Minkowski. *Geometrie der Zahlen.* Teubner-V., 1896.
23. P. Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto'97. In *Proc. of Crypto 1999*, volume 1666 of *LNCS*, pages 288–304. Springer-V., 1999.
24. P. Nguyen and D. Stehlé. LLL on the average. In *Proc. of ANTS VII*, volume 4076 of *LNCS*, pages 238–256. Springer-V., 2006.
25. P. Nguyen and J. Stern. Cryptanalysis of the Ajtai-Dwork cryptosystem. In *Proc. of Crypto 1998*, volume 1462 of *LNCS*, pages 223–242. Springer-V., 1998.
26. P. Nguyen and T. Vidick. Assessing sieve algorithms for the shortest vector problem. Draft, 2007.
27. 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`.
28. C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theoret. Comput. Sci.*, 53:201–224, 1987.
29. C. P. Schnorr. Lattice reduction by random sampling and birthday methods. In *Proc. of STACS 2003*, volume 2607 of *LNCS*, pages 145–156. Springer-V., 2003.
30. C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math. Programming*, 66:181–199, 1994.
31. C. P. Schnorr and H. H. Hörner. Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In *Proceedings of Eurocrypt 1995*, volume 921 of *LNCS*, pages 1–12. Springer-V., 1995.
32. V. Shoup. NTL, Number Theory C++ Library. Available at `http://www.shoup.net/ntl/`.