# Covering Cubes and the Closest Vector Problem

Friedrich Eisenbrand
École Polytechnique Fédérale de Lausanne
Station 8
1015 Lausanne, Switzerland
friedrich.eisenbrand@epfl.ch

Nicolai Hähnle
École Polytechnique Fédérale de Lausanne
Station 8
1015 Lausanne, Switzerland
nicolai.haehnle@epfl.ch

Martin Niemeier
École Polytechnique Fédérale de Lausanne
Station 8
1015 Lausanne, Switzerland
martin.niemeier@epfl.ch

## ABSTRACT

We provide the currently fastest randomized $(1+\varepsilon)$-approximation algorithm for the *closest lattice vector problem* in the $\ell_\infty$-norm. The running time of our method depends on the dimension $n$ and the approximation guarantee $\varepsilon$ by

$$2^{O(n)}(\log 1/\varepsilon)^{O(n)}$$

which improves upon the $(2+1/\varepsilon)^{O(n)}$ running time of the previously best algorithm by Blömer and Naewe.

Our algorithm is based on a solution of the following geometric covering problem that is of interest of its own: Given $\varepsilon \in (0,1)$, how many ellipsoids are necessary to cover the cube $[-1+\varepsilon, 1-\varepsilon]^n$ such that all ellipsoids are contained in the standard unit cube $[-1,1]^n$? We provide an almost optimal bound for the case where the ellipsoids are restricted to be axis-parallel.

We then apply our covering scheme to a variation of this covering problem where one wants to cover $[-1+\varepsilon, 1-\varepsilon]^n$ with parallelepipeds that, if scaled by two, are still contained in the unit cube. Thereby, we obtain a method to boost any 2-approximation algorithm for closest-vector in the $\ell_\infty$-norm to a $(1+\varepsilon)$-approximation algorithm that has the desired running time.

## Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Computations on discrete structures, Geometrical problems and computations*

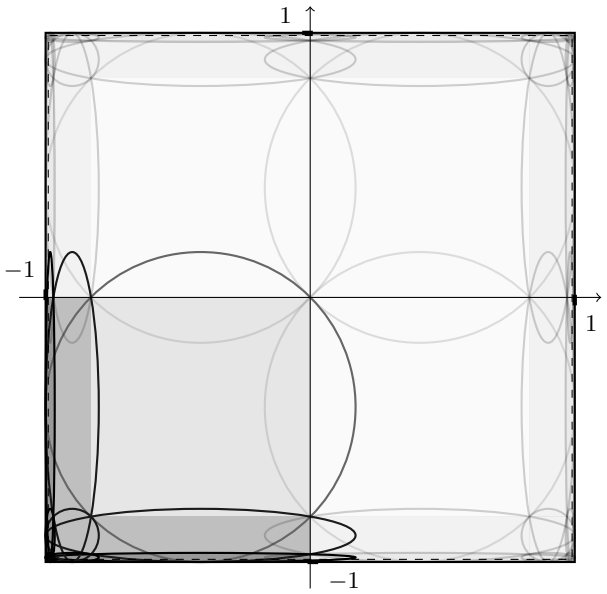
## General Terms

Algorithms, Theory

## Keywords

closest lattice vector problem, approximation algorithm, ellipsoid cover, box cover, covering convex bodies


*SCG'11,* June 13–15, 2011, Paris, France.


## 1. INTRODUCTION

The *closest lattice vector problem* (CVP) is one of the central computational problems in the *geometry of numbers*. Here, one is given a rational *lattice* $\Lambda(A) = \{Ax : x \in \mathbb{Z}^n\}$, $A \in \mathbb{Q}^{n\times n}$, and a *target vector* $t \in \mathbb{Q}^n$. The task is to compute a lattice-point in $\Lambda(A)$ that is closest to $t$ w.r.t. a given norm. In this paper, we focus on the closest vector problem in the $\ell_\infty$-norm. CVP in the $\ell_\infty$-norm is an *integer programming problem*: The points of distance at most $\beta$ from the target vector $t$ are the integer points $x \in \mathbb{Z}^n$ that are contained in the polytope $\{x \in \mathbb{R}^n : -\beta\cdot\mathbf{1} \le Ax - t \le \beta\mathbf{1}\}$, and thus an integer solution minimizing $\beta$ is a solution for $\mathrm{CVP}_\infty$. On the other hand, any integer programming problem as above can be directly reduced to $\mathrm{CVP}_\infty$ in a lattice in $m$-dimensional space, where $m$ is the number of inequalities describing the polytope.[1] Integer programming

[1]To decide if a polytope $P = \{x \in \mathbb{R}^n : Ax \le u\}$, contains an integer point, compute a vector $l < u$ such that $P = \{x \in \mathbb{R}^n : l \le Ax \le u\}$. By rescaling each row we can wlog assume that $u - l = \mathbf{1}$. Now define $t := \frac{l+u}{2}$ and observe that $P$ contains an integer point iff there is a $v \in \Lambda(A)$ with $\|v - t\|_\infty \le \frac{1}{2}$. This lattice is not necessarily of full rank, but the techniques of this paper – whose running time, like those of previous algorithms, depends on the ambient dimension – can be applied.

is one of the most versatile modeling paradigms with a wide range of applications. Thus the closest vector problem in the $\ell_\infty$-norm variant is particularly important.

The development of methods to solve closest-vector and integer programming problems resulted in many deep discoveries in geometry and algorithms. Lenstra [Len83] showed that integer programming and thus $\mathrm{CVP}_\infty$ can be solved in polynomial time if the dimension is fixed. His algorithm lay the first planks between the geometry of numbers and optimization. For varying $n$, the running time of his method is $2^{O(n^3)}$ times a polynomial in the binary encoding length of the input. Kannan [Kan87] presented algorithms for these problems whose running-time dependence on $n$ is bounded by $2^{O(n \log n)}$. An important step forward in the quest for a singly-exponential time algorithm was provided by Ajtai et al. [AKS01]. They presented a $2^{O(n)}$ randomized algorithm for the *shortest vector problem* in the $\ell_2$-norm: Given a lattice, find the shortest nonzero lattice vector. These results have been generalized for any $\ell_p$-norm by Blömer and Naewe [BN09]. Micciancio and Voulgaris [MV10] provided a deterministic singly-exponential time algorithm both for the shortest vector problem as well as for the closest vector problem in the $\ell_2$-norm. Recently Dadush et al. [DPV10] have shown that the shortest vector problem w.r.t. any norm can be solved with a deterministic singly-exponential time algorithm.

## Approximation algorithms

A $(1+\varepsilon)$-*approximation algorithm* for the closest vector problem computes a lattice vector whose distance to the target vector $t$ is at most $(1+\varepsilon)$ times the minimum distance $\min\{\|v-t\| \colon v \in \Lambda(A)\}$. The closest vector problem is NP-hard for any $\ell_p$ norm [vEB81] and NP-hard to approximate within constant factors [Aro94] and even almost polynomial factors [DKRS03]. So clearly one cannot expect to have a *polynomial-time* approximation scheme (PTAS) for closest vector. An interesting problem is however to design exponential-time approximation algorithms whose running-time dependence on the approximation guarantee is not too large. Ajtai et al. [AKS02] provided a $(1+\varepsilon)$-approximation algorithm for $\mathrm{CVP}_2$ with a running time of $2^{O(1+1/\varepsilon)n}$. Blömer and Naewe [BN09] could improve on this and provide a randomized $(1+\varepsilon)$-approximation algorithm for the closest vector problem w.r.t. *any* $\ell_p$ norm that has a running time of $(2+1/\varepsilon)^{O(n)}$.

Our *main result* is a randomized $(1+\varepsilon)$-approximation algorithm for $\mathrm{CVP}_\infty$ whose running time depends on $n$ and $\varepsilon$ by $2^{O(n)}(\log 1/\varepsilon)^{O(n)}$. In fact, we show that any singly-exponential time constant factor approximation algorithm can be strengthened to a $(1+\varepsilon)$-approximation algorithm that, in the end, has this running time. Using the randomized algorithm of Blömer and Naewe [BN09] to obtain 2-approximate solutions, we obtain the desired running time.

## The covering technique

We now explain how coverings of the cube by convex bodies come into play to obtain the complexity result. Suppose that we have an (exact) algorithm for closest vector in the $\ell_2$-norm and we want to apply this to (approximately) decide whether the translated $\ell_\infty$-unit ball

$$B = \{x \in \mathbb{R}^n \colon \|x-t\|_\infty \le 1\}$$

contains a lattice point in $\Lambda(A)$. More precisely, given an $\varepsilon > 0$, we either want

i) to find a lattice point in $B$,

ii) or to assert that the scaled unit ball

$$B' = \{x \in \mathbb{R}^n \colon \|x-t\|_\infty \le 1-\varepsilon\}$$

does not contain a lattice point.

One obvious idea is to determine a set of balls of radius $\varepsilon$ whose centers lie in $B'$ and whose union covers $B'$. If we then use the closest-vector algorithm for the $\ell_2$-norm and target-vectors being the centers of the balls, we can solve the above problem. If one of the calls to a closest vector oracle returns a lattice point of distance at most $\varepsilon$, then we are in case i). Otherwise we are in case ii).

This relates to a classical covering problem. Erdős and Rogers [ER62] (see also [FK08]) showed that the space $\mathbb{R}^n$ can be covered by translates of unit spheres in such a way that no point is covered by more than $O(n \log n)$ spheres. One can use this to cover $[-1+\varepsilon, 1-\varepsilon]^n$ with spheres of radius $\varepsilon$ that then will be contained in $[-1,1]^n$. The Erdős and Rogers technique would yield an upper bound of $O(n \log n)\frac{(2-2\varepsilon)^n}{(\varepsilon/2)^n V_n}$ where $V_n$ is the volume of the $\ell_2$-unit ball. This yields the bound $(n/\varepsilon)^{O(n)}$ for the number of queries to the $\mathrm{CVP}_2$-oracle. Certainly, since the ratio of the volume of the unit cube $[-1,1]$ to the volume of the $\ell_2$-unit ball $\{x \in \mathbb{R}^n \colon \|x\|_2 \le 1\}$ is $2^{\Theta(n \log n)}$, we cannot hope to improve the dependency on the dimension. But can we improve the dependence on $\varepsilon$?

Since an *ellipsoid* is the image of the $\ell_2$-unit-ball $\{x \in \mathbb{R}^n \colon \|x\|_2 \le 1\}$ under an *affine transformation* $f(x) = Ex + d$ for some non-singular matrix $E \in \mathbb{R}^{n \times n}$ and a vector $d \in \mathbb{R}^n$, the problem whether such an ellipsoid contains a lattice vector is the closest vector problem w.r.t. the $\ell_2$-norm in the lattice $\Lambda(E^{-1}A)$ and target vector $E^{-1}d$. Thus, we can apply the algorithm for $\mathrm{CVP}_2$ to decide whether an ellipsoid contains a lattice point or not. This gives us more flexibility for the reduction of approximate $\mathrm{CVP}_\infty$ to $\mathrm{CVP}_2$. Consequently, if we cover $B'$ with ellipsoids that are contained in $B$ we can solve the approximate decision problem via calls to a $\mathrm{CVP}_2$-oracle. This motivates the following covering problem.

> How many ellipsoids that are contained in $[-1,1]^n$ are needed to cover $[-1+\varepsilon, 1-\varepsilon]^n$?

As we mentioned above, the volume of the cube versus the volume of an inscribed ball shows that covering with ellipsoids cannot yield a singly-exponential dependence of the running time on the dimension $n$. However, a similar idea and technique is the basis of our promised complexity result. The image of the unit-cube $[-1,1]^n$ under an affine transformation $f(x) = Ex + d$ is a *parallelepiped*.

With a 2-approximation algorithm for $\mathrm{CVP}_\infty$ one can, for a given parallelepiped $P$ find a lattice point in $P_s$, where $P_s$ stems from $P$ via scaling by 2 from its center of gravity $d$, or assert that $P$ does not contain a lattice point. More precisely this can be done by a call to a 2-approximation algorithm on the lattice $\Lambda(E^{-1}A)$ and target-vector $E^{-1}d$. This motivates the following variant of the above described covering problem.

How many parallelepipeds that, if scaled by 2 from their centers of gravity are contained in the unit cube $[-1,1]^n$, are necessary to cover the cube $[-1+\varepsilon, 1-\varepsilon]^n$?

We consider the two covering problems from above and provide the following results.

- We show that the number of ellipsoids required for the covering is bounded by $2^{O(n \log n)}(1+\log 1/\varepsilon)^n$ from above and provide a $c_n(1+\lfloor \log 1/\varepsilon \rfloor)^{n-1}$ lower bound for axis-parallel ellipsoids.
- We show that the number of required parallelepipeds is bounded from above by $2^n(1+\log 1/\varepsilon)^n$ and from below by $c'_n(1+\lfloor \log 1/\varepsilon \rfloor)^n$.

The second result yields a $2^{O(n)}(\log 1/\varepsilon)^{O(n)}$ randomized algorithm that solves the approximate decision version of closest vector in the $\ell_\infty$-norm. The lower bound shows that this complexity is optimal for an algorithm relying on this covering technique alone. Our main result, the $2^{O(n)}(\log 1/\varepsilon)^{O(n)}$ time $(1+\varepsilon)$-approximation algorithm, is then obtained via a binary-search technique. We explain this in the final section of our paper.

## 2. THE COVERING PROBLEMS

We now consider the two covering problems from the introduction. We denote the cube $[-1,1]^n$ by $H$ and its scaled version $[-1+\varepsilon, 1-\varepsilon]^n$ by $H_\varepsilon$. The questions are again as follows. Given an $\varepsilon \in (0,1)$, what is the smallest number $E(n,\varepsilon)$ of ellipsoids contained in $H$ such that their union covers the smaller cube $H_\varepsilon$? What is the smallest number $P(n,\varepsilon)$ of parallelepipeds that are contained in $H$ after being scaled by 2 and whose union covers $H_\varepsilon$?

### 2.1 Covering with ellipsoids

We first show that $E(n,\varepsilon)$ is bounded by $2^{cn\log n}(1+\log 1/\varepsilon)^n$. Since we can allow us a factor of $2^n$, we cover each intersection of $H_\varepsilon$ with an orthant separately and then combine the different coverings, see also the figure on the title-page. After flipping coordinates and after translation, the problem for one orthant can be interpreted as follows. How many ellipsoids that are contained in $H' := [0,2]^n$ are needed to cover the cube $[\varepsilon,1]^n$?

The following elementary lemma (see also Figure 1) is used in our construction.

LEMMA 2.1. *Let $n \geq 2$, $r = 1 + 2/(\sqrt{n}-1)$ and $Q := [1/r, 1]^n$, then the smallest ball containing $Q$ is contained in $H'$. Furthermore, $r$ is maximal with this property.*

PROOF. Let $B$ be the smallest ball containing $Q$. The center of $Q$ and $B$ is $d \cdot \mathbf{1}$ with

$$d = \frac{1+\frac{1}{r}}{2} = \frac{1+\frac{\sqrt{n}-1}{\sqrt{n}+1}}{2} = \frac{\sqrt{n}}{\sqrt{n}+1}.$$

Thus the radius $R$ of $B$ is simply the distance of $d \cdot \mathbf{1}$ to the vertices of $Q$

$$R = \sqrt{n}(1-d) = \sqrt{n} \cdot \left(1 - \frac{\sqrt{n}}{\sqrt{n}+1}\right) = \frac{\sqrt{n}}{\sqrt{n}+1} = d.$$

Thus the ball is contained in the positive orthant. Furthermore, $d + R < 2$, which shows the first claim, i.e. that $B \subseteq H'$. The choice of $r$ is maximal because the ball touches the coordinate hyperplanes. □

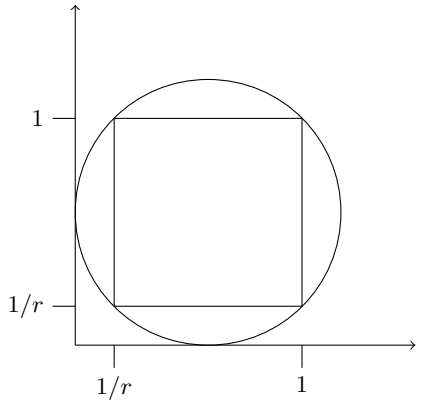



**Figure 1: Illustration of Lemma 2.1.**

COROLLARY 2.2. *Let $n \geq 2$, $r = 1 + 2/(\sqrt{n}-1)$, $v \in (0,1]^n$ and let $Q := [v_1 r^{-1}, v_1] \times \ldots \times [v_n r^{-1}, v_n]$. Then there exists an axis-parallel ellipsoid $E$ such that $Q \subseteq E \subseteq H'$.*

This corollary is obtained from Lemma 2.1 by scaling. We are now ready to prove the upper bound.

THEOREM 2.3. *One has $E(n,\varepsilon) \leq 2^{cn\log n} \cdot (1+\log 1/\varepsilon)^n$ for a fixed constant $c > 0$.*

PROOF. We provide a covering of $[\varepsilon,1]^n$ by ellipsoids contained in $H' = [0,2]^n$. Let $r = 1 + 2/(\sqrt{n}-1)$ as in Corollary 2.2. For every $\alpha \in \mathbb{N}_0^n$, the smallest ellipsoid containing a box of the form

$$Q(\alpha) = \left[r^{-(\alpha_1+1)}, r^{-\alpha_1}\right] \times \cdots \times \left[r^{-(\alpha_n+1)}, r^{\alpha_n}\right],$$

is contained in $H'$. How many of these boxes are needed to cover the cube $[\varepsilon,1]^n$?

It is enough to consider those boxes $Q(\alpha)$ with $r^{-\alpha_j} > \varepsilon$ for all $j$. Taking logarithms, one obtains $\alpha_j \log r < \log 1/\varepsilon$. A standard approximation for the logarithm yields $\log r > c'/\sqrt{n}$ for some constant $c' > 0$, and so we can conclude $\alpha_j < \sqrt{n}\log(1/\varepsilon)/c'$. In total, we require at most

$$\left(1 + \frac{\sqrt{n}}{c'} \cdot \log 1/\varepsilon\right)^n \leq \left(\frac{\sqrt{n}}{c'}\right)^n (1+\log 1/\varepsilon)^n \leq 2^{cn\log n}(1+\log 1/\varepsilon)^n$$

boxes to cover $[\varepsilon,1]^n$. Since by Corollary 2.2, each of these boxes can be covered by an ellipsoid contained in $H'$, this completes the proof. □

#### 2.1.1 A lower bound for axis parallel ellipsoids

Can the dependence on $n$ be improved? Note that the volume of $H_\varepsilon$ is $(2-2\varepsilon)^n$, whereas the largest ellipsoid contained in $H$ is the $n$-dimensional euclidean ball with radius 1 centered in 0 which is of volume $2^{-\Omega(n\log n)}$. So for fixed $\varepsilon \in (0, \frac{1}{2})$, simply by accounting for volume it is clear that we need at least $2^{\Omega(n\log n)}$ ellipsoids.

What about the dependence on $\varepsilon$? This seems to be a more difficult question. We can prove the following.

THEOREM 2.4. *Fix the dimension $n \geq 2$. There exists a constant $c_n > 0$, depending only on $n$, such that for all $\varepsilon \in (0,1)$, any covering of $H_\varepsilon$ by axis parallel ellipsoids contained in $H$ consists of at least $c_n \cdot (1+\lfloor \log 1/\varepsilon \rfloor)^{n-1}$ ellipsoids.*

PROOF. To simplify the argument, we again transform the problem so that we can work entirely within the positive orthant. Consider the grid

$$G_\varepsilon := \{v \in \mathbb{R}^n \mid \ v_j = 2^{-\alpha_j} \geq \varepsilon \text{ with } \alpha_j \in \mathbb{N}_0 \text{ for all } 1 \leq j \leq n\}$$

Every covering of $H_\varepsilon$ using $m$ axis parallel ellipsoids contained in $H$ corresponds, by an affine transformation, to a covering of $G_\varepsilon$ using axis parallel ellipsoids $E_1,\ldots,E_m \subset \mathbb{R}^n_{\geq 0}$. We assume that the ellipsoids touch all coordinate hyperplanes. This is without loss of generality because otherwise we can grow the ellipsoids by scaling (independent in each dimension) around their centers by an adequate factor. The ellipsoids can then be described as

$$E_i = \left\{ x \in \mathbb{R}^n \mid \sum_{j=1}^{n} \left( \frac{2^{-\mu_{ij}} - x_j}{2^{-\mu_{ij}}} \right)^2 \leq 1 \right\},$$

where the center of $E_i$ is at $(2^{-\mu_{i1}}, \ldots, 2^{-\mu_{in}})$.

We will proceed to give an upper bound on the number $|E_i \cap G_\varepsilon|$ of grid points contained in an ellipsoid. Let $v = (2^{-\alpha_j})_{j=1}^n \in E_i \cap G_\varepsilon$.

$$1 \geq \sum_{j=1}^{n} \left( \frac{2^{-\mu_{ij}} - 2^{-\alpha_j}}{2^{-\mu_{ij}}} \right)^2 = \sum_{j=1}^{n} \left(1 - 2^{\mu_{ij} - \alpha_j}\right)^2$$

At most one summand – say the $k$-th – can be greater than one half. We then must have $(1-2^{\mu_{ij}-\alpha_j})^2 \leq \frac{1}{2}$ for all $j \neq k$. A rough calculation shows $-2 < \mu_{ij} - \alpha_j < 1$, so there are at most 3 possible choices of $\alpha_j \in \mathbb{N}_0$ for every $j \neq k$. On the other hand, $\alpha_k$ can take any integer value between 0 and $\left\lfloor \log \frac{1}{\varepsilon} \right\rfloor$. Finally, there are $n$ choices for $k$, giving the upper bound of

$$|E_i \cap G_\varepsilon| \leq n 3^{n-1} \left(1 + \lfloor \log 1/\varepsilon \rfloor\right).$$

Combining this with the total number of grid points, we get

$$\begin{aligned}(1 + \lfloor \log 1/\varepsilon \rfloor)^n = |G_\varepsilon| &\leq \sum_{i=1}^{m} |E_i \cap G_\varepsilon| \\ &\leq m n 3^{n-1} \left(1 + \lfloor \log 1/\varepsilon \rfloor\right).\end{aligned}$$

The statement of the theorem follows by defining the constant $c_n := (n3^{n-1})^{-1}$. □

Note that this proof only works for axis parallel ellipsoids. It seems implausible that allowing arbitrary ellipsoids could yield significantly more efficient coverings.

## 2.2 Covering with parallelepipeds

The goal is to cover $H_\varepsilon = [-1+\varepsilon, 1-\varepsilon]^n$ by parallelepipeds that, if scaled by 2, are contained in $H = [-1,1]^n$. The smallest number of such parallelepipeds is $P(n,\varepsilon)$. We again provide an axis-parallel covering. This time, however, we derive a lower bound that is asymptotically tight in the exponent, even for non-axis-parallel parallelepipeds. We remark that the results of this sections hold with only minor numerical changes for any constant scaling factor. We fix the factor 2 for concreteness and to simplify the presentation. First, we need an elementary lemma whose proof is straightforward. See Figure 2 for an illustration.

LEMMA 2.5. *Let $v \in (0,1]^n$ and $U = [1-v_1, 1-v_1/3] \times \ldots \times [1-v_n, 1-v_n/3]$. If $U$ is scaled by a factor of 2 from its center of gravity, then it is still contained in $[-1,1]^n$.*

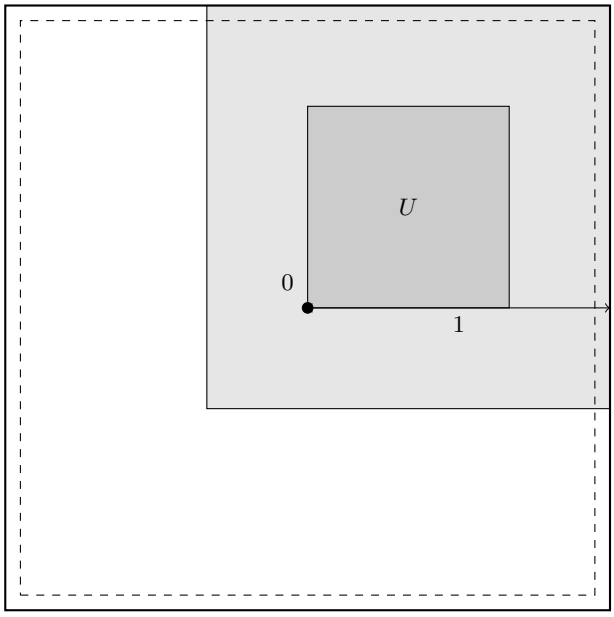



**Figure 2: An illustration of Lemma 2.5 for $v_1 = \cdots = v_n = 1$.**

THEOREM 2.6. *One has $P(n,\varepsilon) \leq 2^n (1 + \log 1/\varepsilon)^n$.*

PROOF. We proceed by covering $[0, 1-\varepsilon]^n$ by boxes that, if scaled by two, are contained in $[-1,1]^n$. Consider a box of the form

$$\begin{aligned}U(\alpha) = &\left[1 - 3^{-\alpha_1}, 1 - 3^{-\alpha_1 - 1}\right] \times \cdots \times \\ &\left[1 - 3^{-\alpha_n}, 1 - 3^{-\alpha_n - 1}\right], \quad \alpha \in \mathbb{N}_0^n.\end{aligned}$$

By Lemma 2.5 these boxes are still contained in $H$ after they are scaled by 2. How many of these boxes are needed to cover $[0, 1-\varepsilon]^n$? We only have to consider $U(\alpha)$ with $3^{-\alpha_j} > \varepsilon$ for all $j$. Taking logarithms, this implies $\alpha_j < \frac{\log(1/\varepsilon)}{\log 3}$. Thus we need at most $(1 + \log 1/\varepsilon)^n$ boxes. Repeating the procedure for each orthant yields the desired bound. □

We refer to Figure 3 for an illustration of the covering scheme.

### 2.2.1 A lower bound

The approach described in the previous section can be thought of, in a more general form, as the problem of covering the cube $H_\varepsilon$ using affine copies of a fixed centrally symmetric convex body $K$, such that constant multiples of the copies are still contained in $H$. We will show that the number of parallelepipeds is optimal as far as the growth of the exponents is concerned.The proof is analogous to that of Theorem 2.4.

THEOREM 2.7. *Let $K \subset \mathbb{R}^n$ be a centrally symmetric body. Let $K_1, \ldots, K_m$ be affine copies of $K$ and let $K_j'$ be the result of scaling $K_j$ by a factor of 2 around its center point. Suppose that $K_j' \subseteq H$ for all $j$, and $K_1, \ldots, K_m$ together cover $H^\varepsilon$. Then $m \geq c_n (1 + \lfloor \log 1/\varepsilon \rfloor)^n$, where $c_n > 0$ only depends on $n$.*

PROOF. By translating the given bodies, we can instead consider a situation where $[\varepsilon, 1]^n$ is covered by $K_1, \ldots, K_m$

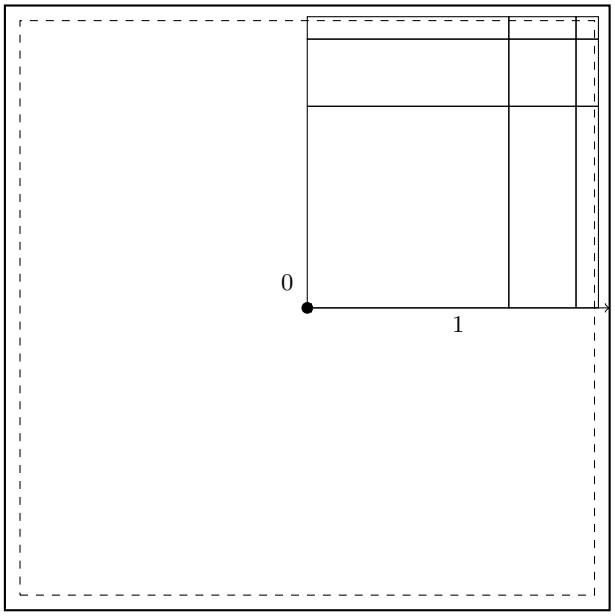



**Figure 3: Covering one orthant with boxes of type $U(\alpha)$.**

and $K'_j \subset \mathbb{R}^n_{\geq 0}$ for all $j$. In particular, this means that the grid

$$G^\varepsilon := \{v \in \mathbb{R}^n \mid v_j = 2^{-\alpha_j} \geq \varepsilon \text{ with } \alpha_j \in \mathbb{N}_0 \text{ for all } 1 \leq j \leq n\}$$

is covered. Let us now determine the number of grid points contained in each $K_j$. Let $a_j$ be the center point of $K_j$. We have

$$K_j \subseteq \{x \in \mathbb{R}^n \mid \frac{1}{2}a_j \leq x \leq \frac{3}{2}a_j\},$$

where the first set of inequalities follows from the fact that $K'_j \subset \mathbb{R}^n_{\geq 0}$, and the second set of inequalities follows from central symmetry of $K_j$. There are at most two choices for $\alpha_i \in \mathbb{N}_0$ such that $x_i = 2^{-\alpha_i}$ satisfies the corresponding lower and upper bound. Consequently, $K_j$ contains at most $2^n$ grid points. Recall that the total number of grid points is $(1 + \lfloor \log 1/\varepsilon \rfloor)^n$, from which the statement of the theorem follows. □

# 3. THE APPROXIMATION ALGORITHM

We now present our $(1+\varepsilon)$-approximation algorithm for the closest vector problem in the $\ell_\infty$ norm. We describe a boosting technique that turns any constant factor approximation algorithm for $\mathrm{CVP}_\infty$ into a $(1+\varepsilon)$-approximation algorithm at the expense of an additional factor of

$$2^{O(n)} \left(\log 1/\varepsilon\right)^{O(n)} b^{O(1)}$$

in the running time, where $b$ denotes the encoding length of the input. It is a Karp reduction approach, i.e. the constant factor approximation algorithm is used as an oracle and called multiple times on different inputs.

We first consider the *$\alpha$-gap* $\mathrm{CVP}_\infty$ problem, which is defined as follows. Given a lattice $\Lambda(A)$, a target vector $t$ and a number $D > 0$, either find a lattice vector $v \in \Lambda(A)$ with $\|v - t\|_\infty \leq D$, or assert that all lattice vectors have distance more than $\alpha^{-1}D$. We show how to construct a $(1+\varepsilon)$-gap algorithm for $\mathrm{CVP}_\infty$ from a 2-gap algorithm using the covering with parallelepipeds described in Section 2.2.

Afterwards we describe a binary search procedure to obtain a $(1+\varepsilon)$-approximation algorithm, using the $(1+\varepsilon)$-gap algorithm as an oracle in each iteration of the binary search.

We plug the currently fastest known constant approximation solver, the Blömer and Naewe (BN) algorithm [BN09], into our construction and boost its success probability so that we obtain the following approximation algorithm.

THEOREM 3.1. *For every $\varepsilon \in (0,1)$, there is a randomized algorithm that $(1+\varepsilon)$-approximates $\mathrm{CVP}_\infty$ in time*

$$2^{O(n)}(\log 1/\varepsilon)^{O(n)} b^{O(1)}$$

*with success probability $1 - 2^{-\Omega(n)}$.*

The randomness is due to the fact that the BN algorithm is randomized. Our construction is deterministic.

## 3.1 Boosting gap solvers

We now describe the $(1+\varepsilon)$-gap algorithm for $\mathrm{CVP}_\infty$ with the following properties.

THEOREM 3.2. *Given an oracle that solves 2-gap $\mathrm{CVP}_\infty$, for every $\varepsilon \in (0,1]$ we can solve $(1+\varepsilon)$-gap $\mathrm{CVP}_\infty$ using at most $2^n \cdot (2 + \log 1/\varepsilon)^n$ oracle calls.*

*The encoding size of instances for each oracle query are polynomial in $n$, the original encoding length and in $\log 1/\varepsilon$.*

In fact, any constant-gap oracle could be used. We choose to fix the approximation factor to 2 for concreteness and to simplify the presentation.

Let $(B,\ t,\ D)$ be the input. To solve the $(1+\varepsilon)$-gap problem, we either need to find a vector $v \in \Lambda(B)$ with $\|v - t\|_\infty \leq D$, or assert that the box

$$T := t + D \cdot [-1+\delta, 1-\delta]^n$$

with $1 - \delta = 1/(1+\varepsilon)$ does not contain a lattice point. By scaling the instance, we can assume without loss of generality that $D = 1$. Hence the box $T$ is a translate of the box $H_\delta = [-1+\delta, 1-\delta]^n$. As discussed in Section 2.2, there is a covering of $H_\delta$ and therefore $T$ with singly exponential many parallelepipeds. These parallelepipeds have the property that if they are scaled by a factor of 2 around their center of gravity, then they are still contained within $t + [-1,1]^n$. This is useful because with one call to a 2-approximation oracle for 2-gap $CVP_\infty$, we can either find a lattice vector with distance at most 1 or assert that one of the parallelepipeds does not contain a lattice vector, as we show in the following lemma.

LEMMA 3.3. *Given a lattice $\Lambda(A)$ and a parallelepiped $P := \{x \in \mathbb{R}^n : \|E(x-d)\|_\infty \leq 1\}$, a single call to a 2-gap oracle for $CVP_\infty$ either asserts that $P \cap \Lambda(A) = \emptyset$ or finds a lattice vector $v \in \Lambda(A)$ contained in $P^s := \{x \in \mathbb{R}^n : \|E(x-d)\|_\infty \leq 2\}$, i.e. $P$ scaled by 2 around its center of gravity $d$.*

PROOF. Define $B := E \cdot A$ and $t := E \cdot d$ and observe that $P \cap \Lambda(B) \neq \emptyset$ ($P^s \cap \Lambda(B) \neq \emptyset$) if and only if there is a vector $v \in \Lambda(B)$ with distance at most 1 (2) from $t$. □

PROOF OF THEOREM 3.2. Let $(B,\ t,\ D)$ be the input. By scaling the instance, we can assume without loss of generality that $D = 1$. Let $\delta = \frac{\varepsilon}{1+\varepsilon}$, so that $1 - \delta = \frac{1}{1+\varepsilon}$. Our

goal is to either assert that the box $T = t + [-1+\delta, 1-\delta]^n$ is empty or to find a lattice vector in $t + [-1,1]^n$.

Let $P_1, \ldots, P_k$ with $k \leq 2^n \cdot \left(2 + \log(\frac{1}{\varepsilon})\right)^n$ be parallelepipeds as in Theorem 2.6. Moreover for each $i$ let $P_i^s$ be the parallelepiped $P_i$ scaled by a factor of 2 around its center of gravity. Then $P_1, \ldots, P_k$ cover $T$ and $P_i^s \subseteq t + [-1,1]^n$ for each $i$. Lemma 3.3 shows that for each $i$, a single call to the 2-gap $CVP_\infty$ oracle either yields a lattice vector in $t + [-1,1]^n$ or asserts that $P_i$ does not contain a lattice vector. Since the parallelepipeds cover $T$, if the answers for all oracle calls are negative, we can assert that $T$ does not contain a lattice vector.

Finally note that the amount of scaling applied before each oracle call is bounded by $O(1/\varepsilon)$. Therefore, the desired bound bound for the encoding size of the instances for each oracle call holds. □

## 3.2 Approximating the closest vector problem

In this section we first describe a procedure to Karp-reduce the problem of computing a $(1+\varepsilon)$-approximation for $\mathrm{CVP}_\infty$ to $(1+O(\varepsilon))$-gap $\mathrm{CVP}_\infty$. Then we combine our constructions with the BN algorithm to obtain the currently fastest (randomized) $(1+\varepsilon)$-approximation algorithm for $\mathrm{CVP}_\infty$.

THEOREM 3.4. *For every $\varepsilon \in (0,1)$ and $\delta := \min\{\frac{\varepsilon}{5}, \frac{1}{2}\}$, given access to a $(1+\delta)$-gap $\mathrm{CVP}_\infty$ oracle, one can compute a $(1+\varepsilon)$-approximation for $\mathrm{CVP}_\infty$ using*

$$O(\log b + \log n + \log 1/\varepsilon)$$

*calls to the oracle.*

We are given as input a lattice $\Lambda = \Lambda(A)$ and target vector $t$. Let us assume that the distance $d(t,\Lambda)$ of a closest vector to the target vector is between 1 and at most $2^{cn^2 \cdot b}$ for some constant $c > 0$. This can be achieved by scaling, see [BN09]. We then perform a simple binary search in the following way:

1. Set $\delta := \min\{\varepsilon/5, 1/2\}$
2. Initialize $L \leftarrow 0$ and $U \leftarrow \left\lceil \log_{1+\delta} 2^{cn^2 \cdot b} \right\rceil$.
3. While $U - L \geq 3$, do a binary search step:
   (a) Solve the $(1+\delta)$-gap problem with input $(A, t, (1+\delta)^{L + \lceil (U-L)/2 \rceil})$.
   (b) If a lattice vector $v$ is returned, update $U \leftarrow \lceil \log_{1+\delta} \|v - t\|_\infty \rceil$.
   (c) Otherwise, update $L \leftarrow L + \lceil (U-L)/2 \rceil - 1$.
4. Solve the $(1+\delta)$-gap problem with input $(A, t, (1+\delta)^{U+1})$ and return the resulting lattice vector.

We first prove the correctness of this procedure before we analyze its running time.

LEMMA 3.5. *The algorithm from above has the following properties.*

1. *The binary search routine maintains the invariant that $(1+\delta)^L \leq d(t,\Lambda) \leq (1+\delta)^U$.*
2. *The algorithm returns a lattice vector $v$ that satisfies $\|v - t\|_\infty \leq (1+\varepsilon) d(t,\Lambda)$.*

PROOF. 1. The initial choices of $L$ and $U$ are appropriate after scaling the lattice as mentioned in the beginning of this section. In the case 3(b), the existence of the lattice vector $v$ proves that the invariant is maintained by the update of $U$. In the case 3(c), that is, when the $(1+\delta)$-gap problem does not return a lattice vector, this implies by definition that $d(t,\Lambda) \geq (1+\delta)^{L + \lceil (U-L)/2 \rceil - 1}$ and so the invariant is maintained.

2. In the end, we know that $d(t,\Lambda) \leq (1+\delta)^U$, so the final application of the $(1+\delta)$-gap problem is guaranteed to find a lattice vector $v$. This lattice vector satisfies

$$\begin{aligned} \|v - t\| &\leq (1+\delta)^{U+1} \\ &\leq (1+\delta)^{L+3} \\ &\leq (1+\delta)^3 d(t,\Lambda) \\ &\leq (1+5\delta) d(t,\Lambda) \\ &\leq (1+\varepsilon) d(t,\Lambda). \end{aligned}$$

For the second inequality, we used the fact that $U - L$ is an integer and therefore $U - L \leq 2$. □

PROOF OF THEOREM 3.4. Correctness of the procedure has already been shown in Lemma 3.5. It remains to bound the number of oracle calls. Let $M_j$ be the difference $U - L$ after the $j$-th search step. By the initial choices of $L$ and $U$ we have

$$M_0 = \left\lceil \log_{1+\delta} 2^{cn^2 \cdot b} \right\rceil = \left\lceil \frac{cn^2 b}{\log(1+\delta)} \right\rceil \leq c' n^2 b/\delta$$

for some constant $c' > 0$. Let us analyze what happens in step $j$. In the case 3(b), we know that $\|v - t\|_\infty \leq (1+\delta)^{\lceil L + (U-L)/2 \rceil}$. Denoting the updated value of $U$ by $U'$, this implies $U' \leq L + \lceil (U-L)/2 \rceil$, and so $M_j \leq \lceil (U-L)/2 \rceil \leq M_{j-1}/2 + 1$.

In the case 3(c), we get

$$\begin{aligned} M_j &= U - (L + \lceil (U-L)/2 \rceil - 1) \\ &= M_{j-1} - \lceil M_{j-1}/2 \rceil + 1 \\ &\leq M_{j-1}/2 + 1. \end{aligned}$$

We get the same upper bound in both cases and can conclude using induction that

$$M_j \leq 2^{-j} M_0 + 1 + \frac{1}{2} + \frac{1}{4} + \cdots \leq 2^{-j} M_0 + 2.$$

This implies that the number of steps is bounded by $\lceil \log M_0 \rceil$ because the iteration stops when $M_j$ drops below 3. From this we can derive the desired upper bound for the number of oracle calls. □

We now prove the main theorem by using the BN algorithm as a 2-approximation and applying the boosting technique for gap CVP combined with the binary search procedure. As their algorithm is randomized, one has to take care of the success probabilities. Their algorithm has a failure probability of $2^{-\Omega(n)}$. Considering the amount of oracle queries we have to issue and the requirement that *every* call has to be successful, that failure probability is too high, so we boost the success probability using standard techniques.

PROOF OF THEOREM 3.1. The BN algorithm has a success probability of at least $1 - 2^{-c \cdot n}$ for some constant $c > 0$

and a running time of $(2+\frac{1}{\varepsilon'})^{O(n)} \cdot b^{O(1)}$ when used as a $(1+\varepsilon')$-approximation algorithm.

Set $a := c' \cdot \left\lceil (1+\max\{\log\log 1/\varepsilon,\ 1\} + \frac{1}{n}\log b) \right\rceil$ for an appropriate constant $c' > 0$ that will be determined later. Let BN+ be an algorithm that runs BN as a 2-approximation algorithm $a$ times on the same input and returns the closest vector that was found among all runs. This aggregated algorithm is a 2-approximation algorithm with a running time of $\max\{\log\log\left(\frac{1}{\varepsilon}\right), 1\} \cdot 2^{O(n)} \cdot b^{O(1)}$ and success probability at least

$$1 - 2^{-acn} \geq 1 - 2^{-c'cn} (\log 1/\varepsilon)^{-cc'n} \cdot b^{-cc'}.$$

Using the boosting technique from Theorem 3.2, we can construct a $(1+\delta)$-gap algorithm with

$$\delta := \min\{\varepsilon/5, 1/2\},$$

using BN+ as a 2-gap oracle. This amounts to a running time of $2^{O(n)} \cdot (\log 1/\varepsilon)^{O(n)} \cdot b^{O(1)}$ for the $(1+\delta)$-gap algorithm. Plugging this as a black-box into the binary search procedure, we get a $(1+\varepsilon)$-approximation algorithm by Theorem 3.4. Moreover, the number of calls to the $(1+\delta)$-gap algorithm is bounded by $O(\log n + \log b + \log 1/\varepsilon)$. Thus in total we get the desired running time bound of

$$2^{O(n)} (\log 1/\varepsilon)^{O(n)} b^{O(1)}$$

which is also an upper bound to the number of calls to BN+. The probability for failure of the $(1+\varepsilon)$-approximation algorithm is bounded by the probability that one of the runs of BN+ fails. By choosing $c'$ large enough, we get an upper bound of $2^{-\Omega(n)}$ for the failure probability from the union bound.

We remark that, although the encoding size for some of the instances we query the oracle for may exceed $b$, it always stays within $\mathrm{poly}(n, \log 1/\varepsilon, b)$ by Theorem 3.2 so the asymptotic running time indicated above is not affected. □

## Acknowledgment

We would like to thank János Pach for discussions on coverings and for pointing us to [ER62].

## 4. REFERENCES

[AKS01] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, pages 601–610 (electronic), New York, 2001. ACM.

[AKS02] M. Ajtai, R. Kumar, and D. Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In *Computational Complexity, 2002. Proceedings. 17th IEEE Annual Conference on*, 2002.

[Aro94] S. Arora. *Probabilistic checking of proofs and the hardness of approximation problems*. PhD thesis, UC Berkeley, 1994.

[BN09] J. Blömer and S. Naewe. Sampling methods for shortest vectors, closest vectors and successive minima. *Theoretical Computer Science*, 410(18):1648–1665, 2009.

[DKRS03] I. Dinur, G. Kindler, R. Raz, and S. Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003.

[DPV10] Daniel Dadush, Chris Peikert, and Santosh Vempala. Enumerative algorithms for the shortest and closest lattice vector problems in any norm via M-ellipsoid coverings. Manuscript, 2010.

[ER62] P. Erdős and C. A. Rogers. Covering space with convex bodies. *Acta Arith.*, 7:281–285, 1961/1962.

[FK08] Z. Füredi and J.-H. Kang. Covering the $n$-space by convex bodies and its chromatic number. *Discrete Math.*, 308(19):4495–4500, 2008.

[Kan87] R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):pp. 415–440, 1987.

[Len83] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538 – 548, 1983.

[MV10] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 351–358, New York, NY, USA, 2010. ACM.

[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.