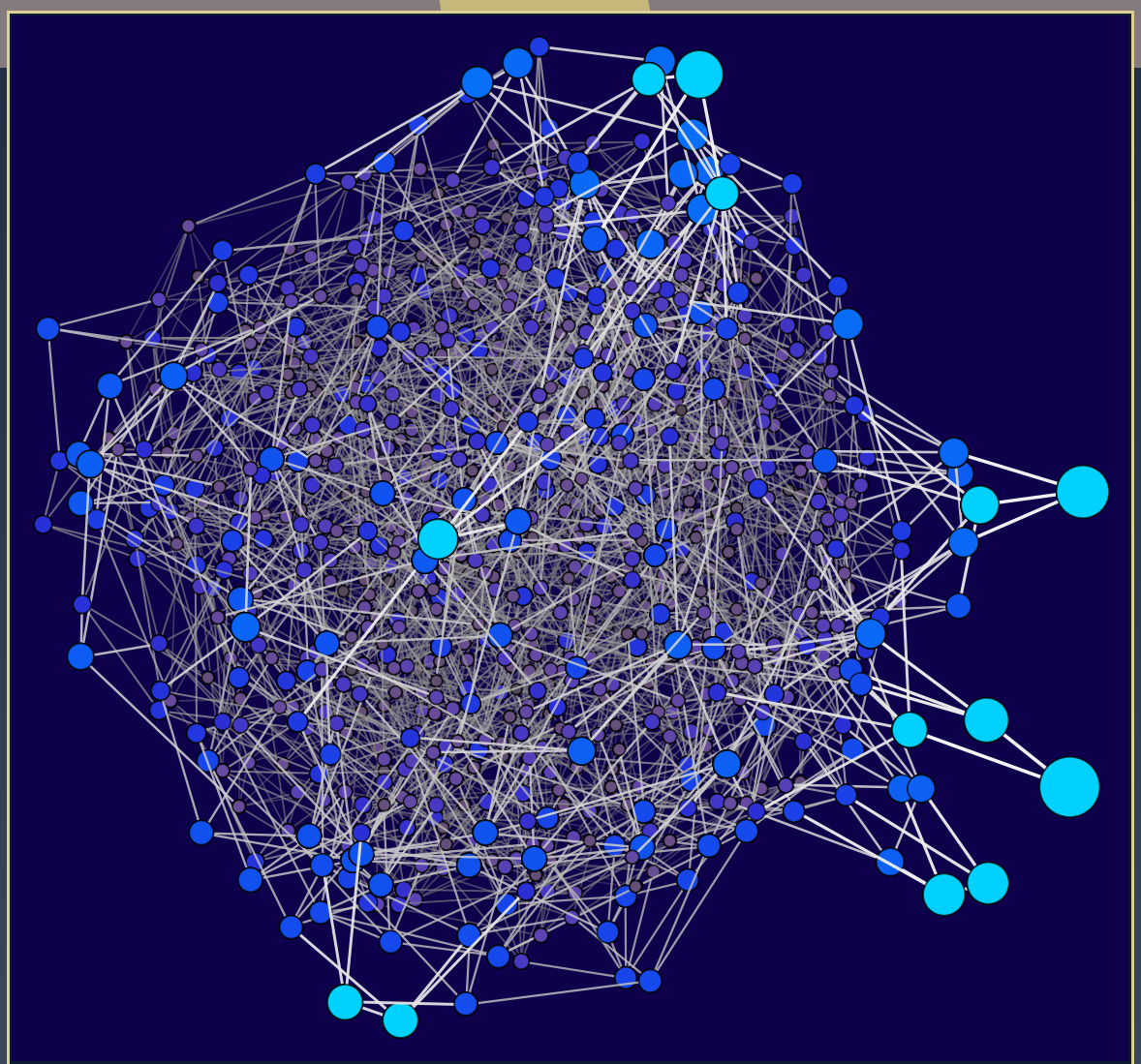
THE OPEN BOOK SERIES 4

# ANTS XIV
## Proceedings of the Fourteenth Algorithmic Number Theory Symposium

### The nearest-colattice algorithm: Time-approximation tradeoff for approx-CVP

Thomas Espitau and Paul Kirchner



msp

THE OPEN BOOK SERIES **4** (2020)
**Fourteenth Algorithmic Number Theory Symposium**
https://doi.org/10.2140/obs.2020.4.251

# The nearest-colattice algorithm: Time-approximation tradeoff for approx-CVP

Thomas Espitau and Paul Kirchner

We exhibit a hierarchy of polynomial time algorithms solving approximate variants of the closest vector problem (CVP). Our first contribution is a heuristic algorithm achieving the same distance tradeoff as HSVP algorithms, namely $\approx \beta^{n/(2\beta)}\mathrm{covol}(\Lambda)^{1/n}$ for a random lattice $\Lambda$ of rank $n$. Compared to the so-called Kannan's embedding technique, our algorithm allows the use of precomputations and can be used for efficient batch CVP instances. This implies that some attacks on lattice-based signatures lead to very cheap forgeries, after a precomputation. Our second contribution is a proven reduction from approximating the closest vector with a factor $\approx n^{3/2}\beta^{3n/(2\beta)}$ to the shortest vector problem (SVP) in dimension $\beta$.

## 1. Introduction

***Lattices, CVP, SVP.*** In a general setting, a real *lattice* $\Lambda$ is a finitely generated free $\mathbb{Z}$-module, endowed with a positive-definite quadratic form on its ambient space $\Lambda \otimes_{\mathbb{Z}} \mathbb{R}$, or equivalently is a discrete subgroup of a Euclidean space.

A fundamental lattice problem is the *closest vector problem*, or CVP for short. The goal of this problem is to find a lattice point that is closest to a given point in its ambient space. This problem is provably difficult to solve, being actually an **NP**-hard problem. It is known to be harder than the *shortest vector problem* (SVP) [19], which asks for the shortest nonzero lattice point. SVP is the cornerstone of lattice reduction algorithms (see, for instance, [33; 20; 29]). These algorithms are at the heart of lattice-based cryptography [31], and are invaluable in plenty of computational problems, including Diophantine approximation, algebraic number theory or optimization (see [30] for a survey on the applications of the LLL algorithm).

***On CVP-solving algorithms.*** There are three families of algorithms solving CVP:

*Enumeration algorithms.* These consist in recursively exploring all vectors in a set containing a closest vector. Kannan's algorithm takes time $n^{\mathrm{O}(n)}$ and polynomial space [24]. This estimate was later refined to $n^{n/2+\mathrm{o}(n)}$ by Hanrot and Stehlé [21].

*MSC2010*: 11HXX, 68W40.
*Keywords*: lattice, closest vector problem.

*Voronoi cell computation.* Micciancio and Voulgaris' Voronoi cell algorithm solves CVP in $(4+o(1))^n$ time but uses a space of $(2+o(1))^n$ [28].

*Sieving algorithms.* Here, vectors are combined in order to get closer and closer to the target vector. Heuristic variants take as little as $\left(\frac{4}{3}+o(1)\right)^{n/2}$ time [7], but proven variants of classical sieves [3; 8; 15] could only solve CVP with approximation factor $1+\epsilon$ at a cost in the exponent. In 2015, a $(2+o(1))^n$ sieve for *exact* CVP was finally proven by Aggarwal, Dadush and Stephen-Davidowitz [1] thanks to the properties of discrete Gaussians.

Many algorithms for solving the relaxed variant, APPROX-CVP, have been proposed. However, they come with caveats. For example, Dadush, Regev and Stephens-Davidowitz [10] give algorithms for this problem, but only with exponential time precomputations. Babai [5, Theorem 3.1] showed that one can reach a $2^{n/2}$-approximation factor for CVP in polynomial time. To the authors' knowledge, this has never been improved (while keeping the polynomial-time requirement), though the approximation factor for SVP has been significantly reduced [33; 20; 29].

We aim to solve the relaxed version of CVP for relatively large approximation factors, and study the tradeoff between the quality of the approximation of the solution found and the time required to actually find it. In particular, we exhibit a hierarchy of polynomial-time algorithms solving APPROX-CVP, ranging from Babai's nearest plane algorithm to an actual CVP oracle.

***Contributions and summary of the techniques.*** We introduce our so-called `Nearest-Colattice` algorithm in Section 3. Inspired by Babai's algorithm, it shows that in practice, we can achieve the performance of Kannan's embedding but with a basis which is *independent* of the target vector. Denote by $T(\beta)$ (resp. $T_{\mathrm{CVP}}(\beta)$) the time required to solve $\sqrt{\beta}$-Hermite-SVP (resp. exactly solve CVP) in rank $\beta$). Quantitatively, we show:

**Theorem 1.1** (informal). *Let $\beta > 0$ be a positive integer and $B$ be a basis of a lattice $\Lambda$ of rank $n > 2\beta$. After precomputations using a time bounded by $T(\beta)(n+\log\|B\|)^{O(1)}$, given a target $t \in \Lambda_{\mathbb{R}}$ and under a heuristic on the covering radius of a random lattice, the algorithm* `Nearest-Colattice` *finds a vector $x \in \Lambda$ such that*

$$\|x-t\| \leq \Theta(\beta)^{\frac{n}{2\beta}}\,\mathrm{covol}(\Lambda)^{\frac{1}{n}}$$

*in time $T_{\mathrm{CVP}}(\beta)(n+\log\|t\|+\log\|B\|)^{O(1)}$.*

Furthermore, the structure of the algorithms allows time-memory tradeoff and batch CVP oracle to be used.

We believe that this algorithm has been in the folklore for some time, and it is somehow hinted at in `ModFalcon`'s security analysis [9, Subsection 4.2], but without analysis of the heuristics introduced.

Our second contribution is an APPROX-CVP algorithm, which gives a time-quality tradeoff similar to the one given by the BKZ algorithm [33; 21], or variants of it [17; 2]. Note however that the approximation factor is significantly higher than the corresponding theorems for APPROX-SVP. Written as a reduction, we prove that, for a $\gamma$-HSVP oracle $\mathcal{O}$:

**Theorem 1.2** (APPROX-CVPP oracle from APPROX-SVP oracle). *Let $\Lambda$ be a lattice of rank $n$. Then one can solve the $(n^{3/2}\gamma^3)$-closest vector problem in $\Lambda$, using $2n^2$ calls to the oracle $\mathcal{O}$ during precomputation, and polynomial-time computations.*

Babai's algorithm requires that the Gram-Schmidt norms do not decrease by too much in the reduced basis. While this is true for an LLL reduced basis [26], we do not know a way to guarantee this in the general case. To overcome this difficulty, the proof technique goes as follows: first we show that it is possible to find a vector within distance $\frac{1}{2}(\sqrt{n}\gamma)\lambda_n(\Lambda)$ of the target vector, with the help of a highly-reduced basis. This is not enough, as the target can be very closed compared to $\lambda_n(\Lambda)$. We treat this peculiar case by finding a short vector in the dual lattice and then directly computing the inner product of the close vectors with our short dual vector. In the other case, Banaszczyk's transference theorem [6] guarantees that $\lambda_n(\Lambda)$ is comparable to the distance to the lattice, so that we can use our first algorithm directly.

**Remark 1.3.** Based on a result due to Kannan (see for instance [12]) that $\sqrt{n}\gamma^2$ CVP reduces to $\gamma$-SVP. Combined with the reduction from $\gamma^2$-SVP to $\gamma$-HSVP of [27], we get a polynomial time reduction from $\sqrt{n}\gamma^4$-CVP to $\gamma$-HSVP. Hence, our result is better when $n^{3/2}\gamma^3$ is at most $\sqrt{n}\gamma^4$, i.e., when $n < \gamma$.

## 2. Algebraic and computational background

In this preliminary section, we recall the notions of geometry of numbers used throughout this paper, the computational problems related to SVP and CVP, and a brief presentation of some lattice reduction algorithms solving these problems.

***Notation and conventions.***

*General notations.* $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ refer as usual to the ring of integers and the fields of rational and real numbers. Given a real number $x$, the integral roundings *floor*, *ceil* and *round to the nearest integer* are denoted respectively by $\lfloor x \rfloor$, $\lceil x \rceil$, $\lfloor x \rceil$. All logarithms are taken in base 2, unless explicitly stated otherwise.

*Computational setting.* The generic complexity model used in this work is the random-access machine (RAM) model and the computational cost is measured in operations.

### 2.1. *Euclidean lattices and their geometric invariants.*

#### 2.1.1. *Lattices.*

**Definition 2.1** (lattice). A (real) *lattice* $\Lambda$ is a finitely generated free $\mathbb{Z}$-module, endowed with a Euclidean norm $\|\cdot\|$ on the real vector space $\Lambda_{\mathbb{R}} = \Lambda \otimes_{\mathbb{Z}} \mathbb{R}$.

We may omit to write down the norm to refer to a lattice $\Lambda$ when any ambiguity is removed by the context. By definition of a finitely-generated free module, there exists a finite family $(v_1, \ldots, v_n) \in \Lambda^n$ such that $\Lambda = \bigoplus_{i=1}^{n} v_i\mathbb{Z}$, called a *basis* of $\Lambda$. Every basis has the same number of elements $\mathrm{rk}(\Lambda)$, called the rank of the lattice.

**2.1.2.** *Sublattices and quotient lattice.* Let $(\Lambda, \|\cdot\|)$ be a lattice, and let $\Lambda'$ be a submodule of $\Lambda$. Then the restriction of $\|\cdot\|$ to $\Lambda'$ endows $\Lambda$ with a lattice structure. The pair $(\Lambda', \|\cdot\|)$ is called a *sublattice* of $\Lambda$. In the remainder of this paper, we restrict ourselves to so-called *pure sublattices*, that is, those such that the quotient $\Lambda/\Lambda'$ is torsion-free. In this case, the quotient can be endowed with a canonical lattice structure by defining

$$\|v + \Lambda'\|_{\Lambda/\Lambda'} = \inf_{v' \in \Lambda'_{\mathbb{R}}} \|v - v'\|_{\Lambda}.$$

This lattice is isometric to the projection of $\Lambda$ orthogonally to the subspace of $\Lambda_{\mathbb{R}}$ spanned by $\Lambda'$.

**2.1.3.** *On effective lifting.* Given a coset $v + \Lambda'$ of the quotient $\Lambda/\Lambda'$, we might need to find a representative of this class in $\Lambda$. While any element could be theoretically taken, from an algorithmic point of view, we shall take an element of norm somewhat small, so that its coefficients remain polynomial in the input representation of the lattice. An effective solution to do so consists in using, for instance, the *Babai's rounding* or *Babai's nearest plane* algorithms. For completeness purposes we recast here the pseudo-code of such a `Lift` function using the nearest-plane procedure.

**Algorithm 1:** `Lift` (by Babai's nearest plane)

Input: A lattice basis $B = (v_1, \ldots, v_k)$ of $\Lambda'$ in $\Lambda$, a vector $t \in \Lambda_{\mathbb{R}}$.
Result: A vector of the class $\tilde{t} + \Lambda' \in \Lambda$.

Compute the Gram-Schmidt orthogonalization $(v_1^*, \ldots, v_k^*)$ of $B$
$s \leftarrow -t$
for $i = k$ downto 1 do
    $s \leftarrow s - \left\lfloor \frac{\langle s, v_i^* \rangle}{\|v_i^*\|^2} \right\rceil v_i$
return $t + s$

**2.1.4.** *Orthogonality and algebraic duality.* The *dual* lattice $\Lambda^\vee$ of a lattice $\Lambda$ is defined as the module $\mathrm{Hom}(\Lambda, \mathbb{Z})$ of integral linear forms, endowed with the derived norm defined by

$$\|\varphi\| = \inf_{v \in \Lambda_{\mathbb{R}} \setminus \{0\}} \frac{|\varphi(v)|}{\|v\|_{\Lambda}}$$

for $\varphi \in \Lambda^\vee$. By Riesz's representation theorem, it is isometric to $\{x \in \Lambda_{\mathbb{R}} \mid \langle x, v \rangle \in \mathbb{Z} \text{ for all } v \in \Lambda\}$ endowed with the dual of $\|\cdot\|_{\Lambda}$.

Let $\Lambda' \subset \Lambda$ be a sublattice. Define its *orthogonal* in $\Lambda$ to be the sublattice $\Lambda'_\perp = \{x \in \Lambda^\vee : \langle x, \Lambda' \rangle = 0\}$ of $\Lambda^\vee$. It is isometric to $(\Lambda/\Lambda')^\vee$, and by biduality $\Lambda_\perp'^\vee$ shall be identified with $\Lambda/\Lambda'$.

**2.1.5.** *Filtrations.* A filtration (or flag) of a lattice $\Lambda$ is an increasing sequence of submodules of $\Lambda$, i.e., each submodule is a proper submodule of the next: $\{0\} = \Lambda_0 \subset \Lambda_1 \subset \Lambda_2 \subset \cdots \subset \Lambda_k = \Lambda$. If we write $\mathrm{rk}(\Lambda_i) = d_i$, then we have $0 = d_0 < d_1 < d_2 < \cdots < d_k = \mathrm{rk}(\Lambda)$. A filtration is called *complete* if $d_i = i$ for all i.

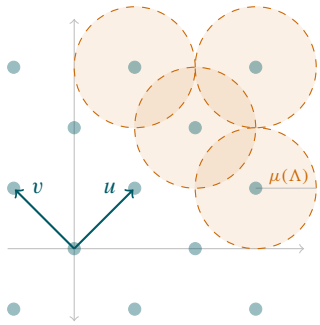



**Figure 1.** Covering radius $\mu(\Lambda)$ of a two-dimensional lattice $\Lambda$.

**2.1.6.** *Successive minima, covering radius and transference.* Let $\Lambda$ be a lattice of rank $n$. By discreteness in $\Lambda_{\mathbb{R}}$, there exists a vector of minimal norm in $\Lambda$. This parameter is called the *first minimum* of the lattice and is denoted by $\lambda_1(\Lambda)$. An equivalent way to define this invariant is to see it as the smallest positive real $r$ such that the lattice points inside a ball of radius $r$ span a space of dimension 1. This definition leads to the following generalization, known as successive minima.

**Definition 2.2** (successive minima). Let $\Lambda$ be a lattice of rank $n$. For $1 \leq i \leq n$, define the $i$-th minimum of $\Lambda$ as $\lambda_i(\Lambda) = \inf\{r \in \mathbb{R} | \dim(\mathrm{span}(\Lambda \cap B(0,r))) \geq i\}$.

**Definition 2.3.** The covering radius of a lattice $\Lambda$ or rank $n$ is defined as

$$\mu(\Lambda) = \max_{x \in \Lambda_{\mathbb{R}}} \mathrm{dist}(x, \Lambda).$$

It means that for any vector of the ambient space $x \in \Lambda_{\mathbb{R}}$ there exists a lattice point $v \in \Lambda$ at distance at most $\mu(\Lambda)$.

We now recall Banaszczyk's transference theorem, relating the extremal minima of a lattice and its dual:

**Theorem 2.4** (Banaszczyk's transference theorem [6]). *For any lattice $\Lambda$ of dimension $n$, we have*

$$1 \leq 2\lambda_1(\Lambda^\vee)\mu(\Lambda) \leq n,$$

*implying*

$$1 \leq \lambda_1(\Lambda^\vee)\lambda_n(\Lambda) \leq n.$$

## 2.2. *Computational problems in geometry of numbers.*

**2.2.1.** *The shortest vector problem.* In this section, we introduce formally the SVP problem and its variants and discuss their computational hardness.

**Definition 2.5** ($\gamma$-SVP). Let $\gamma = \gamma(n) \geq 1$. The $\gamma$-shortest vector problem ($\gamma$-SVP) is defined as follows.

**Input:** A basis $(v_1, \ldots, v_n)$ of a lattice $\Lambda$ and a target vector $t \in \Lambda_{\mathbb{R}}$.

**Output:** A lattice vector $v \in \Lambda \setminus \{0\}$ satisfying $\|v\| \leq \gamma\lambda_1(\Lambda)$.

In the case where $\gamma = 1$, the corresponding problem is simply called SVP.

**Theorem 2.6** (Haviv and Regev [22]). APPROX-SVP *is **NP**-hard under randomized reductions for every constant approximation factor.*

A variant of the problem consists of finding vectors in Hermite-like inequalities.

**Definition 2.7** ($\gamma$-HSVP). Let $\gamma = \gamma(n) \geq 1$. The $\gamma$-Hermite shortest vector problem ($\gamma$-HSVP) is defined as follows.

**Input:** A basis $(v_1, \ldots, v_n)$ of a lattice $\Lambda$.

**Output:** A lattice vector $v \in \Lambda \setminus \{0\}$ satisfying $\|v\| \leq \gamma \operatorname{covol}(\Lambda)^{1/n}$.

There exists a simple polynomial-time dimension-preserving reduction between these two problems, as stated by Lovász in [27, 1.2.20]:

**Theorem 2.8.** *One can solve $\gamma^2$-SVP using $2n$ calls to a $\gamma$-HSVP oracle and polynomial time.*

This can be slightly improved where the HSVP oracle is built from an HSVP oracle in lower dimension [2].

**2.2.2.** *An oracle for $\gamma$-HSVP.* We note a function $T(\beta)$ such that we can solve $\mathrm{O}(\sqrt{\beta})$-HSVP in time at most $T(\beta)$ times the input size. We have the following bounds on $T$, depending on if we are looking at an algorithm which is:

*deterministic*: $T(\beta) = (4 + \mathrm{o}(1))^{\beta/2}$, proven by Micciancio and Voulgaris in[28];

*randomized*: $T(\beta) = (4/3 + \mathrm{o}(1))^{\beta/2}$ , introduced by Wei, Liu and Wang in [36];

*heuristic*: $T(\beta) = (3/2 + \mathrm{o}(1))^{\beta/2}$, given in [7] by Becker, Ducas, Gama, Laarhoven.

There also exist variants for quantum computers [25], and time-memory tradeoffs, such as [23]. By providing a back-and-forth strategy coupled with enumeration in the dual lattice, the *self dual block Korkine-Zolotarev* (DBKZ) algorithm provides an algorithm better than the famous BKZ algorithm.

**Theorem 2.9** (Micciancio and Walter [29]). *There exists an algorithm outputting a vector $v$ of a lattice $\Lambda$ satisfying*

$$\|v\| \leq \beta^{\frac{n-1}{2(\beta-1)}} \cdot \operatorname{covol}(\Lambda)^{\frac{1}{n}}.$$

*Such a bound can be achieved in time $(n + \log \|B\|)^{\mathrm{O}(1)} T(\beta)$, where $B$ is the integer input basis representing $\Lambda$.*

*Proof.* The bound we get is a direct consequence of [29, Theorem 1]. We only replaced the *Hermite constant* $\gamma_\beta$ by an upper bound in $\mathrm{O}(\beta)$. □

A stronger variant of this estimate is heuristically true, at least for "random" lattices, as it is suggested by the Gaussian heuristic in [29, Corollary 2]. Under this assumption, one can bound not only the length of the first vector but also the gap between the covolumes of the filtration induced by the outputted basis.

**Theorem 2.10.** *There exists an algorithm outputting a complete filtration of a lattice $\Lambda$ satisfying*:

$$\operatorname{covol}(\Lambda_i/\Lambda_{i-1}) \approx \Theta(\beta)^{\frac{n+1-2i}{2(\beta-1)}} \operatorname{covol}(\Lambda)^{\frac{1}{n}}.$$

*Such a bound can be achieved in time $(n+\log\|B\|)^{O(1)}T(\beta)$, where $B$ is the integer-valued input basis. Further, we have*

$$\Theta(\sqrt{\beta}) \operatorname{covol}^{\frac{1}{\beta}}(\Lambda_n/\Lambda_{n-\beta}) \approx \operatorname{covol}(\Lambda_{n-\beta+1}/\Lambda_{n-\beta}).$$

**2.3.** ***The closest vector problem.*** In this section we introduce formally the CVP problem and its variants and discuss their computational hardness.

**Definition 2.11** ($\gamma$-CVP). Let $\gamma = \gamma(n) \geq 1$. The $\gamma$-closest vector problem ($\gamma$-CVP) is defined as follows.

**Input:** A basis $(v_1, \ldots, v_n)$ of a lattice $\Lambda$ and a target vector $t \in \Lambda \otimes \mathbb{R}$.

**Output:** A lattice vector $v \in \Lambda$ satisfying $\|x - t\| \leq \gamma \min_{v\in\Lambda} \|v - t\|$.

In the case where $\gamma = 1$, the corresponding problem is called CVP.

**Theorem 2.12** (Dinur, Kindler and Shafra [11]). $n^{c/(\log\log n)}$-APPROX-CVP *is* ***NP****-hard for any $c > 0$.*

We let $T_{\mathrm{CVP}}(\beta)$ be such that we can solve CVP in dimension $\beta$ in running time bounded by $T_{\mathrm{CVP}}(\beta)$ times the size of the input. Hanrot and Stehlé proved $\beta^{\beta/2+o(\beta)}$ with polynomial memory [21]. Sieves can provably reach $(2+o(1))^{\beta}$ with exponential memory [1]. More importantly for this paper, heuristic sieves can reach $(4/3+o(1))^{\beta/2}$ for solving an entire batch of $2^{0.058\beta}$ instances [13].

## 3. The nearest colattice algorithm

We aim to solve the $\gamma$-APPROX-CVP by recursively exploiting the datum of a filtration

$$\Lambda_0 \subset \Lambda_1 \subset \cdots \subset \Lambda_k = \Lambda$$

via recursive approximations. The central object used during this reduction is the *nearest colattice* relative to a target vector.

In this section, and the next one, we assume that the size of the bases is always small, essentially as small as the input basis. This is classic, and can be easily proven.

### 3.1. *Nearest colattice to a vector.*

**Definition 3.1.** Let $0 \to \Lambda' \to \Lambda \to \Lambda/\Lambda' \to 0$ be a short exact sequence of lattices, and set $t \in \Lambda_{\mathbb{R}}$ to be a target vector. A nearest $\Lambda'$-colattice to $t$ is a coset $\overline{v} = v + \Lambda' \in \Lambda/\Lambda'$ which is the closest to the projection of $t$ in $\Lambda_{\mathbb{R}}/\Lambda'_{\mathbb{R}}$, i.e., such that $\overline{v} = \operatorname{argmin}_{v\in\Lambda} \|(t-v)+\Lambda'\|_{\Lambda_{\mathbb{R}}/\Lambda'_{\mathbb{R}}}$.

This definition makes sense thanks to the discreteness of the quotient lattice $\Lambda/\Lambda'$ in the real vector space $\Lambda_{\mathbb{R}}/\Lambda'_{\mathbb{R}}$.

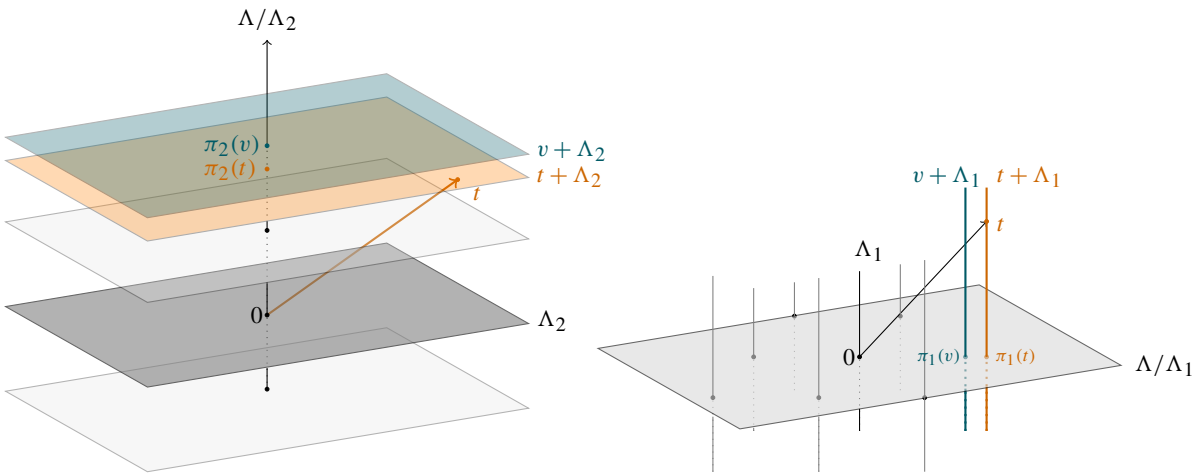



**Figure 2.** The $\Lambda_2$-nearest colattice $v + \Lambda_2$ relative to $t$, in green (left). The $\Lambda_1$-nearest colattice $v + \Lambda_1$ relative to $t$ (right).

**Example.** To illustrate this definition, we give two examples in dimension 3, of rank 1 and 2 nearest colattices. Set $\Lambda$ to be a rank 3 lattice, and fix $\Lambda_1$ and $\Lambda_2$ to be two pure sublattices of respective ranks 1 and 2. Denote by $\pi_i$ the canonical projection onto the quotient $\Lambda/\Lambda_i$, which is of dimension $3 - i$ for $i \in \{1, 2\}$. The $\Lambda_i$-closest colattice to $t$, denoted by $v_i + \Lambda_i$, is such that $\pi_i(v_i)$ is a closest vector to $\pi_i(t)$ in the corresponding quotient lattice. Figure 2 (left) and (right), respectively, depict these situations.

**Remark 3.2.** A computational insight into Definition 3.1 is given by viewing a nearest colattice as a solution to an instance of exact-CVP in the quotient lattice $\Lambda/\Lambda'$.

Taking the same notation as in Definition 3.1, let us project $t$ orthogonally onto the affine space $v + \Lambda'_{\mathbb{R}}$, and take $w$ to be a closest vector to this projection. The vector $w$ is then relatively close to $t$. Let us quantify its defect of closeness towards an actual closest vector to $t$:

**Proposition 3.3.** *With the same notation as above*: $\|t - w\|^2 \leq \mu(\Lambda/\Lambda')^2 + \mu(\Lambda')^2$.

*Proof.* This is clear by Pythagoras' theorem. □

By definition of the covering radius, we then have:

**Corollary 3.4** (subadditivity of the covering radius over short exact sequences). *Let*

$$0 \to \Lambda' \to \Lambda \to \Lambda/\Lambda' \to 0$$

*be a short exact sequence of lattices. Then we have* $\mu(\Lambda)^2 \leq \mu(\Lambda/\Lambda')^2 + \mu(\Lambda')^2$.

This inequality is tight, and is an equality when there exists a sublattice $\Lambda''$ such that $\Lambda' \oplus \Lambda'' = \Lambda$ and $\Lambda'' \subseteq \Lambda'^{\perp}$.

**3.2.** ***Recursion along a filtration.*** Let us now consider a filtration $\Lambda_0 \subset \Lambda_1 \subset \cdots \subset \Lambda_k = \Lambda$ and a target vector $t \in \Lambda_{\mathbb{R}}$. Repeatedly applying Corollary 3.4 along the subfiltrations $0 \subset \Lambda_i \subset \Lambda_{i+1}$, yields a sequence of inequalities $\mu(\Lambda_{i+1})^2 - \mu(\Lambda_i)^2 \leq \mu(\Lambda_{i+1}/\Lambda_i)^2$. The telescoping sum now gives the relation $\mu(\Lambda)^2 \leq \sum_{i=1}^{k} \mu(\Lambda_{i+1}/\Lambda_i)^2$. This formula has a very natural algorithmic interpretation as a recursive oracle for approx-CVP:

(1) Starting from the target vector $t$, we solve the CVP instance corresponding to $\pi(t)$ in the quotient $\Lambda_k/\Lambda_{k-1}$ with $\pi$ the canonical projection onto this quotient to find $v + \Lambda_{k-1}$, the nearest $\Lambda_{k-1}$-colattice to $t$.

(2) We then project $t$ orthogonally onto $v + (\Lambda_{k-1} \otimes_{\mathbb{Z}} \mathbb{R})$. Call this vector $t'$.

(3) A recursive call to the algorithm on the instance $(t' - v, \Lambda_0 \subset \cdots \subset \Lambda_{k-1}))$ yields a vector $w \in \Lambda_2$.

(4) Return $w + v$.

Its translation in pseudo-code is given in an iterative manner in the algorithm `Nearest-Colattice`.

**Algorithm 2:** `Nearest-Colattice`

Input: A filtration $\{0\} = \Lambda_0 \subset \Lambda_1 \subset \cdots \subset \Lambda_k = \Lambda$, a target $t \in \Lambda_{\mathbb{R}}$.
Result: A vector in $\Lambda$ close to $t$.

```
s ← −t
for i = k downto 1 do
    s ← s − Lift(argmin_{h∈Λ_i/Λ_{i−1}} ‖v − h‖)
return t + s
```

**Proposition 3.5.** *Let $B$ be a basis of a lattice $\Lambda$ of rank $n$. Given a target $t \in \Lambda_{\mathbb{R}}$, the algorithm* `Nearest-Colattice` *finds a vector $x \in \Lambda$ such that $\|x - t\|^2 \leq \sum_{i=1}^{k} \mu(\Lambda_{i+1}/\Lambda_i)^2$ in time*

$$T_{\mathrm{CVP}}(\beta)(n + \log \|t\| + \log \|B\|)^{\mathrm{O}(1)},$$

*where $\beta$ is the largest gap of rank in the filtration $\beta = \max_i(\mathrm{rk}(\Lambda_{i+1}) - \mathrm{rk}(\Lambda_i))$.*

*Proof.* The bound on the quality of the approximation is a direct consequence of the previous discussion. The running time bound derives from the definition of $T_{\mathrm{CVP}}$ and the fact that the `Lift` operations can be conducted in polynomial time. □

**Remark 3.6** (retrieving Babai's algorithm). In the specific case where the filtration is complete, that is to say that $\mathrm{rk}(\Lambda_i) = i$ for each $1 \leq i \leq n$, the `Nearest-Colattice` algorithm coincides with the so-called *Babai's nearest plane* algorithm. In particular, it recovers a vector at distance

$$\sqrt{\sum_{i=1}^{n} \mu(\Lambda_i/\Lambda_{i-1})^2} = \frac{1}{2}\sqrt{\sum_{i=1}^{n} \mathrm{covol}(\Lambda_i/\Lambda_{i-1})^2},$$

since for each index $i$, we have $\mu(\Lambda_i/\Lambda_{i-1}) = \frac{1}{2}\,\mathrm{covol}(\Lambda_i/\Lambda_{i-1})$ as these quotients are one-dimensional.

The bound given in Proposition 3.5 is not easily instantiable as it requires having access to the covering radius of the successive quotients of the filtration. However, under a mild heuristic on random lattices, we now exhibit a bound which only depends on the parameter $\beta$ and the covolume of $\Lambda$.

**3.3.** ***On the covering radius of a random lattice.*** In this section we prove that the covering radius of a random lattice behaves essentially in $\sqrt{\mathrm{rk}(\Lambda)}$.

In 1945, Siegel [34] proved that the projection of the Haar measure of $\mathrm{SL}_n(\mathbb{R})$ over the quotient $\mathrm{SL}_n(\mathbb{R})/\mathrm{SL}_n(\mathbb{Z})$ is of finite mass, yielding a natural probability distribution $\nu_n$ over the moduli space $\mathcal{L}_n$ of unit-volume lattices. By construction this distribution is translation-invariant, that is, for any measurable set $\mathcal{S} \subseteq \mathcal{L}_n$ and all $U \in \mathrm{SL}_n(\mathbb{Z})$, we have $\nu_n(\mathcal{S}) = \nu_n(\mathcal{S}U)$. A *random lattice* is then defined as a unit-covolume lattice in $\mathbb{R}^n$ drawn under the probability distribution $\nu_n$.

We first recall an estimate due to Rogers [32], giving the expectation[1] of the number of lattice points in a fixed set.

**Theorem 3.7** (Rogers' average). *Let $n \leq 4$ be an integer and $\rho$ be the characteristic function of a Borel set $C$ of $\mathbb{R}^n$ whose volume is $V$, centered at 0. Then*:

$$0 \leq \int_{\mathcal{L}_n} \rho(\Lambda \setminus \{0\})\, d\nu_n(\Lambda) - 2e^{-V/2} \sum_{r=0}^{\infty} \frac{r}{r!}(V/2)^r \leq (V+1)\left(6\left(\sqrt{\tfrac{3}{4}}\right)^n + 105 \cdot 2^{-n}\right).$$

This allows us to prove that the first minimum of a random lattice is greater than a multiple of $\sqrt{n}$.

**Lemma 4.** *Let $\Lambda$ be a random lattice of rank $n$. Then, with probability $1 - 2^{-\Omega(n)}$, $\lambda_1(\Lambda) > c\sqrt{n}$ for a universal constant $c > 0$.*

*Proof.* Consider the ball $C$ of volume $V = 0.99^n$. Its radius is equal to $0.99\pi^{-1/2}\Gamma\left(\frac{n}{2}+1\right)^{1/n}$, which is lower bounded by $c\sqrt{n}$ for a constant $c > 0$, using for instance Stirling's estimate. By Theorem 3.7, the expectation of the number of lattice points in $C$ is at most

$$128\left(\tfrac{3}{4}\right)^{\frac{n}{2}}(V+1) + V \in (1 + \mathrm{o}(1))V.$$

This estimate upper bounds the probability that there exists a nonzero lattice vector in $C$ by $2^{-\Omega(n)}$, using Markov's inequality on the positive random variable $|\Lambda \cap C|$. □

Using the transference theorem, we then derive the following estimate on the covering radius of a random lattice:

**Theorem 4.1.** *Let $\Lambda$ be a random lattice of rank $n$. Then, with probability $1 - 2^{-\Omega(n)}$, $\mu(\Lambda) < d\sqrt{n}$ for a universal constant $d$.*

*Proof.* First note that the dual lattice $\Lambda^\vee$ follows the same distribution as $\Lambda$. Hence, using the estimate of Lemma 4, we know that with probability $1 - 2^{-\Omega(n)}$, $\lambda_1(\Lambda^\vee) > c\sqrt{n}$. Banaszczyk's transference

[1] The result proved by Rogers is actually more general and bounds all the moments of the enumerator of lattice points. For the purpose of this work, only the first moment is actually required.

theorem indicates that in this case,

$$\mu(\Lambda) \leq \frac{n}{\lambda_1(\Lambda^\vee)} \leq \frac{\sqrt{n}}{c},$$

concluding the proof. □

This justifies the following heuristic:

**Heuristic 4.2.** In algorithm `Nearest-Colattice`, for any index $i$, we have $\mu(\Lambda_{i+1}/\Lambda_i) \leq c\lambda_1(\Lambda_{i+1}/\Lambda_i)$ for some universal constant $c$.

The Gaussian heuristic suggests that "almost all" targets $t$ are at distance $(1+o(1))\lambda_1(\Lambda)$, so that for practical purposes in the analysis we can take $c = 1$ in Heuristic 4.2.

### 4.1. *Quality of the algorithm on random lattices.*

**Theorem 4.3.** *Let $\beta > 0$ be a positive integer and $B$ be a basis of a lattice $\Lambda$ of rank $n > 2\beta$. After precomputations using a time bounded by $T(\beta)(n + \log \|B\|)^{O(1)}$, given a target $t \in \Lambda_{\mathbb{R}}$ and under Heuristic 4.2, the algorithm* `Nearest-Colattice` *finds a vector $x \in \Lambda$ such that*

$$\|x - t\| \leq \Theta(\beta)^{\frac{n}{2\beta}} \operatorname{covol}(\Lambda)^{\frac{1}{n}}$$

*in time $T_{\mathrm{CVP}}(\beta)\,\mathrm{Poly}(n, \log\|t\|, \log\|B\|)$.*

*Proof.* We start by reducing the basis $B$ of $\Lambda$ using the DBKZ algorithm, and collect the vectors in blocks of size $\beta$, giving a filtration

$$\{0\} = \Lambda_0 \subset \Lambda_1 \subset \cdots \subset \Lambda_k = \Lambda,$$

for $k = \lceil \frac{n}{\beta} \rceil$ and $\mathrm{rk}(\Lambda_{i+1}/\Lambda_i) = \beta$ for each index $i$ except the penultimate one, of rank $n - \beta\lfloor \frac{n}{\beta} \rfloor$. We define $l_i$ as $\mathrm{rk}(\Lambda_{i+1}/\Lambda_i)$. By Theorem 2.10 and finite induction in each block using the multiplicativity of the covolume over short exact sequences, we have for $i < k-1$,

$$\begin{aligned}\operatorname{covol}(\Lambda_{i+1}/\Lambda_i)^{\frac{1}{l_i}} &\approx \operatorname{covol}(\Lambda)^{\frac{1}{n}} \left( \prod_{j=i\beta}^{i\beta+l_i-1} \Theta(\beta)^{\frac{n+1-2j}{2(\beta-1)}} \right)^{\frac{1}{l_i}} \\ &= \Theta(\beta)^{\frac{n+2-2i\beta-l_i}{2(\beta-1)}} \operatorname{covol}(\Lambda)^{\frac{1}{n}}.\end{aligned}$$

We also have

$$\Theta(\sqrt{\beta}) \operatorname{covol}(\Lambda_k/\Lambda_{k-1})^{1/\beta} \approx \Theta(\beta)^{\frac{n+1-2(n-\beta)}{2(\beta-1)}} \operatorname{covol}^{\frac{1}{n}} \Lambda$$

so that the previous approximation is also true for $i = k-1$. Using Heuristic 4.2 and Minkowski's first theorem, we can estimate the covering radius of this quotient as

$$\mu(\Lambda_{i+1}/\Lambda_i) \leq \Theta(\sqrt{l_i})\Theta(\beta)^{\frac{n+2-2i\beta-l_i}{2(\beta-1)}} \operatorname{covol}^{\frac{1}{n}} \Lambda.$$

Proposition 3.5 now asserts that `Nearest-Colattice` returns a vector at distance from $t$ bounded by

$$\mathrm{covol}(\Lambda)^{\frac{1}{n}} \sum_{i=0}^{k} \Theta(\sqrt{l_i})\Theta(\beta)^{\frac{n+2-2i\beta-l_i}{2(\beta-1)}} = \Theta(\beta)^{\frac{n}{2\beta-2}}\,\mathrm{covol}(\Lambda)^{\frac{1}{n}}$$

where the last equality stems from the condition $n \geq 2\beta$, so that only the first term is significant. □

Note that in the algorithm, all lattices depend only on $\Lambda$, not on the targets. Therefore, it is possible to use CVP algorithms after precomputations. These algorithms are significantly faster; we refer to [13] for heuristic ones and to [10; 35] for proven approximation algorithms.

## 5. Proven APPROX-CVP algorithm with precomputation

In all of this section, let us fix an oracle $\mathcal{O}$, solving the $\gamma$-HSVP. We solve APPROX-CVP with preprocessing from the oracle $\mathcal{O}$.

**Theorem 5.1** (APPROX-CVPP oracle from HSVP oracle). *Let $\Lambda$ be a lattice of rank $n$. Then one can solve the $(n^{3/2}\gamma^3)$-closest vector problem in $\Lambda$, using $2n^2$ calls to the oracle $\mathcal{O}$ during precomputation, and polynomial time computations.*

The first step of this reduction consists in proving that we can find a lattice point at a distance roughly $\lambda_n(\Lambda)$.

**Theorem 5.2.** *Let $\Lambda$ be a lattice of rank $n$ and $t \in \Lambda \otimes \mathbb{R}$ a target vector; then one can find a lattice vector $c \in \Lambda$ satisfying $\|c - t\| \leq \frac{1}{2}\sqrt{n}\gamma\lambda_n(\Lambda)$, using $n$ calls to the oracle $\mathcal{O}$ during precomputation, and polynomial time computations.*

*Proof.* We aim to construct a complete filtration $\{0\} \subset \Lambda_1 \subset \cdots \subset \Lambda_n = \Lambda$ of the input lattice $\Lambda$ such that for any index $1 \leq i \leq n-1$, we have $\mathrm{covol}(\Lambda_i/\Lambda_{i-1}) \leq \gamma\lambda_n(\Lambda)$. We proceed inductively:

- By a call to the oracle $\mathcal{O}$ on the lattice $\Lambda$, we find a vector $b_1$. Set $\Lambda_1 = b_1\mathbb{Z}$ to be the corresponding sublattice.
- Suppose that the filtration is constructed up to index $i$. Then we call the oracle $\mathcal{O}$ on the quotient sublattice $\Lambda/\Lambda_i$ (or equivalently on the projection of $\Lambda$ orthogonally to $\Lambda_i$), and lift the returned vector using the `Lift` function in $v \in \Lambda$. Eventually we set $\Lambda_{i+1} = \Lambda_i \oplus v\mathbb{Z}$.

At each index, we have by construction $\lambda_{n-i+1}(\Lambda/\Lambda_i) \leq \lambda_n(\Lambda)$. As such, $\mathrm{covol}(\Lambda/\Lambda_i) \leq \lambda_n(\Lambda)^{n-i+1}$, and, eventually, we have, for each index $i$,

$$\mathrm{covol}(\Lambda_i/\Lambda_{i-1}) \leq \gamma \cdot \lambda_n(\Lambda).$$

As stated in Remark 3.6, Babai's algorithm on the point $t$ returns a lattice vector $c \in \Lambda$ such that $\|c - t\| \leq \sqrt{\sum_{i=1}^{n} \mu(\Lambda_i/\Lambda_{i-1})^2} \leq \frac{1}{2}(\sqrt{n}\gamma\lambda_n(\Lambda))$. □

**Remark 5.3** (on the quality of this decoding). For a random lattice, we expect $\lambda_n(\Lambda) \approx \sqrt{n}\,\mathrm{covol}(\Lambda)^{1/n}$, so that the distance between the decoded vector and the target is only a factor $\gamma$ times larger than the guaranteed output of the oracle.

We can now complete the reduction:

*Proof of Theorem 5.1.* Let $\Lambda$ be a rank $n$ lattice. Without loss of generality, we might assume that the norm $\|\cdot\|$ of $\Lambda$ coincides with its dual norm, so that the dual $\Lambda^\vee$ can be isometrically embedded in $\Lambda_{\mathbb{R}}$. We first find a nonzero vector in the dual lattice $c \in \Lambda^\vee$, where $\|c\| \leq \gamma^2 \lambda_1(\Lambda^\vee)$ using Lovász's reduction stated in Theorem 2.8 on the oracle $\mathcal{O}$. Define $v \in \Lambda$ and $e \in \Lambda \otimes \mathbb{R}$ to satisfy $t = v + e$ with $\|e\|$ minimal. We now have two cases, depending on how large the error term $e$ is:

*Case* $\|c\|\|e\| \geq \frac{1}{2}$ (*large case*): Then, by plugging Banaszczyk's transference inequality to the bound on $\|c\|$, we get

$$\|e\| \geq \frac{1}{2\gamma^2 \lambda_1(\Lambda^\vee)} \geq \frac{\lambda_n(\Lambda)}{2n\gamma^2}.$$

Thus, we can use Theorem 5.2 to solve APPROX-CVP with approximation factor equal to

$$\frac{\sqrt{n}\gamma}{2}\left(\frac{1}{2n\gamma^2}\right)^{-1} = n^{\frac{3}{2}}\gamma^3.$$

*Case* $\|c\|\|e\| < \frac{1}{2}$ (*small case*): Then, we have by linearity, $\langle c, t\rangle = \langle c, v\rangle + \langle c, e\rangle$. Hence, by the Cauchy–Schwarz inequality and the assumption on $\|c\|\|e\|$ we can assert that

$$\lfloor \langle c, t\rangle \rceil = \langle c, v\rangle.$$

Let $\Lambda'$ be the projection of $\Lambda$ over the orthogonal space to $c$ and denote by $\pi$ the corresponding orthogonal projection.

Let us prove that $\pi(v)$ is a closest vector of $\pi(t)$ in $\Lambda'$. To do so, let us take $\tilde{p}$ a shortest vector $\pi(t)$ in $\Lambda$. We now look at the fiber (in $\Lambda$) above $\tilde{p}$ and take the closest element $p$ to $t$ in this set. Then by Pythagoras' theorem, $p$ is an element of the intersection of $\pi^{-1}(\tilde{p})$ with the convex body $\mathcal{D} = \left\{x \mid |\langle c, x\rangle| < \frac{1}{2}\right\}$. As the vector $c$ belongs to the dual of $\Lambda$, we have that for any $p_1, p_2 \in \pi^{-1}(\tilde{p})$, $\langle p_1 - p_2, c\rangle \in \mathbb{Z}$, so that $\pi^{-1}(\tilde{p}) \cap \mathcal{D}$ is of cardinality one. Write $p$ for this point. Then, $\langle p, c\rangle = \langle v, c\rangle$, as $|\langle p - v, c\rangle| < \frac{1}{2}$ and is an integer. Now remark that by minimality of $\|v - t\|$, we have by Pythagoras' theorem that $v = p$, implying that $\pi(v) = \tilde{p}$.

By induction, we find $w \in \Lambda$ such that $\|\pi(w-t)\| \leq n^{3/2}\gamma^3\|\pi(v-t)\|$ and since $\langle c, w - t\rangle = \langle c, v - t\rangle$ we obtain $\|w - t\| \leq n^{3/2}\gamma^3\|v - t\|$. □

Overall, we get the following corollary by using the Micciancio-Voulgaris algorithm for the oracle $\mathcal{O}$:

**Corollary 5.4.** *We can solve* $\beta^{\mathrm{O}(n/\beta)}$-APPROX-CVP *deterministically in time bounded by* $2^\beta$ *times the size of the input.*

**Remark 5.5.** Using exactly the same proof scheme, we can refine the approximation factor to an $n^{3/2}\gamma_S\gamma$ by using a separate $\gamma_S$-SVP oracle instead of using $\gamma$-HSVP as a $\gamma^2$-SVP oracle.

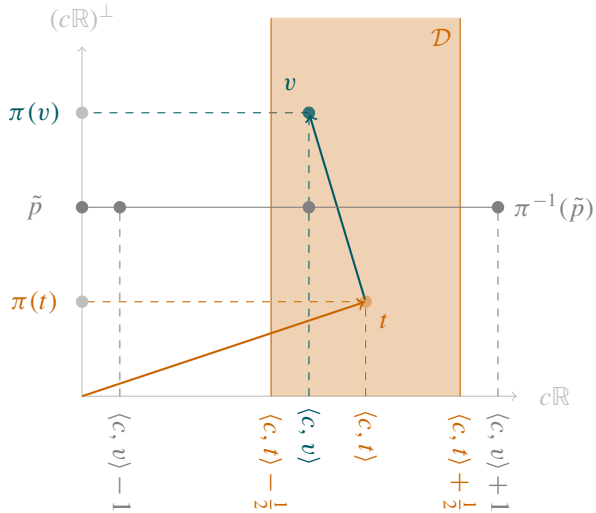



**Figure 3.** Illustration of the situation depicted in the proof, in the two-dimensional case.

## 6. Cryptographic perspectives

In cryptography, the bounded distance decoding (BDD) problem[2] has a lot of importance, as it directly relates to the celebrated learning with error (LWE) problem [31]. This latter problem can be reduced to APPROX-CVP, but our theoretical reduction with HSVP has a loss which is too large to be competitive.

In the so-called GPV framework [18], instantiated in the DLP cryptosystem [14] and its follow-ups FALCON [16], MODFALCON [9], a valid signature is a point close to a target, which is the hash of the message. Hence, forging a signature boils down to finding a close vector to a random target. Our first (heuristic) result implies that, once a reduced basis has been found, forging a message is relatively easy. Previous methods such as in [16] used Kannan's embedding [24] so that the cost given only applies for one forgery, whereas a batch forgery is possible for roughly the same cost.

The same remark applies for practically solving the BDD problem, and indeed the LWE problem. Once a highly reduced basis is found, it is enough to compute a CVP on the tail of the basis, and finish with Babai's algorithm. More precisely, by using the same notation and exploiting the proof of Theorem 4.3, a sufficient condition for decoding will be

$$\|\pi(e)\| \leq \theta(\beta)^{\frac{2\beta-n}{2\beta}} \operatorname{covol}(\Lambda)^{\frac{1}{n}},$$

where, $\pi$ is the orthogonal projection onto $\Lambda/\Lambda_k$ and $\beta$ is the rank of this latter lattice.

This trick seems to have been in the folklore for some time, and is the reason given by NEWHOPE [4] designers for selecting a random "$a$", which corresponds to a random lattice (where the authors of [4] claim that Babai's algorithm is enough, but it seems to be practically true in general for an extremely well reduced basis, i.e., with more precomputations performed).

[2]This problem being defined as finding the closest lattice vector of a target, provided it is within a fraction of $\lambda_1(\Lambda)$.

## Acknowledgments

This work was done while the authors were visiting the Simons Institute for the theory of computing in February 2020. They also thanks the anonymous reviewers for their insightful comments on this work.

## References

[1] D. Aggarwal, D. Dadush, and N. Stephens-Davidowitz. Solving the closest vector problem in $2^n$ time - the discrete Gaussian strikes again! In *56th FOCS*. IEEE Computer Society Press. 2015.

[2] D. Aggarwal, J. Li, P. Q. Nguyen, and N. Stephens-Davidowitz. Slide reduction, revisited—filling the gaps in SVP approximation. *arXiv preprint arXiv:1908.03724*, 2019.

[3] M. Ajtai, R. Kumar, and D. Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In *Proceedings 17th IEEE Annual Conference on Computational Complexity*. IEEE, 2002.

[4] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange - A new hope. In *USENIX Security 2016*.

[5] L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1), 1986.

[6] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1), 1993.

[7] A. Becker, L. Ducas, N. Gama, and T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *27th SODA*. ACM-SIAM, 2016.

[8] J. Blömer and S. Naewe. Sampling methods for shortest vectors, closest vectors and successive minima. *Theoretical Computer Science*, 410(18) 2009.

[9] C. Chuengsatiansup, T. Prest, D. Stehlé, A. Wallet, and K. Xagawa. Modfalcon: compact signatures based on module NTRU lattices. *IACR Cryptology ePrint Archive*, 2019.

[10] D. Dadush, O. Regev, and N. Stephens-Davidowitz. On the closest vector problem with a distance guarantee. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, IEEE, 2014.

[11] I. Dinur, G. Kindler, and S. Safra. Approximating-CVP to within almost-polynomial factors is np-hard. In *Proceedings 39th Annual Symposium on Foundations of Computer Science*. IEEE, 1998.

[12] C. Dubey, and T .Holenstein. Approximating the closest vector problem using an approximate shortest vector oracle Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques. 2011

[13] L. Ducas, T. Laarhoven, and W. P. van Woerden. The randomized slicer for CVPP: sharper, faster, smaller, batchier. Cryptology ePrint, Report 2020/120.

[14] L. Ducas, V. Lyubashevsky, and T. Prest. Efficient identity-based encryption over NTRU lattices. In *ASIACRYPT 2014*. Springer 2014.

[15] F. Eisenbrand, N. Hähnle, and M. Niemeier. Covering cubes and the closest vector problem. In *the 27th symposium on Computational geometry*, 2011.

[16] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. *Submission to the NIST's post-quantum cryptography standardization process*, 2018.

[17] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. In *40th ACM STOC*. ACM Press, 2008.

[18] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *40th ACM STOC*. ACM Press, 2008.

[19] O. Goldreich, D. Micciancio, S. Safra, and J.-P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2) 1999.

[20] G. Hanrot, X. Pujol, and D. Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In *CRYPTO 2011*. Springer, 2011.

[21] G. Hanrot and D. Stehlé. Improved analysis of Kannan's shortest lattice vector algorithm. In *CRYPTO 2007*. Springer, 2007.

[22] I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *39th ACM STOC*. ACM Press, 2007.

[23] G. Herold, E. Kirshanova, and T. Laarhoven. Speed-ups and time-memory trade-offs for tuple lattice sieving. In *PKC 2018*. Springer, 2018.

[24] R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of operations research*, 12(3) 1987.

[25] T. Laarhoven, M. Mosca, and J. Van De Pol. Finding shortest lattice vectors faster using quantum search. *Designs, Codes and Cryptography*, 77(2-3) 2015.

[26] A. K. Lenstra, H. W. J. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261 1982.

[27] L. Lovász. *An algorithmic theory of numbers, graphs, and convexity.* SIAM, 1986.

[28] D. Micciancio and P. Voulgaris. Faster exponential time algorithms for the shortest vector problem. In *21st SODA*. ACM-SIAM, 2010.

[29] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. In *EUROCRYPT 2016*. Springer, 2016.

[30] P. Q. Nguyen and B. Vallée. *The LLL algorithm*. Springer, 2010.

[31] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6) 2009.

[32] C. A. Rogers et al. Mean values over the space of lattices. *Acta mathematica*, 94 1955.

[33] C. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53 1987.

[34] C. L. Siegel. A mean value theorem in Geometry of Numbers. *Annals of Mathematics*, 46(2) 1945.

[35] N. Stephens-Davidowitz. A time-distance trade-off for GDD with preprocessing—instantiating the DLW heuristic. *preprint* arXiv:1902.08340, 2019.

[36] W. Wei, M. Liu, and X. Wang. Finding shortest lattice vectors in the presence of gaps. In *CT-RSA 2015*. Springer, 2015.

Received 28 Feb 2020. Revised 28 Feb 2020.

THOMAS ESPITAU: t.espitau@gmail.com
*NTT Corporation, Tokyo, Japan*

PAUL KIRCHNER: paul.kirchner@irisa.fr
*Rennes University, Rennes, France*

VOLUME EDITORS

Stephen D. Galbraith
Mathematics Department
University of Auckland
New Zealand
https://orcid.org/0000-0001-7114-8377

---

The cover image is based on an illustration from the article "Supersingular curves with small noninteger endomorphisms", by Jonathan Love and Dan Boneh (see p. 9).



Electronic copies can be obtained free of charge from http://msp.org/obs/4 and printed copies can be ordered from MSP (contact@msp.org).

The Open Book Series is a trademark of Mathematical Sciences Publishers.

ISSN: 2329-9061 (print), 2329-907X (electronic)

ISBN: 978-1-935107-07-1 (print), 978-1-935107-08-8 (electronic)

First published 2020.

---

**MATHEMATICAL SCIENCES PUBLISHERS**
798 Evans Hall #3840, c/o University of California, Berkeley CA 94720-3840
contact@msp.org **http://msp.org**

# THE OPEN BOOK SERIES 4

## Fourteenth Algorithmic Number Theory Symposium

The Algorithmic Number Theory Symposium (ANTS), held biennially since 1994, is the premier international forum for research in computational and algorithmic number theory. ANTS is devoted to algorithmic aspects of number theory, including elementary, algebraic, and analytic number theory, the geometry of numbers, arithmetic algebraic geometry, the theory of finite fields, and cryptography.

This volume is the proceedings of the fourteenth ANTS meeting, which took place 29 June to 4 July 2020 via video conference, the plans for holding it at the University of Auckland, New Zealand, having been disrupted by the COVID-19 pandemic. The volume contains revised and edited versions of 24 refereed papers and one invited paper presented at the conference.

TABLE OF CONTENTS