Approximate CVP\(_{\infty}\) in time \(2^{0.802n}\)

Friedrich Eisenbrand * 
EPFL 
Switzerland 
friedrich.eisenbrand@epfl.ch

Moritz Venzin 
EPFL 
Switzerland 
moritz.venzin@epfl.ch

May 12, 2020

Abstract

We show that a constant factor approximation of the shortest and closest lattice vector problem w.r.t. \(\ell_{\infty}\) can be computed in time \(2^{(0.802+\epsilon)n}\). This is breaking the kissing number barrier of \(3^n\) that is inherent in the previous best approaches tackling this problem.

We obtain this improvement by incorporating a bound on the number of scaled hypercubes that are necessary to cover the \(\ell_2\)-ball of radius \(\sqrt{n}\). The final procedure is then a modification of the list-sieve algorithm for \(\ell_2\). It is to pick the smallest pairwise difference w.r.t. \(\ell_{\infty}\) of the generated lattice vectors.

1 Introduction

The shortest vector problem (SVP) and the closest vector problem (CVP) are important algorithmic problems in the geometry of numbers. Given a rational lattice

\[ \mathcal{L}(B) = \{Bx : x \in \mathbb{Z}^n\} \]

with \(B \in \mathbb{Q}^{n \times n}\) and a target vector \(t \in \mathbb{Q}^n\) the closest vector problem asks for lattice vector \(v \in \mathcal{L}(B)\) minimizing \(\|t - v\|\). The shortest vector problem asks for a nonzero lattice vector \(v \in \mathcal{L}(B)\) of minimal norm. When using the \(\ell_p\) norms for \(1 \leq p \leq \infty\), we denote the problems by SVP\(_p\) resp. CVP\(_p\).

Much attention has been devoted to the hardness of approximating SVP and CVP. In a long sequence of papers, including [vEB81, Ajt98, Mic01, Aro95, DKRS03, Kho05, HR07] it has been shown that SVP and CVP are hard to approximate to within almost polynomial factors under reasonable complexity assumptions. The best polynomial-time approximation algorithms have exponential approximation factors [LLL82, Sch87, AKS01].

The first algorithm to solve CVP for any norm that has exponential running time in the dimension only was given by Lenstra [Len83]. The running time of his procedure is \(2^{O(n^2)}\) times a polynomial in the encoding length. In fact, Lenstra’s algorithm solves the more general integer programming problem. Kannan [Kan87] improved this to \(n^{O(n)}\) time and polynomial space. It took almost 15 years until Ajtai, Kumar and Sivakumar presented a randomized algorithm for SVP\(_2\) with time and space \(2^{O(n)}\) and a \(2^{(1+1/\epsilon)n}\) time and space algorithm for \((1+\epsilon)\)-CVP\(_2\) [AKS01, AKS02]. Here \((1+\epsilon)\)-CVP\(_2\) is the problem of finding a lattice vector, whose distance to the target is at most \(1 + \epsilon\) times the minimal distance. Blömer and Naewe [BN09] extended the randomized sieving algorithm of Ajtai et al. to solve SVP\(_p\) and obtain a \(2^{O(n)}\) time and space exact algorithm for SVP\(_p\) and an \(O(1+1/\epsilon)^{2n}\) time algorithm to compute a \((1+\epsilon)\) approximation for CVP\(_p\). For CVP\(_\infty\), one has a faster approximation algorithm.

*The author acknowledges support from the Swiss National Science Foundation (SNSF) within the project Lattice Algorithms and Integer Programming (Nr. 185030).
Eisenbrand et al. [EHN11] showed how to boost any constant approximation algorithm for \( \text{CVP}_\infty \) to a \((1+\varepsilon)\)-approximation algorithm in time \( O(\log(1+1/\varepsilon)^n) \). Recently, this idea was adapted in [NV19] to all \( \ell_p \) norms, showing that \((1+\varepsilon)\) approximate \( \text{CVP}_p \) can be solved in time \((1+1/\varepsilon)^{n/\min(2,p)} \) by boosting the deterministic CVP algorithm for general (even asymmetric) norms with a running time of \((1+1/\varepsilon)^n \) that was developed by Dadush and Kun [DK16].

The first deterministic singly-exponential time and space algorithm for exact \( \text{CVP}_2 \) (and \( \text{SVP}_2 \)) was developed by [MV10a]. The fastest exact algorithms for \( \text{CVP}_2 \) and \( \text{CVP}_p \) run in time and space \( 2^{n+o(n)} \) [ADRS15, ADS15, AS18b]. Single exponential time and space algorithms for exact \( \text{CVP} \) are only known for \( \ell_2 \). Whether \( \text{CVP} \) and the more general integer programming problem can be solved in time \( 2^{O(n)} \) is a prominent mystery in algorithms.

Recently there has been exciting progress in understanding the fine grained complexity of exact and constant approximation algorithms for \( \text{CVP} \) [ABGS19, BGS17, AS18a]. Under the assumption of a strong exponential time hypothesis (SETH) and for \( p \neq 0 \) (mod 2), exact \( \text{CVP}_p \) cannot be solved in time \( 2^{(1-\varepsilon)d} \). Here \( d \) is the ambient dimension of the lattice, which is the number of vectors in a basis of the lattice. Under the assumption of a gap-version of the strong exponential time hypothesis (gap-SETH) these lower bounds also hold for the approximate versions of \( \text{CVP}_p \). More precisely, for each \( \varepsilon > 0 \) there exists a constant \( \gamma_\varepsilon > 1 \) such that there exists no \( 2^{(1-\varepsilon)d} \) algorithm that computes a \( \gamma_\varepsilon \)-approximation of \( \text{CVP}_p \).

In the case of \( \ell_\infty \), the current best constant approximation algorithms for \( \text{CVP}_\infty \) run in time \( 3^n \) [AM18, Muk19]. This is related to the kissing number for \( \ell_\infty \) which is the maximum number of unit boxes that can be arranged in such a way that they touch another given unit box. The kissing number for \( \ell_\infty \) is \( 3^n - 1 \). Aggarwal et al. [ABGS19] raise the question whether the kissing number is a natural running time for a constant approximation algorithm for \( \text{CVP}_\infty \).

Our main result is the following theorem.

**Theorem.** For each \( \varepsilon > 0 \), there exists a constant \( \gamma_\varepsilon \) such that a \( \gamma_\varepsilon \)-approximate solution to \( \text{CVP}_\infty \), as well as to \( \text{SVP}_\infty \), can be found in time \( 2^{O(0.802+\varepsilon)n} \).

This shows in particular that the kissing number is not a lower bound for the running time of a constant factor approximation algorithm. The main idea of our approach is to establish a direct link between approximation algorithms for \( \ell_2 \) and \( \ell_\infty \) via a covering argument.

## 2 Covering balls with boxes

We now outline the main idea for an approximate \( \text{SVP}_\infty \) algorithm that runs in time \( 2^{0.802n} \). This matches the currently fastest constant approximation for their respective counterparts w.r.t. \( \ell_2 \), see [LWXZ11, PS09]. Let us assume that the shortest vector of \( \mathcal{L} \) w.r.t. \( \ell_\infty \) is \( s \in \mathcal{L} \setminus \{0\} \). We can assume that the lattice is scaled such that \( \|s\|_\infty = 1 \) holds. The euclidean norm of \( s \) is then bounded by \( \sqrt{n} \). Suppose now that there is a procedure that, for some constant \( \gamma > 1 \) independent of \( n \), generates distinct lattice vectors \( v_1,\ldots,v_n \in \mathcal{L} \) of length at most \( \|v_i\|_2 \leq \gamma \sqrt{n} \).

How large does the number of vectors \( N \) have to be such that we can guarantee that there exists two indices \( i \neq j \) with

\[
\|v_i - v_j\|_\infty \leq \alpha, \tag{1}
\]

where \( \alpha \geq 1 \) is the approximation guarantee for \( \text{SVP}_\infty \) that we want to achieve? Suppose that \( N \) is larger than the minimal number of copies of the box \((\alpha/2)b_\infty^n \) that are required to cover the ball \( \sqrt{n}b_\infty^n \). Here \( b_\infty^n = \{x \in \mathbb{R}^n : \|x\|_p \leq 1\} \) denotes the unit ball w.r.t. the \( \ell_p \)-norm. Then, by the pigeon-hole principle, two different vectors \( v_i \) and \( v_j \) must be in the same box. Their difference satisfies (1) and thus is an \( \alpha \)-approximate shortest vector w.r.t. \( \ell_\infty \), see Figure 1.
Thus we are interested in the translative covering number $N(\sqrt{n}B_2^n, aB_\infty^n)$, which is the number of translated copies of the box $aB_\infty^n$ that are needed to cover the $\ell_2$-ball of radius $\sqrt{n}$. In the setting above, $a$ is the constant $\alpha/(2\gamma)$. Covering problems like these have received considerable attention in the field of convex geometry, see [AAS15, Nas14]. These techniques rely on the classical set-cover problem and the logarithmic integrality gap of its standard LP-relaxation, see, e.g. [Vaz13, Chv79]. To keep this paper self-contained, we briefly explain how this can be applied to our setting.

If we cover the finite set $(1/n)Z^n \cap \sqrt{n}B_2^n$ with cubes whose centers are on the grid $(1/n)Z^n$, then by increasing the side-length of those cubes by an additive $1/n$, one obtains a full covering of $\sqrt{n}B_2^n$. This is a set-covering problem with ground set $U = (1/n)Z^n \cap \sqrt{n}B_2^n$ and sets

$$S_t = U \cap aB_\infty^n + t, \; t \in (1/n)Z^n,$$

ignoring empty sets. An element of the ground set is contained in exactly $|(1/n)Z^n \cap aB_\infty^n|$ many sets. Therefore, by assigning each element of the ground set the fractional value $1/|(1/n)Z^n \cap aB_\infty^n|$, one obtains a feasible fractional covering. The weight of this fractional covering is

$$\frac{T}{|(1/n)Z^n \cap aB_\infty^n|}$$

where $T$ is the number of sets. Clearly, if a cube intersects $\sqrt{n}B_2^n$, then its center is contained in the Minkowski sum $\sqrt{n}B_2^n + aB_\infty^n$ and thus the weight of the fractional covering is

$$\frac{|(\sqrt{n}B_2^n + aB_\infty^n) \cap \frac{1}{n}Z^n|}{|\frac{1}{n}Z^n \cap aB_\infty^n|} = O\left(\frac{\text{vol}(\sqrt{n}B_2^n + aB_\infty^n)}{\text{vol}(aB_\infty^n)}\right)$$

Since the size of the ground-set is bounded by $n^{O(n)}$ and since the integrality gap of the set-cover LP is at most the logarithm of this size, one obtains

$$N(\sqrt{n}B_2^n, aB_\infty^n) \leq \text{poly}(n)\frac{\text{vol}(\sqrt{n}B_2^n + aB_\infty^n)}{\text{vol}(aB_\infty^n)}\quad(2)$$

By Steiner’s formula, see [Gru07, Sch13, HRGZ97], the volume of $K + tB_2^n$ is a polynomial in $t$, with coefficients $V_j(K)$ only depending on the convex body $K$:

$$\text{vol}(K + tB_2^n) = \sum_{j=0}^{n} V_j(K) \text{vol}(B_2^{n-j}) t^{n-j}$$

For $K = aB_\infty^n$, $V_j(K) = (2a)^j \binom{n}{j}$. Setting $t = \sqrt{n}$, the resulting expression has been evaluated in [JA15, Theorem 7.1].
Theorem 2.1 ([JA15]). Denote by $H$ the binary entropy function and let $\phi \in (0, 1)$ the unique solution to
\[
\frac{1 - \phi^2}{\phi^3} = \frac{2a^2}{\pi}.
\]
Then
\[
\text{vol}(aB_2^n + \sqrt{n}B_2^n) = O(2^{n[H(\phi) + (1 - \phi)\log(2a) + \frac{\phi}{2}\log(\frac{2\pi}{\phi})]})
\]
Using this bound in inequality (2) and simplifying, we find
\[
N(\sqrt{n}B_2^n, aB_\infty^n) \leq \text{poly}(n)2^{n[H(\phi) + \phi^2\log(\frac{2\pi}{\phi})]}
\]
Both $H(\phi)$ and $\frac{\phi}{2}\log(\frac{2\pi}{\phi})$ decrease to 0 as $\phi$ decreases to 0. Since $\phi$, the unique solution to (3), satisfies $\phi \leq \frac{\sqrt{15\pi}}{3n}$, we obtain the following bound.

Lemma 2.2. For each $\varepsilon > 0$, there exists $a_\varepsilon \in \mathbb{R}_{>0}$ independent of $n$, such that
\[
N(\sqrt{n}B_2^n, a_\varepsilon B_\infty^n) \leq 2^{\varepsilon n}.
\]

3 Approximate SVP$_\infty$ and CVP$_\infty$

We now describe our main contribution. As we mentioned already, SVP$_2$ can be approximated up to a constant factor in time $2^{O(0.802 + \varepsilon)n}$ for each $\varepsilon > 0$. This follows from a careful analysis of the list sieve algorithm of Micciancio and Voulgaris [MV10b], see [LWXZ11, PS09]. The running time and space of this algorithm is directly related to the kissing number of the $l_2$-norm. The running time is the square of the best known upper bound by Kabatiansky and Levenshtein [KL78]. The kissing number of the $l_\infty$-norm is $3^n - 1$. In light of this, Aggarwal et al. [ABGS19] raised the question whether $3^n$ is a natural bound on the running time of CVP$_\infty$.

The main insight of our paper is that the current list-sieve variants can be used to approximate SVP$_\infty$ and CVP$_\infty$ by testing all pairwise differences of the generated lattice vectors. We note that straightforward adaption of the algorithm then also gives an approximation for $l_p$ ($p \geq 2$) with same running time.

3.1 List sieve

We begin by describing the list-sieve method [MV10b] to a level of detail that is necessary to understand our main result. Our exposition follows closely the one given in [PS09]. Let $\mathcal{L}(B)$ be a given lattice and $s \in \mathcal{L}$ be an unknown lattice vector. This unknown lattice vector $s$ is typically the shortest, respectively closest vector in $\mathcal{L}(B)$.

The list-sieve algorithm has two stages. The input to the first stage of the algorithm is an LLL-reduced lattice basis $B$ of $\mathcal{L}(B)$, a constant $\varepsilon > 0$ and a guess $\mu$ on the length of $s$ that satisfies
\[
\|s\|_2 \leq \mu \leq (1 + 1/n)\|s\|_2.
\]
The first stage then constructs a list of lattice vectors $L \subseteq \mathcal{L}(B)$ that is random. This list of lattice vectors is then passed on to the second stage of the algorithm.

The second stage of the algorithm proceeds by sampling points $y_1, \ldots, y_N$ uniformly and independently at random from the ball
\[
(\xi \cdot \mu)B_2^n.
\]
where \( \xi_\varepsilon \) is an explicit constant depending on \( \varepsilon \) only. It then transforms these points via a deterministic algorithm \( \text{ListRed}_L \) into lattice points

\[
\text{ListRed}_L(y_1), \ldots, \text{ListRed}_L(y_N) \in \mathcal{L}(B).
\]

The deterministic algorithm \( \text{ListRed}_L \) uses the list \( L \subseteq \mathcal{L}(B) \) from the first stage.

As we mentioned above, the list \( L \subseteq \mathcal{L}(B) \) that is used by the deterministic algorithm \( \text{ListRed}_L \) is random. We will show the following theorem in the next section. The novelty compared to the literature is the reasoning about pairwise differences lying in centrally symmetric sets. In this theorem, \( \varepsilon > 0 \) is an arbitrary constant, \( \xi_\varepsilon \) as well as \( c_\varepsilon \) are explicit constants and \( K \) is some centrally symmetric set. Furthermore, we assume that \( \mu \) satisfies (4).

The theorem reasons about an area \( I_s \) that is often referred as the \textit{lens}, see Figure 2. The lens was introduced by Regev as a conceptual modification to facilitate the proof of the original AKS algorithm [Reg04].

\[
I_s = (\xi_\varepsilon \cdot \mu B^n_2) \cap \left( -s + (\xi_\varepsilon \cdot \mu B^n_2) \right)
\]

\( \text{(5)} \)

**Theorem 3.1.** With probability at least 1/2, the list \( L \) that was generated in the first stage satisfies the following. If \( y_1, \ldots, y_N \) are chosen independently and uniformly at random within \( B^n(0, \xi_\varepsilon \mu) \) then

i) The probability of the event that two different samples \( y_i, y_j \) satisfy

\[
y_i, y_j \in I_s \text{ and } \text{ListRed}_L(y_i) - \text{ListRed}_L(y_j) \in K
\]

is at most twice the probability of the event that two different samples \( y_i, y_j \) satisfy

\[
\text{ListRed}_L(y_i) - \text{ListRed}_L(y_j) \in K + s
\]

ii) For each sample \( y_i \) the probability of the event

\[
\|\text{ListRed}_L(y_i)\|_2 \leq c_\varepsilon \|s\|_2 \text{ and } y_i \in I_s
\]

is at least \( 2^{-\varepsilon n} \).

The complete procedure, i.e. the construction of the list \( L \) in stage one and applying \( \text{ListRed}_L \) to the \( N \) samples \( y_1, \ldots, y_N \) in stage two takes time \( N(2^{0.401+\varepsilon}n + 2^{0.802+\varepsilon}n) \) and space \( N + 2^{(0.401+\varepsilon)n} \).

The proof of Theorem 3.1 follows verbatim from Pujol and Stehlé [PS09], see also [LWXZ11]. In [PS09], \( s \) is a shortest vector w.r.t. \( \ell_2 \). But this fact is never used in the proof and in the analysis. Part ii) follows from Lemma 5 and Lemma 6 in [PS09]. Their probability of a sample being in the lens \( I_s \subseteq \xi \|s\|_2 B^n_2 \) depends only on \( \varepsilon \) (corresponding to our \( \xi_\varepsilon \)). By choosing \( \xi \) large enough, this
happens with probability at least $2^{-cn}$. Their Lemma 6 then guarantees that the list $L$, with probability 1/2, when $y_i \sim I_s$ is sampled uniformly, returns a lattice vector of length at most $r_0 \|s\|_2$ ($r_0$ corresponds to our $c_2$). This corresponds to part ii) in our setting. The size of their list (denoted by $N_T$) is bounded above by $2^{0.401+\delta)n}$ where $\delta > 0$ decreases to 0 as the ratio $r_0/\xi$ increases, this is their Lemma 4.

Finally, part i) also follows from Pujol and Stelhe [PS09]. It is in their proof of correctness, Lemma 7, involving the lens $I_s$. We briefly comment on our general viewpoint. Given $y \sim (\xi, \mu)B_2^n$, the algorithm computes the linear combination w.r.t. to the lattice basis $b_1, \ldots, b_n$

$$y = \sum_{i=1}^{n} \lambda_i b_i$$

and then the remainder

$$y \pmod{\Lambda} = \sum_{i=1}^{n} |\lambda_i| b_i.$$ 

The important observation is that this remainder is the same for all vectors $y + v$, $v \in \Lambda$. Next, it keeps reducing the remainder w.r.t. the list, as long as the length decreases. This results in a vector of the form

$$y \pmod{\Lambda} = v_1 - \cdots - v_k, \text{ for some } v_i \in L.$$ 

The output $\text{ListRed}_L(y)$ is then

$$y \pmod{\Lambda} = v_1 - \cdots - v_k + y \in \Lambda.$$ 

The algorithm bases its decisions on $y \pmod{\Lambda}$ and not on $y$ directly. This is why one can imagine that, after $y \pmod{\Lambda}$ has been created, one applies a bijection $\tau$ of the ball $\tau(y) = \xi \mu B_2^n$ on $y$ with probability 1/2. For $y \in I_s$ one has $\tau(y) = y + s$. We refer to [PS09] for the definition of $\tau$. Since $\tau$ is a bijection, the result of applying $\tau(y)$ with probability 1/2 is distributed uniformly. This means that for $y \in I_s$ this modified but equivalent procedure outputs $\text{ListRed}_L(y)$ or $\text{ListRed}_L(y) + s$, both with probability 1/2. If $\text{ListRed}_L(y_i) - \text{ListRed}_L(y_j) \in K$, we toss a for $i$ and $j$ each. With probability 1/2, their difference is in $\pm K + s$.

With Theorem 3.1 at hand we can now prove our main result.

**Theorem 3.2.** There is a randomized algorithm that computes with constant probability a constant factor approximation to SVP$_\infty$ and CVP$_\infty$ respectively. The algorithm runs in time $2^{0.802+\epsilon}n$ and it requires space $2^{2(0.401+\epsilon)n}$.

In short, the algorithm is the standard list-sieve algorithm with a slight twist: Check all pairwise differences.

**Proof.** We assume that the list $L$ that was computed in the fist stage satisfies the properties described in Theorem 3.1. Recall that this is the case with probability at least 1/2. We first consider SVP$_\infty$. Choose $a > 0$ such that $N(\sqrt{n}B_2^n, aB_\infty^n) \leq 2^{0.401n}$ and let $s$ be a shortest vector w.r.t. $\ell_\infty$. Furthermore let $\mu > 0$ such that $\|s\|_2 \leq \mu < (1 + \frac{1}{2}) \|s\|_2$ as above. Since $\|s\|_2 \leq \sqrt{n} \|s\|_\infty$, we have $N(c_2 \|s\|_2 B_2^n, c_2 a \|s\|_\infty B_\infty^n) \leq 2^{0.401n}$. This means that, if $2^{0.401n} + 1$ lattice vectors are contained in the ball $c_2 \|s\|_2 B_2^n$ at least two of them have $\ell_\infty$-distance bounded by $2c_2 a$ which is a constant.

Set $N = 2 \cdot [2^{0.401n} + 1]$ and $(y_1, \ldots, y_N) \sim \text{i.i.d. } B_2^n(0, \xi, \mu)$ uniformly and independently at random. By Theorem 3.1 ii) and by the Chebychev inequality, see [PS09], the following event has a probability at least 1/2.

(6) $(E_A)$: There is a subset $S \subseteq \{1, \ldots, N\}$ with $S = [2^{0.401n}] + 1$ such that for each $i \in S$

$$y_i \in I_s \text{ and } \|\text{ListRed}_L(y_i)\|_2 \leq c_2 \|s\|_2.$$
This event is the disjoint union of the event $A \cap B$ and $A \cap \overline{B}$, where $B$ denotes the event where the vectors $\text{ListRed}_L(y_i)$, $y_i \in I_s$ are all distinct. Thus
\[
\Pr(A) = \Pr(A \cap B) + \Pr(A \cap \overline{B}).
\]
The probability of at least one of the events $A \cap B$ and $A \cap \overline{B}$ is bounded below by $1/4$. In the event $A \cap B$, there exists $i \neq j$ such that
\[
\|\text{ListRed}_L(y_i) - \text{ListRed}_L(y_j)\|_\infty \leq 2c_e a.
\]
By Theorem 3.1 i) with $K = \{0\}$ one has
\[
\Pr(A \cap \overline{B}) \leq 2\Pr(\exists i \neq j: \text{ListRed}_L(y_i) - \text{ListRed}_L(y_j) = s).
\]
Therefore, with constant probability, there exist $i, j \in \{1, \ldots, N\}$ with
\[
0 < \|\text{ListRed}_L(y_i) - \text{ListRed}_L(y_j)\|_\infty \leq 2c_e a.
\]
We try out all the pairs of $N$ elements, which amounts to $N^2 = 2^{(0.802 + \epsilon)n}$ additional time.
We next describe how list-sieve yields a constant approximation for CVP$_\infty$. Let $w \in \mathcal{L}(B)$ be the closest lattice vector w.r.t. $\ell_\infty$ to $t \in \mathbb{R}^n$ and let $\mu > 0$ such that $\|t - w\|_2 \leq \mu < (1 + \frac{1}{n}) \|t - w\|_2$. We use Kannan's embedding technique [Kan87] and define a new lattice $\mathcal{L}'$ with basis
\[
\tilde{B} = \begin{pmatrix} B & t \\ 0 & \frac{1}{n} \mu \end{pmatrix} \in Q^{(n+1) \times (n+1)},
\]
Finding the closest vector to $t$ w.r.t. $\ell_\infty$ in $\mathcal{L}(B)$ amounts to finding the shortest vector w.r.t. $\ell_\infty$ in $\mathcal{L}'(\tilde{B}) \cap \{x \in \mathbb{R}^{n+1}: x_{n+1} = 1/\mu\}$. The vector $s = (t - w, \frac{1}{n})$ is such a vector and its euclidean length is smaller than $(1 + \frac{1}{n}) \mu$. Let $a > 0$ be such that
\[
N(\sqrt{n} \mathbb{B}_{\mathbb{Z}}^n, a \mathbb{B}_{\infty}^n) \leq 2^{0.401n}.
\]
This means that there is a covering of the $n$-dimensional ball $(c_\epsilon \|s\|_2) \mathbb{B}_{\infty}^{n+1} \cap \{x \in \mathbb{R}^{n+1}: x_{n+1} = 0\}$ by $2^{0.401n}$ translated copies of $K$, where
\[
K = (c_\epsilon \cdot a(1 + 1/n)\|s\|_\infty) \mathbb{B}_{\infty}^{n+1} \cap \{x \in \mathbb{R}^{n+1}: x_{n+1} = 0\}.
\]
(The factor $(1 + 1/n)$ is a reminiscent of the embedding trick, $s$ is $n + 1$ dimensional.) Similarly, we may cover $(c_\epsilon \|s\|_2) \mathbb{B}_{\infty}^{n+1} \cap \{x \in \mathbb{R}^{n+1}: x_{n+1} = k \cdot \frac{1}{n}\}$ for all $k \in \mathbb{Z}$ (such that the intersection is not empty) by translates of $K$. There are only $2c_\epsilon(n + 1) + 1$ such layers to consider and so $(2c_\epsilon(n + 1) + 1)2^{0.401n}$ translates of $K$ suffice. The last component of a lattice vector of $\mathcal{L}'$ is of the form $k \cdot \frac{1}{n}$ and it follows that these translates of $K$ cover all lattice vectors of euclidean norm smaller than $c_\epsilon \|s\|_2$, see Figure 3. Set $N = \lceil (2c_\epsilon(n + 1) + 2)2^{(\epsilon + 0.401)n} \rceil$ and sample again $\{y_1, \ldots, y_N\} \sim \mathbb{B}_{\mathbb{Z}}^n(0, \ell_\epsilon \mu)$ uniformly and independently at random. By Theorem 3.1 ii) and by the Chebychev inequality, see [PS09], the following event has a probability at least $1/2$.

(5) EVENT A': There is a subset $S \subseteq \{1, \ldots, N\}$ with $S = (2c_\epsilon(n + 1) + 1)2^{0.401n} + 1$ such that for each $i \in S$
\[
y_i \in I_s \text{ and } \|\text{ListRed}_L(y_i)\|_2 \leq c_\epsilon \|s\|_2.
\]
In this case, there exists a translate of $K$ that holds at least two vectors $\text{ListRed}_L(y_i)$ and $\text{ListRed}_L(y_j)$ for different samples $y_i$ and $y_j$, see Figure 3 with $v_i, v_j \in \mathcal{L}'$ instead. Thus, with probability at least $1/2$, there are $i, j \in \{N\}$ with $y_i, y_j \in I_s$ such that
\[
\text{ListRed}_L(y_i) - \text{ListRed}_L(y_j) \in 2K.
\]
Figure 3: Covering the lattice points with translates of $K$

Theorem 3.1 i) implies that, with probability at least $1/4$, there exist different samples $y_i$ and $y_j$ such that

$$\text{ListRed}_L(y_i) - \text{ListRed}_L(y_j) \in 2K + s$$

In this case, the first $n$ coordinates of $\text{ListRed}_L(y_i) - \text{ListRed}_L(y_j)$ can be written of the form $t - v$ for $v \in L$ and the first $n$ coordinates on the right hand side are of the form $(t - w) + z$, where $z \in L'$ and $\|z\|_\infty \leq 2c_\epsilon(1 + 1/n)\|s\|_\infty = 2c_\epsilon(1 + 1/n)\|t - w\|_\infty$. In particular, the lattice vector $v \in L$ is a $2ac_\epsilon$ approximation to the closest vector to $t$. Since we need to try out all pairs of the $N$ elements, this takes time $N^2 = 2^{(0.802 + \epsilon)n}$ and space $N$.

Remark 3.3.

i) For clarity we have not optimized the approximation factor. There are various ways to do so. We remark that for SVP$_\infty$ we actually get a smaller approximation factor than the one that we describe. Let $\tilde{a}$ be such that $N(\sqrt{n}B_2^n, \tilde{a}B_\infty^n) \leq 2^{0.802n}$, the algorithm described above yields a $2c_\epsilon\tilde{a}$ approximation instead of a $2c_\epsilon a$ approximation to the shortest vector. This follows by applying the birthday paradox in the way that it was used by Pujol and Stehlé [PS09]. The same argument also applies to CVP$_\infty$.

ii) For $p \geq 2$, SVP$_p$ and CVP$_p$ can also be approximated to within a constant factor in time $2^{(0.802 + \epsilon)n}$ and space $2^{(0.401 + \epsilon)n}$: We define $s$ to be shortest (resp. closest) vector w.r.t. $\ell_p$, by Hölder's inequality we have $\|s\|_2 \leq n^{1/2 - 1/p} \|s\|_2$. The rest follows immediately from our description of the algorithm described above and the following analogue of Lemma 2.2. It directly follows from Lemma 2.2 since $n^{-1/p}B_2^n \subseteq B_p^n$.

Let $p \geq 2$. For each $\epsilon > 0$, there exists $a_\epsilon \in \mathbb{R}_{>0}$ independent of $n$, such that

$$N(n^{1/2 - 1/p}B_2^n, a_\epsilon B_p^n) \leq 2^{\epsilon n}.$$ 

References

[AAS15] Shiri Artstein-Avidan and Boaz A Slomka, On weighted covering numbers and the levi-hadwiger conjecture, Israel Journal of Mathematics 209 (2015), no. 1, 125–155.

[ABGS19] Divesh Aggarwal, Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz, Fine-grained hardness of cvp (p)—everything that we can prove (and nothing else), arXiv preprint arXiv:1911.02440 (2019).
[Kan87] Ravi Kannan, *Minkowski’s convex body theorem and integer programming*, Math. Oper. Res. **12** (1987), no. 3, 415–440.

[Kho05] Subhash Khot, *Hardness of approximating the shortest vector problem in lattices*, J. ACM **52** (2005), no. 5, 789–808.

[KL78] Grigorii Anatol’evich Kabatiansky and Vladimir Iosifovich Levenshtein, *On bounds for packings on a sphere and in space*, Problemy Peredachi Informatsii **14** (1978), no. 1, 3–25.

[Len83] Hendrik W. Lenstra, *Integer programming with a fixed number of variables*, Math. Oper. Res. **8** (1983), no. 4, 538–548.

[LLL82] A. K. Lenstra, H. W. Lenstra, and L. Lovász, *Factoring polynomials with rational coefficients*, Mathematische Annalen **261** (1982), no. 4, 515–534.

[LWXZ11] Mingjie Liu, Xiaoyun Wang, Guangwu Xu, and Xuexin Zheng, *Shortest lattice vectors in the presence of gaps*, IACR Cryptology ePrint Archive 2011 (2011), 139.

[Mic01] Daniele Micciancio, *The shortest vector in a lattice is hard to approximate to within some constant*, SIAM journal on Computing **30** (2001), no. 6, 2008–2035.

[Muk19] Priyanka Mukhopadhyay, *Faster provable sieving algorithms for the shortest vector problem and the closest vector problem on lattices in $\ell_p$ norm*, CoRR abs/1907.04406 (2019).

[MV10a] Daniele Micciancio and Panagiotis Voulgaris, *A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations*, Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010, 2010, pp. 351–358.

[MV10b] Daniele Micciancio and Panagiotis Voulgaris, *Faster exponential time algorithms for the shortest vector problem*, Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (USA), SODA ’10, Society for Industrial and Applied Mathematics, 2010, p. 1468–1480.

[Nas14] Márton Naszódi, *On some covering problems in geometry*, Proceedings of the American Mathematical Society **144** (2014).

[NV19] Márton Naszódi and Moritz Venzin, *Covering convex bodies and the closest vector problem*, arXiv preprint arXiv:1908.08384 (2019).

[PS09] Xavier Pujol and Damien Stehlé, *Solving the shortest lattice vector problem in time $2^{2.465n}$*, IACR Cryptology ePrint Archive **2009** (2009), 605.

[Reg04] Oded Regev, *Lattices in computer science, lecture 8: $2^{O(n)}$ algorithm for svp*, 2004.

[Sch77] Claus-Peter Schnorr, *A hierarchy of polynomial time lattice basis reduction algorithms*, Theoretical computer science **53** (1987), no. 2-3, 201–224.

[Sch13] Rolf Schneider, *Convex bodies: The brunn–minkowski theory*, 2 ed., Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2013.

[Vaz13] Vijay V Vazirani, *Approximation algorithms*, Springer Science & Business Media, 2013.

[vEB81] P. van Emde Boas, *Another NP-complete problem and the complexity of computing short vectors in a lattice*, Technical Report 81-04, Mathematische Instituut, University of Amsterdam (1981).