# The closest vector problem in tensored root lattices of type A and in their duals

**Léo Ducas · Wessel P.J. van Woerden**

September 21, 2016

**Abstract** In this work we consider the closest vector problem (CVP) —a problem also known as maximum-likelihood decoding— in the tensor of two root lattices of type A ($A_m \otimes A_n$), as well as in their duals ($A_m^* \otimes A_n^*$). This problem is mainly motivated by *lattice based cryptography*, where the cyclotomic rings $\mathbb{Z}[\zeta_c]$ (resp. its co-different $\mathbb{Z}[\zeta_c]^\vee$) play a central role, and turn out to be isomorphic as lattices to tensors of $A^*$ lattices (resp. $A$ root lattices). In particular, our results lead to solving CVP in $\mathbb{Z}[\zeta_c]$ and in $\mathbb{Z}[\zeta_c]^\vee$ for conductors of the form $c = 2^\alpha p^\beta q^\gamma$ for any two odd primes $p, q$.

For the primal case $A_m \otimes A_n$, we provide a full characterization of the Voronoi region in terms of simple cycles in the complete directed bipartite graph $K_{m+1,n+1}$. This leads —relying on the Bellman-Ford algorithm for negative cycle detection— to a CVP algorithm running in *polynomial time*. Precisely, our algorithm performs $O(l\ m^2 n^2 \min\{m, n\})$ operations on reals, where $l$ is the number of bits per coordinate of the input target. For the dual case, we use a gluing-construction to solve CVP in sub-exponential time $O(nm^{n+1})$.

**Keywords** Lattice based cryptography, Cyclotomic lattices, Tensored root lattices, Closest vector problem, Maximum likelihood decoding.

This work has been supported by a grant from CWI from budget for public-private-partnerships and in part by a grant from NXP Semiconductors.

L. Ducas
CWI (Centrum Wiskunde & Informatica), Amsterdam, The Netherlands.
E-mail: ducas@cwi.nl

W.P.J. van Woerden
Mathematical Institute and LIACS, Leiden University, The Netherlands.
E-mail: wesselvanwoerden@gmail.com

## 1 Introduction

The root lattices and their duals are well known distinguished lattices, and their application as lattice-codes and as quantizers are well understood [4,5]. In particular, quasi-linear time algorithms [3,9] are known for the root lattice $A_n$ and its dual $A_n^*$ (see definition 7), or even linear time ones [10].

In this work, we are interested in generalizing those results to tensors of such lattices. A motivation could be to use root lattices as building blocks for larger lattice-codes and quantizers. But more naturally, those tensors appear when considering cyclotomic rings $\mathbb{Z}[\zeta_c]$ as lattices[1] via the Minkowski embedding [12,8]. The structure of $\mathbb{Z}[\zeta_c]$ as a lattice may be deduced inductively from the following lattice isomorphisms [2]

$$\begin{aligned}
\mathbb{Z}[\zeta_{2^k}] &\simeq \mathbb{Z}^{2^{k-1}}, \\
\mathbb{Z}[\zeta_p] &\simeq A_{p-1}^*, && p \text{ an odd prime}, \\
\mathbb{Z}[\zeta_{p^k}] &\simeq \bigoplus_{i=1}^{p^{k-1}} \mathbb{Z}[\zeta_p], && p \text{ a prime}, \\
\mathbb{Z}[\zeta_{mn}] &\simeq \mathbb{Z}[\zeta_m] \otimes \mathbb{Z}[\zeta_n] && m, n \text{ coprime}.
\end{aligned}$$

We note that the direct sum $\oplus$ of lattices is very easy to handle (see Lemma 4), and so are tensors with $\mathbb{Z}^\ell$ thanks to the identity $\mathbb{Z}^\ell \otimes \Lambda \simeq \bigoplus_{i=1}^{\ell} \Lambda$. Therefore treating CVP in lattices $A_m^* \otimes A_n^*$ suffices to solve CVP in $\mathbb{Z}[\zeta_c]$ for any $c = 2^\alpha p^\beta q^\gamma$, where $p, q$ are any odd primes. In a dual fashion, solving CVP in $A_m \otimes A_n$ leads to a solution for CVP in *the co-different* ideal $\mathbb{Z}[\zeta_c]^\vee$ of the ring $\mathbb{Z}[\zeta_c]$.

*Cryptographic motivations* Our motivation to solve CVP in $\mathbb{Z}[\zeta_c]$ and $\mathbb{Z}[\zeta_c]^\vee$ comes from ideal-lattice based cryptography. The worst-case to average-case reduction of Lyubashevsky et al. [7] has given a central role to cyclotomic rings in this field of research. One key step in such cryptosystems is to decode in the lattice $\mathbb{Z}[\zeta_c]^\vee$, and —unless $c$ is a power of 2— then only approximated CVP algorithms were considered, relying on special decoding bases [8].

Improving this step using an exact CVP algorithm would lead to improve those cryptosystems (better error tolerance, and therefore smaller parameters). Theoretically, it would also bring the satisfaction that the decoding algorithm respect the symmetry of the lattice. Note that our remark that $\mathbb{Z}[\zeta_p]^\vee \simeq A_{p-1}$ already trivializes this question for $c$ a prime or even when $c = 2^\alpha p^\beta$.

*Contributions* Our main contribution is a polynomial-time algorithm to solve CVP in the lattice $A_m \otimes A_n$, more precisely an algorithm performing $O(ln^2m^2\min\{m,n\})$ operations on reals, where $l$ is the number of bits per coordinate of the input target. This gives a satisfactory solution to our cryptographic application for any $c = 2^\alpha p^\beta q^\gamma$.

[1] We remind that this lattice has dimension $\varphi(c)$, the Euler totient of $c$.

[2] Such details are out of the scope of this paper, but are described in the B.S. Thesis of the second author, available online `https://www.math.leidenuniv.nl/scripties/BachVanWoerden.pdf`

This algorithm is derived from a very explicit characterization of the Voronoi region of the lattice $A_m \otimes A_n$, which is expressed —perhaps surprisingly— as a one-to-one map between the Voronoi-relevant vectors and the simple cycles in the complete bipartite directed graph $K_{m+1,n+1}$.

As a secondary contribution, we also study the dual case $A_m^* \otimes A_n^*$, for which we obtain only a weaker result: an algorithm in time $O(nm^{n+1})$, which is at worse sub-exponential $2^{\tilde{O}(\sqrt{mn})}$ in the dimension $mn$ —assuming without loss of generality that $n \leq m$. This result is obtained by classical gluing theory, with a small *completion trick*.

*Open problem* The most natural open problem is to improve the result for the dual case, ideally to a polynomial runtime algorithm. It would be even nicer if it would come with a characterization of its Voronoi cell, and if it'd respect the symmetry between $m$ and $n$, as in the primal case.

One could also be curious about the case of tensors of *three* root lattices $A_\ell \otimes A_m \otimes A_n$ or more. But it would of course also be interesting to improve further the polynomial running-time for CVP in $A_m \otimes A_n$.

*Plan* In the preliminaries (Section 2), we review the definitions of lattices, Voronoi region, direct sums and tensor products of lattices, as well as the definitions and basic properties of the root lattices $A_n$ and their duals $A_n^*$. In Section 3, we describe our sub-exponential time algorithm for CVP in $A_m^* \otimes A_n^*$. The last section (Section 4) presents our main result, the polynomial time algorithm for CVP in $A_m \otimes A_n$.

## 2 Preliminaries

For the rest of the paper, we fix two integers $m, n \geq 1$ and let $m' = m + 1$ and $n' = n + 1$.

### 2.1 Lattices, and the Closest Vector Problem

**Definition 1 (Lattice)** A *lattice* $\Lambda$ with $\mathbb{R}$-linearly independent *(lattice) basis* vectors $\boldsymbol{b_1}, \ldots, \boldsymbol{b_r} \in \mathbb{R}^d$ is the discrete additive subgroup

$$\Lambda := \left\{ \sum_{i=1}^{r} z_i \boldsymbol{b_i} : z_i \in \mathbb{Z} \right\}$$

of $\mathbb{R}^d$. Let $\boldsymbol{B} \in \mathbb{R}^{r \times d}$ be the matrix with rows $\boldsymbol{b_1}, \ldots, \boldsymbol{b_r}$. We say that $\Lambda$ has *rank* $r$ and *generator matrix* $\boldsymbol{B}$. Let $\mathrm{span}(\Lambda)$ be the linear subspace of $\mathbb{R}^d$ spanned by the elements of $\Lambda$ over $\mathbb{R}$.

The *shortest vectors* of $\Lambda$ are the nonzero points of $\Lambda$ with minimal norm. If $\boldsymbol{v} \in \Lambda$ is a shortest vector then $\rho = \frac{\|\boldsymbol{v}\|}{2}$ is the *packing radius* of $\Lambda$. The *covering radius* $R$ is the minimal distance such that any point in $\mathrm{span}(\Lambda)$ is at distance at most $R$ to a lattice point. Another lattice $\Lambda' \subset \mathbb{R}^d$ of the same rank $r$ such that $\Lambda' \subset \Lambda$ is called a full rank sublattice of $\Lambda$.

**Definition 2 (Closest Vector Problem)** Let $\Lambda \subset \mathbb{R}^d$ be a lattice. Given an arbitrary point $\boldsymbol{t} \in \mathrm{span}(\Lambda)$, the goal is to find a closest lattice point of $\Lambda$ to $\boldsymbol{t}$, i.e., an $\boldsymbol{x} \in \Lambda$ that minimizes the distance $\|\boldsymbol{t}-\boldsymbol{x}\| := \sqrt{\langle \boldsymbol{t}-\boldsymbol{x}, \boldsymbol{t}-\boldsymbol{x}\rangle}$. Such an $\boldsymbol{x}$ is also called a *closest vector* to $\boldsymbol{t}$.

A natural geometric body associated with the closest vector problem is the Voronoi region, defined below.

**Definition 3 (Voronoi region and relevant vectors)** Let $H_{\boldsymbol{v}} = \{\boldsymbol{x} \in \mathrm{span}(\Lambda) : \|\boldsymbol{x}\| \leq \|\boldsymbol{x}-\boldsymbol{v}\|\}$ for $\boldsymbol{v} \in \Lambda$ be the half space consisting of points at least as close to $\mathbf{0}$ as to $\boldsymbol{v}$. The Voronoi region (around $\mathbf{0}$) of a lattice $\Lambda$ is defined by

$$V(\Lambda) := \bigcap_{\boldsymbol{v} \in \Lambda} H_{\boldsymbol{v}},$$

consisting of all points in $\mathrm{span}(\Lambda)$ that have $\mathbf{0}$ as a closest vector. It is easy to confirm that the Voronoi region is a convex polytope which is symmetric by reflection in $\mathbf{0}$ [5].

The *Voronoi relevant vectors* are the vectors forming the minimal set $RV(\Lambda) \subset \Lambda$ of vectors such that

$$V(\Lambda) = \bigcap_{\boldsymbol{v} \in RV(\Lambda)} H_{\boldsymbol{v}}.$$

Voronoi showed in [13] that for $\boldsymbol{v} \in \Lambda \setminus \{0\}$ we have that $\boldsymbol{v}$ is a Voronoi relevant vector iff $\mathbf{0}$ and $\boldsymbol{v}$ are the only closest vectors to $\frac{1}{2}\boldsymbol{v}$ in $\Lambda$, i.e. iff $\langle \boldsymbol{v}, \boldsymbol{x}\rangle < \langle \boldsymbol{x}, \boldsymbol{x}\rangle$ for all $\boldsymbol{x} \in \Lambda \setminus \{\mathbf{0}, \boldsymbol{v}\}$. Interestingly, Voronoi relevant vectors suffice to decide if a lattice vector is a closest vector to a given target, and if not, to find a closer one. This gave rise to generic CVP algorithms, running in exponential time [11,2].

**Lemma 1** *Let $\boldsymbol{t} \in \mathrm{span}(\Lambda)$ and $\boldsymbol{x} \in \Lambda$. There exists a vector $\boldsymbol{y} \in \Lambda$ such that $\|(\boldsymbol{x}+\boldsymbol{y})-\boldsymbol{t}\| < \|\boldsymbol{x}-\boldsymbol{t}\|$ iff there exists a Voronoi relevant vector $\boldsymbol{v} \in RV(\Lambda)$ such that $\|(\boldsymbol{x}+\boldsymbol{v})-\boldsymbol{t}\| < \|\boldsymbol{x}-\boldsymbol{t}\|$.*

*Proof* The implication from right to left is trivial by taking $\boldsymbol{y} = \boldsymbol{v}$. Now suppose there exists a vector $\boldsymbol{y} \in \Lambda$ such that $\|(\boldsymbol{x}+\boldsymbol{y})-\boldsymbol{t}\| < \|\boldsymbol{x}-\boldsymbol{t}\|$. Then by definition $\boldsymbol{t}-\boldsymbol{x} \notin V(\Lambda)$. So there exists a $\boldsymbol{v} \in RV(\Lambda)$ such that $\|\boldsymbol{t}-\boldsymbol{x}\| > \|(\boldsymbol{t}-\boldsymbol{x})-\boldsymbol{v}\|$.

### 2.2 Combining lattices: sums, tensors and duals

**Definition 4 (Direct sum and orthogonal sum)** Let $\Lambda_1 \subset \mathbb{R}^{d_1}$ and $\Lambda_2 \subset \mathbb{R}^{d_2}$ be lattices of rank $r_1$ and $r_2$ respectively. Then the *direct sum* $\Lambda_1 \oplus \Lambda_2 \subset \mathbb{R}^{d_1+d_2}$ between $\Lambda_1$ and $\Lambda_2$ is defined as

$$\Lambda_1 \oplus \Lambda_2 = \{\boldsymbol{x}_1 \oplus \boldsymbol{x}_2 \in \mathbb{R}^{d_1+d_2} : \boldsymbol{x}_1 \in \Lambda_1, \boldsymbol{x}_2 \in \Lambda_2\}$$

where $\boldsymbol{x}_1 \oplus \boldsymbol{x}_2$ is just the concatenation of the two vectors. Note that the inner product between elements in $\Lambda_1$ or $\Lambda_2$ (embedded as $\boldsymbol{x}_1 \mapsto \boldsymbol{x}_1 \oplus \mathbf{0}$ and $\boldsymbol{x}_2 \mapsto \mathbf{0} \oplus \boldsymbol{x}_2$) stays the same and that each two elements $\boldsymbol{x}_1 \in \Lambda_1$ and $\boldsymbol{x}_2 \in \Lambda_2$ are orthogonal in $\Lambda_1 \oplus \Lambda_2$.

Let $\Lambda_1, \Lambda_2 \subset \mathbb{R}^d$ be lattices. Suppose $\Lambda_1$ has basis $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{r_1}$ and $\Lambda_2$ has basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{r_2}$. In the case that $\langle \boldsymbol{a}_i, \boldsymbol{b}_j\rangle = 0$ for all $i = 1, \ldots, r_1$ and $j = 1, \ldots, r_2$ we call $\Lambda_1$ and $\Lambda_2$ orthogonal and the *orthogonal sum* $\Lambda_1 \perp \Lambda_2$ between $\Lambda_1$ and $\Lambda_2$ is defined as the lattice with basis $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{r_1}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_{r_2}$.

**Definition 5 (Tensor product)** Let $\Lambda_1 \subset \mathbb{R}^{d_1}$ and $\Lambda_2 \subset \mathbb{R}^{d_2}$ be lattices of respective ranks $r_1$ and $r_2$ and let $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{r_1} \in \mathbb{R}^{d_1}$ and $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{r_2} \in \mathbb{R}^{d_2}$ be respective bases. The *tensor product* $\Lambda_1 \otimes \Lambda_2 \subset \mathbb{R}^{d_1 d_2}$ is defined as the lattice with basis $\{\boldsymbol{a}_i \otimes \boldsymbol{b}_j : i \in \{1, \ldots, r_1\}, j \in \{1, \ldots, r_2\}\}$. Here $\boldsymbol{x} \otimes \boldsymbol{y} = (x_1, \ldots, x_{d_1}) \otimes (y_1, \ldots, y_{d_1})$ with $\boldsymbol{x} \in \mathbb{R}^{d_1}$ and $\boldsymbol{y} \in \mathbb{R}^{d_2}$ is defined as the natural embedding in $\mathbb{R}^{d_1 d_2}$ as follows:

$$\boldsymbol{x} \otimes \boldsymbol{y} := (x_1 y_1, x_1 y_2, \ldots, x_1 y_{d_2}, x_2 y_1, \ldots, x_{d_1} y_{d_2}) \in \mathbb{R}^{d_1 d_2}.$$

**Definition 6 (Dual lattice)** For a lattice $\Lambda \subset \mathbb{R}^d$ its dual lattice $\Lambda^* \subset \mathbb{R}^d$ is defined as

$$\Lambda^* := \{\boldsymbol{y} \in \mathrm{span}(\Lambda) : \forall \boldsymbol{x} \in \Lambda, \langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \mathbb{Z}\}.$$

As expected we have the following identities:

$$\begin{aligned} (\Lambda^*)^* &= \Lambda \\ (\Lambda_1 \oplus \Lambda_2)^* &= \Lambda_1^* \oplus \Lambda_2^* \\ (\Lambda_1 \otimes \Lambda_2)^* &= \Lambda_1^* \otimes \Lambda_2^*. \end{aligned}$$

### 2.3 Root lattices of type $A$ and their duals

Root lattices emerge from so called root systems of vectors. There are three families of root lattices ($A, D$ and $E$), and they have been the object of very detailed studies [4,5,3,9,10] to cite a few. We recall the definition of the root lattice of type $A$ below, characterize its dual lattice, and provide bases for both.

**Definition 7 (root lattice $A_m$ [5])** Let $m \geq 1$. The lattice $A_m \subset \mathbb{R}^{m+1}$ of rank $m$ is defined as

$$A_m := \left\{ (x_1, \ldots, x_{m+1}) \in \mathbb{Z}^{m+1} : \sum_{i=1}^{m+1} x_i = 0 \right\},$$

i.e., all integer vectors of $\mathbb{Z}^{m+1}$ that sum up to zero.

**Lemma 2 (root lattice $A_m^*$ [5])** *The lattice $A_m^*$ dual to $A_m$ is*

$$A_m^* = \bigcup_{i=0}^{m} ([i] + A_m),$$

*where*

$$[i] = \left( \frac{i}{m'}, \ldots, \frac{i}{m'}, \frac{-j}{m'}, \ldots, \frac{-j}{m'} \right)$$

*has $j$ components equal to $\frac{i}{m'}$ and $i$ components equal to $\frac{j}{m'}$.*

In Sections 3 and 4 it will be usefull to know a basis for $A_m^*$ and $A_m$ respectively.

**Lemma 3 (bases of $A_m$ and $A_m^*$ [5])** *The $m \times (m+1)$-matrix $\boldsymbol{B}$ given by*

$$\boldsymbol{B} = \frac{1}{m+1} \begin{pmatrix} m & -1 & \dots & -1 & -1 \\ -1 & m & \dots & -1 & -1 \\ \vdots & & \ddots & \vdots & \vdots \\ -1 & -1 & \dots & m & -1 \end{pmatrix}$$

*with $\frac{m}{m+1}$ on the diagonal and $\frac{-1}{m+1}$ everywhere else is a generator matrix of $A_m^*$. Furthermore the vectors $\boldsymbol{b}^1, \dots, \boldsymbol{b}^m \in A_m$ given by $b_i^i = 1, b_{i+1}^i = -1$ and $0$ otherwise form a basis of $A_m$.*

## 3 Solving the closest vector problem in $A_m^* \otimes A_n^*$

*Overview* In this section, we make use of gluing theory [5, Chap. 4, Sec. 3, pp 99] to derive a sub-exponential time algorithm for CVP in $A_m^* \otimes A_n^*$. The most direct approach would consist of tensoring the glue constructions of $A_m^*$ and $A_n^*$ which would lead to an algorithm running in time $O((n')^{m+2} \cdot (m')^{n+2})$. Yet, thanks to a completion trick, we can decrease this complexity down to $O(n \cdot m^{n+1})$, which can be significantly better.

*Computational model* For this algorithm, we require only a very simple computational model, namely, circuits over real numbers with the arithmetic operations $\{+, -, \times\}$, a "compare-and-choose" gate operating on four inputs:

$$\text{cac} : (a, b, c, d) \mapsto \begin{cases} c & \text{if } a \leq b, \\ d & \text{otherwise.} \end{cases}$$

as well as a "round" gate: $\text{round} : a \mapsto \lfloor a \rceil$. Any gate may use a fixed constant as some of its input. The size of circuit is defined as the number of gates it requires.

**Definition 8** For a lattice $\Lambda \subset \mathbb{R}^d$, let $C(\Lambda)$ be the size of the smallest circuit as above, that given on input wires the coordinates $t_1, \dots, t_d$ of any vector $\mathbf{t} \in \text{span}(\Lambda)$ computes a closest vector to $\mathbf{t}$ in $\Lambda$.

We start with a basic lemma on solving CVP on a lattice written as a direct or orthogonal sum of smaller lattices. Amusingly, we mostly make use of the reciprocal property: solving CVP in the full lattice also solves it in any of its orthogonal components. This idea will allow us to perform the completion trick aforementioned.

**Lemma 4 (Direct sum and orthogonal sum)** *Let $\Lambda \subset \mathbb{R}^d$ be a lattice and let $\Lambda_1, \dots, \Lambda_k \subset \Lambda$ be orthogonal lattices of dimensions $r_1, \dots r_k$ such that:*

$$\Lambda = \Lambda_1 \perp \dots \perp \Lambda_k.$$

*Then:*

1. $C(\Lambda) \leq \sum_{i=1}^{k} C(\Lambda_i) + p_i + s_i$,
2. $C(\Lambda_i) \leq C(\Lambda)$ *for all* $i = 1, \dots, k$,

*Here, $p_i$ and $s_i$ denote the size of the minimal circuit to compute the orthogonal projection from* $\mathrm{span}(\Lambda)$ *to* $\mathrm{span}(\Lambda_i)$ *and the addition of two vectors from* $\mathrm{span}(\Lambda_i)$ *and* $\mathrm{span}(\Lambda_1 \perp \ldots \perp \Lambda_{i-1})$ *respectively. It also holds that $p_i \leq O(dr_i)$ and $s_i \leq d$.*

*If the sum is direct, i.e. if $\Lambda$ and $\Lambda_1, \ldots, \Lambda_k$ are lattices such that*

$$\Lambda = \Lambda_1 \oplus \ldots \oplus \Lambda_k,$$

*then we have the same inequalities with $p_i = s_i = 0$ for all $i = 1, \ldots, k$.*

*Proof* We start with the case of the orthogonal sum. For (1), suppose that $\boldsymbol{t} \in \mathrm{span}(\Lambda)$ is the target and $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_k$ are the projections onto $\mathrm{span}(\Lambda_1), \ldots, \mathrm{span}(\Lambda_k)$ of $\boldsymbol{t}$. For each $\boldsymbol{t}_i$ we can compute a closest vector $\boldsymbol{x}_i \in \Lambda_i$ in $C(\Lambda_i)$ operations. Then $\boldsymbol{x} = \boldsymbol{x}_1 + \ldots + \boldsymbol{x}_k \in \Lambda$ is a closest vector to $\boldsymbol{t}$ by the orthogonality. The projection and last summation take $p_i + s_i$ operations for each $i = 1, \ldots, k$. Note that the projection onto $\mathrm{span}(\Lambda_i)$ can be written as $\boldsymbol{x} \mapsto \boldsymbol{x}\boldsymbol{B}_i\boldsymbol{B}_i^t$ for some matrix $\boldsymbol{B}_i \in \mathbb{R}^{d \times r_i}$, therefore $p_i \leq O(r_i d)$. The inequality $s_i \leq d$ is straightforward.

For (2) suppose $\boldsymbol{t}_i \in \mathrm{span}(\Lambda_i) \subset \mathrm{span}(\Lambda)$ is our target. Suppose $\boldsymbol{x} \in \Lambda$ is a closest vector to $\boldsymbol{t}_i$ in $\Lambda$ which can be obtained in $C(\Lambda)$ operations. Then $\boldsymbol{x} \in \Lambda_i$ by the orthogonality because $\boldsymbol{t}_i \in \mathrm{span}(\Lambda_i)$ and thus $\boldsymbol{x}$ is a closest vector to $\boldsymbol{t}_i$ in $\Lambda_i$.

For the direct sum the proof is identical by using the embedding $\Lambda_i' = \boldsymbol{0} \oplus \ldots \oplus \Lambda_i \oplus \ldots \oplus \boldsymbol{0} \subset \Lambda$ such that $\Lambda = \Lambda_1' \perp \ldots \perp \Lambda_k'$. In this case the projections are along the coordinates and the summation is just concatenation and thus $p_i = s_i = 0$ for all $i = 1, \ldots, k$. □

Our second lemma allows to solve CVP in a lattice written as a union of cosets of a sparser lattice, i.e. a glue-construction.

**Lemma 5 (Gluing Lemma)** *Let $\Lambda \subset \mathbb{R}^d$ be a lattice and let $\Lambda' \subset \Lambda$ be a full rank sublattice. Note that $\Lambda$ consists of multiple translated copies of $\Lambda'$. To be more precise, we can see $\Lambda'$ as a subgroup of $\Lambda$, and then let $G = \Lambda/\Lambda'$ be the so called glue group consisting of cosets. Let $[\Lambda : \Lambda'] =: |G|$ denote the index of $\Lambda'$ in $\Lambda$ and let $\mathcal{G} \subset \Lambda$ be a set consisting of a single representative for each coset in $G$, so called glue vectors. Then*

$$\Lambda = \bigcup_{\boldsymbol{g} \in \mathcal{G}} (\boldsymbol{g} + \Lambda')$$

*and we have that*

$$C(\Lambda) \leq |G|(O(d) + C(\Lambda')).$$

*Proof* We make use of the fact that if $\boldsymbol{x} \in \Lambda$ is a closest vector to $\boldsymbol{t} \in \mathrm{span}(\Lambda)$ then $\boldsymbol{x} \in \boldsymbol{g} + \Lambda'$ for some $\boldsymbol{g} \in \mathcal{G}$. This is equivalent to the fact that $\boldsymbol{x} - \boldsymbol{g}$ is a closest vector to $\boldsymbol{t} - \boldsymbol{g}$ in $\Lambda'$. So for all $\boldsymbol{g} \in \mathcal{G}$ we find the closest vector $\boldsymbol{x}_g$ to $\boldsymbol{t} - \boldsymbol{g}$ in $\Lambda'$ in $C(\Lambda')$ operations and we remember the $\boldsymbol{h} = \boldsymbol{g}$ for which which $\boldsymbol{x}_g$ has the minimal distance to their respective $\boldsymbol{t} - \boldsymbol{g}$. Then $\boldsymbol{x}_h + \boldsymbol{h}$ is a closest vector to $\boldsymbol{t}$ in $\Lambda$. Because we are calculating a distance and adding and subtracting vectors of length $d$ for each $\boldsymbol{g} \in \mathcal{G}$ we get the extra $O(d)$ operations on top of $C(\Lambda')$. □

The key idea now is that $A_m^*$ completed with a certain orthogonal lattice can be obtained as a glue-construction from $\mathbb{Z}^{m'}$, with a glue-group of size $m'$. Yet completing both $A_m^*$ and $A_n^*$ and then tensoring two such glue constructions leads to a much larger glue group $G_\otimes$ of size $(m')^{n'} \cdot (n')^{m'}$. Instead, we only complete one of them, and the completed lattice $\overline{A_m^*} \otimes A_n^*$ forms a much smaller glue construction over $\mathbb{Z}^{m'} \otimes A_n^* \simeq \bigoplus_{i=1}^{m'} A_n^*$. This approach therefore allows to exploit the existing algorithms for CVP in $A_n^*$.

**Theorem 1** *It holds that*

$$C(A_m^* \otimes A_n^*) \leq O(nm^{n+1}).$$

*Otherly said, given $\boldsymbol{t} \in \mathrm{span}(A_m^* \otimes A_n^*)$ we can find a closest vector $\boldsymbol{x} \in A_m^* \otimes A_n^*$ to $\boldsymbol{t}$ in $O(n \cdot m^{n+1})$ arithmetic operations on real numbers.*

*Proof* Let $I_{m'}$ be the lattice with basis $\frac{1}{m'}\mathbf{1} \in \frac{1}{m'}\mathbb{Z}^{m'}$. Note that $A_m^*$ and $I_{m'}$ are orthogonal and let $\overline{A_m^*} := A_m^* \perp I_{m'}$. By adding $\frac{1}{m'}\mathbf{1}$ to every row of the generator matrix of $A_m^*$ given in Lemma 3 it is clear that $\mathbb{Z}^{m'}$ is a full rank sublattice of $\overline{A_m^*}$.

Now we consider the lattice $\overline{A_m^*} \otimes A_n^* \supset A_m^* \otimes A_n^*$. We get that:

$$\overline{A_m^*} \otimes A_n^* = (A_m^* \otimes A_n^*) \perp (I_{m'} \otimes A_n^*)$$

such that $C(A_m^* \otimes A_n^*) \leq C(\overline{A_m^*} \otimes A_n^*)$ by Lemma 4. Also note that $\mathbb{Z}^{m'} \otimes A_n^* = \bigoplus_{i=1}^{m'} A_n^*$ is a full rank sublattice of $\overline{A_m^*} \otimes A_n^*$ and furthermore $C(\mathbb{Z}^{m'} \otimes A_n^*) = m' \cdot C(A_n^*) \in O(mn)$ by Lemma 4 and the linear time algorithm for $A_n^*$ [10].

The glue group $G := (\overline{A_m^*} \otimes A_n^*)/(\mathbb{Z}^{m'} \otimes A_n^*)$ consists of $(m')^n$ cosets represented by glue vectors

$$\mathcal{G} = \{\sum_{i=1}^{n} (\boldsymbol{b}_i \otimes \frac{a_j}{m'}\mathbf{1}) : (a_1, \ldots, a_n) \in \{0, \ldots, m\}^n\}$$

where the basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is the basis corresponding to the generator matrix of $A_n^*$ given in Lemma 3. Summarizing we get a time complexity of

$$\begin{aligned} C(A_m^* \otimes A_n^*) \leq C(\overline{A_m^*} \otimes A_n^*) &= C(\bigcup_{\boldsymbol{g} \in \mathcal{G}} \boldsymbol{g} + (\mathbb{Z}^{m'} \otimes A_n^*)) \\ &\leq (m')^n \cdot (O(m'n') + C(\mathbb{Z}^{m'} \otimes A_n^*)) \\ &\leq O(m'n'(m')^n) = O(n(m')^{n'}) \end{aligned}$$

by using Lemmas 4 and 5. □

## 4 Solving the closest vector problem in $A_m \otimes A_n$

*Overview* In this section, we first find a characterization of the Voronoi relevant vectors of $A_m \otimes A_n$ in terms of simple cycles in the complete directed bipartite graph $K_{m',n'}$. Then, we weight the edges of $K_{m',n'}$ depending on a given target $\mathbf{t}$ and current approximation $\mathbf{x}$, in such a way that a simple cycle has negative weight iff the corresponding relevant vector improves the distance to $\mathbf{t}$. Such negative cycles can be found efficiently via the Bellman-Ford algorithm.

Such successive improvements do not directly lead to a polynomial-time algorithm: in a general lattices, each such improvement may be minuscule. Yet, because our lattice is an integer lattice, improvements are guaranteed to be not too small if the target itself is rational with a small common divisor. We finally reach a polynomial time algorithm using successive rational approximation.

### 4.1 Characterizing the Voronoi relevant vectors

Note that the lattice $A_m \otimes A_n$ consists of all $\boldsymbol{x} = (x_{11}, \ldots, x_{1n'}, x_{21}, \ldots, x_{m'n'}) \in \mathbb{Z}^{m' \cdot n'}$ which satisfy the following conditions:

- $\sum_{i=1}^{m'} x_{ij} = 0$ for all $j = 1, \ldots, n'$
- $\sum_{j=1}^{n'} x_{ij} = 0$ for all $i = 1, \ldots, m'$.

Note that those constraints are invariant by negation $\boldsymbol{x} \mapsto -\boldsymbol{x}$, and by permutations of the coordinates of the form $\sigma \times \tau : (i, j) \mapsto (\sigma(i), \tau(j))$ where $\sigma \in \mathfrak{S}_{m'}$ and $\tau \in \mathfrak{S}_{n'}$. Our first lemma allows us to limit our search space for the Voronoi relevant vectors of $A_m \otimes A_n$.

**Lemma 6** *For all Voronoi relevant vectors $\boldsymbol{v} \in RV(A_m \otimes A_n)$ we have that $|v_{ij}| < 2$ for all $i = 1, \ldots, m'$ and $j = 1, \ldots, n'$.*

*Proof* Let $\boldsymbol{v} \in A_m \otimes A_n$ be a Voronoi relevant vector. Assume for contradiction that $|v_{ij}| \geq 2$ for some pair $i, j$. Because of symmetries we can assume without loss of generality that $v_{11} \geq 2$. Let $\boldsymbol{x}^{ij} \in A_m \otimes A_n$ for all $i = 2, \ldots, m'$ and $j = 2, \ldots, n'$ be given by $x_{11} = 1$, $x_{i1} = -1$, $x_{1j} = -1$, $x_{ij} = 1$ and 0 otherwise. Note that this is indeed a lattice point of $A_m \otimes A_n$ and that it is not the same as $\mathbf{0}$ or $\boldsymbol{v}$. Also note that $\langle \boldsymbol{x}^{ij}, \boldsymbol{x}^{ij} \rangle = 4$ for all $i, j$. Then by Definition 3 we get that

$$v_{11} - v_{1j} - v_{i1} + v_{ij} = \langle \boldsymbol{v}, \boldsymbol{x}^{ij} \rangle < \langle \boldsymbol{x}^{ij}, \boldsymbol{x}^{ij} \rangle = 4$$

for all $i = 2, \ldots, m'$ and $j = 2, \ldots, n'$. Also note that because these are all integers we even have that $v_{11} - v_{1j} - v_{i1} + v_{ij} \leq 3$. Summing multiple of these relations for a fixed $i = 2, \ldots, m'$ gives

$$n \cdot v_{11} - n \cdot v_{i1} - \sum_{j=2}^{n'} v_{1j} + \sum_{j=2}^{n'} v_{ij} = \sum_{j=2}^{n'} (v_{11} - v_{1j} - v_{i1} + v_{ij}) \leq 3(n' - 1)$$

but we have that $-\sum_{j=2}^{n'} v_{1j} = v_{11}$ and $\sum_{j=2}^{n'} v_{ij} = v_{i1}$ and thus this gives us

$$n' \cdot v_{11} - n' \cdot v_{i1} \leq 3(n' - 1).$$

As a result of $v_{11} \geq 2$ we now get that $n' \cdot v_{i1} \geq -n' + 3$ and thus $v_{i1} \geq -1 + \frac{3}{n'} > -1$, which again means that $v_{i1} \geq 0$ because it is an integer. So $v_{i1} \geq 0$ for all $i = 2, \ldots, m'$ and $v_{11} \geq 2$. But in that case

$$0 = \sum_{i=1}^{m'} v_{i1} \geq 2 + 0 + \ldots + 0 = 2$$

which gives a contradiction. So $|v_{11}| < 2$. □

Now we have limited our search space for the Voronoi relevant vectors to $X := \{-1,0,1\}^{m'\cdot n'} \cap A_m \otimes A_n$ we can define a subgraph of the complete directed bipartite graph $K_{m',n'}$ for every such element in quite a natural way.

**Definition 9 (Graph $G_{\boldsymbol{x}}$)** Let $\boldsymbol{x} \in \{-1,0,1\}^{m'\cdot n'}$ be given. Let $K_{m',n'}$ be the complete directed bipartite graph with $m'$ nodes $v_1,\dots,v_{m'}$ and $n'$ nodes $w_1,\dots,w_{n'}$. We define the subgraph $G_{\boldsymbol{x}} = (V_{\boldsymbol{x}}, E_{\boldsymbol{x}}) \subset K_{m',n'}$ corresponding to $\boldsymbol{x}$ where $E_{\boldsymbol{x}}$ is defined as

$$E_{\boldsymbol{x}} = \{(v_i, w_j) : x_{ij} = -1\} \cup \{(w_j, v_i) : x_{ij} = 1\}$$

and $V_{\boldsymbol{x}}$ as all nodes with nonzero in- or outdegree.

Now note that for any $\boldsymbol{x} \in \{-1,0,1\}^{m'\cdot n'}$ we have that $\boldsymbol{x} \in X$ iff every node of $G_{\boldsymbol{x}}$ has its indegree equal to its outdegree. So for $\boldsymbol{x} \in X$ we have that $G_{\boldsymbol{x}}$ is a union of disconnected cycles. The following lemma uses this fact to characterize the Voronoi relevant vectors of $A_m \otimes A_n$.

**Theorem 2 (Voronoi relevant vectors of $\boldsymbol{A_m \otimes A_n}$)** *The Voronoi relevant vectors of $A_m \otimes A_n$ are precisely all $\boldsymbol{v} \in X \setminus \{0\}$ such that $G_{\boldsymbol{v}}$ consists of a single simple cycle.*

*Proof* Let $\boldsymbol{v} \in X \setminus \{0\}$ be given. Note that we already have

$$\langle \boldsymbol{v}, \boldsymbol{x} \rangle \le \sum_{i,j} |x_{ij}| \le \sum_{i,j} |x_{ij}|^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle$$

for all $\boldsymbol{x} \in A_m \otimes A_n$ because $\boldsymbol{v} \in X \subset \{-1,0,1\}^{m'n'}$. The second inequality can only be an equality if also $\boldsymbol{x} \in X$. The first inequality then becomes an equality iff $v_{ij}x_{ij} = |x_{ij}|$ for all $i = 1,\dots,m'$ and $j = 1,\dots,n'$. So $x_{ij} = 0$ or $x_{ij} = v_{ij}$. This makes it clear that the only candidates such that $\langle \boldsymbol{v}, \boldsymbol{x} \rangle = \langle \boldsymbol{x}, \boldsymbol{x} \rangle$ are those $\boldsymbol{x} \in X$ such that $G_{\boldsymbol{x}} \subset G_{\boldsymbol{v}}$. By Definition 3 we then get that $\boldsymbol{v} \in RV(A_m \otimes A_n)$ iff $G_{\boldsymbol{0}}$ and $G_{\boldsymbol{v}}$ are the only subgraphs of that form of $G_{\boldsymbol{v}}$.

In fact note that each $G_{\boldsymbol{x}}$ with $\boldsymbol{x} \in X$ consists of a union of disconnected Eulerian graphs and thus a union of disconnected cycles. Furthermore note that every cycle in $G_{\boldsymbol{x}}$ corresponds to a subgraph $H \subset G_{\boldsymbol{x}}$ for which there exists an $\boldsymbol{x}' \in X$ such that $H = G_{\boldsymbol{x}'}$. So $G_{\boldsymbol{v}}$ is a Voronoi relevant vector iff $G_{\boldsymbol{v}}$ contains only the trivial cycles $G_{\boldsymbol{0}}$ and $G_{\boldsymbol{v}}$ and no other cycles. But this is the case iff $G_{\boldsymbol{v}}$ itself consists of a single simple cycle. □

### 4.2 Finding a closer vector in $A_m \otimes A_n$

Now that we have characterized the Voronoi relevant vectors of $A_m \otimes A_n$ we can consider, given a lattice point and a target, the problem of finding an improving Voronoi relevant vector (as in Lemma 1) if one exists. From Theorem 2 we can deduce that $A_m \otimes A_n$ has

$$\sum_{i=2}^{\min\{m',n'\}} \binom{m'}{i}\binom{n'}{i} \cdot i! \cdot (i-1)!$$

many Voronoi relevant vectors and thus checking all of them would not be efficient. Instead, we notice that appropriately weighting the edges of $K_{m',n'}$ allows to evaluate the inner product of an RV vector with a given target, as the weight of the associated simple cycle. An RV vector will be improving iff the weight of the associated cycle is negative.

**Lemma 7** *Let $\boldsymbol{x} \in A_m \otimes A_n$ and let $\boldsymbol{t} \in Span(A_m \otimes A_n)$ be our target. If there exists a Voronoi relevant vector $\boldsymbol{v} \in RV(A_m \otimes A_n)$ such that $\|(\boldsymbol{x}+\boldsymbol{v})-\boldsymbol{t}\| < \|\boldsymbol{x}-\boldsymbol{t}\|$ we can find such a Voronoi relevant vector in $O(\min\{m,n\})mn)$ arithmetic operations on reals. If it doesn't exist this will also be detected by the algorithm.*

*Proof* Let $\boldsymbol{u} := \boldsymbol{x}-\boldsymbol{t}$ be the difference vector of $\boldsymbol{t}$ and $\boldsymbol{x}$. We construct the weighted directed complete bipartite graph $K_{m',n'}(u)$ with weight function $W$ defined as follows for $i = 1,\dots,m'$ and $j = 1,\dots,n'$:

$$W(v_i, w_j) = (u_{ij}-1)^2 - u_{ij}^2 = 1 - 2u_{ij}$$
$$W(w_j, v_i) = (u_{ij}+1)^2 - u_{ij}^2 = 1 + 2u_{ij}.$$

Now consider some $G_{\boldsymbol{v}} \subset K_{m',n'}(u)$ with the same weights for an arbitrary $\boldsymbol{v} \in RV(A_m \otimes A_n)$. Then by construction

$$W(G_{\boldsymbol{v}}) = \sum_{i,j:v_{ij}\neq 0} 1 + 2v_{ij}\cdot u_{ij} = \langle \boldsymbol{v},\boldsymbol{v}\rangle + 2\langle \boldsymbol{v},\boldsymbol{u}\rangle = \|\boldsymbol{u}+\boldsymbol{v}\|^2 - \|\boldsymbol{u}\|^2 .$$

So $\|(\boldsymbol{x}+\boldsymbol{v})-\boldsymbol{t}\| < \|\boldsymbol{x}-\boldsymbol{t}\|$ for a $\boldsymbol{v} \in RV(A_m \otimes A_n)$ iff $G_{\boldsymbol{v}} \subset K_{m',n'}(u)$ has negative weight. By Theorem 2 every simple cycle of length at least 4 in $K_{m',n'}$ corresponds to a Voronoi relevant vector. So the problem of finding a $\boldsymbol{v} \in RV(A_m \otimes A_n)$ such that $\|(\boldsymbol{x}+\boldsymbol{v})-\boldsymbol{t}\| < \|\boldsymbol{x}-\boldsymbol{t}\|$ is equivalent to finding a simple cycle of length at least 4 with negative weight in $K_{m',n'}$. Note that because $W(v_i,w_j) + W(w_j,v_i) = 2 \geq 0$ for all $i = 1,\dots,m'$ and $j = 1,\dots,n'$ there exist no simple cycles of length 2. So we just need to find a simple cycle of negative weight. This can be done by the Bellman-Ford algorithm in $O(C\cdot|E|) = O(\min\{m',n'\}m'n') = O(\min\{m,n\}mn)$ operations, where $C = 2\min\{m',n'\}$ bounds the length of the cycles considered[3]. The construction of the graph itself can easily be done in $O(m+n+mn)$ operations and thus adds nothing to the complexity. The Bellman-Ford algorithm also detects if simple negative weight cycles exist or not [6]. □

### 4.3 Finding a closest vector in $A_m \otimes A_n$

Before we can use Lemma 7 to create a polynomial iterative CVP algorithm for the lattice $A_m \otimes A_n$ we first need a reasonably close starting point and a polynomial bound on the covering radius of $A_m \otimes A_n$. To accomplish this in the following lemma we will use Babai's rounding technique [1] on a sparse and reduced basis of $A_m \otimes A_n$.

**Lemma 8** *For any $\boldsymbol{t} \in \operatorname{span}(A_m \otimes A_n)$ we can find an $\boldsymbol{x} \in A_m \otimes A_n$ such that $\|\boldsymbol{x}-\boldsymbol{t}\| \leq 2\sqrt{m'n'}$ in $O(mn)$ arithmetic operations.*

[3] The algorithm is typically stated with $C = |V|$, the number of vertices

*Proof* Let $\boldsymbol{x}^1, \dots, \boldsymbol{x}^m$ and $\boldsymbol{y}^1, \dots, \boldsymbol{y}^n$ be the basis of $A_m$ and $A_n$ respectively as given in Lemma 3. Then $\{\boldsymbol{b}^{ij} := \boldsymbol{x}^i \otimes \boldsymbol{y}^j : i = 1, \dots, m \text{ and } j = 1, \dots, n\}$ is a basis of $A_m \otimes A_n$.

Suppose that $\boldsymbol{t}' := \boldsymbol{t} = \sum_{i,j} a_{ij}\boldsymbol{b}^{ij}$. Then we have that $a_{11} = t'_{11}$ as all other basis elements have coefficient 0 there. Then let $\boldsymbol{t}' \leftarrow \boldsymbol{t}' - a_{11} \cdot \boldsymbol{b}^{11}$ and consider $a_{12}$. We again have that $a_{12} = t'_{12}$ and after this we set $\boldsymbol{t}' \leftarrow \boldsymbol{t}' - a_{12} \cdot \boldsymbol{b}^{12}$. This equality will be the case for all basis elements if we continue $\boldsymbol{b}^{13}, \dots, \boldsymbol{b}^{1n}, \boldsymbol{b}^{2m}, \dots, \boldsymbol{b}^{mn}$. Note that computing $\boldsymbol{t}' \leftarrow \boldsymbol{t}' - a_{ij}\boldsymbol{b}^{ij}$ can be done in a constant amount of operations as $\boldsymbol{b}^{ij}$ always has only 4 nonzero coefficients. In total calculating all $a_{ij}$ can thus be done in $O(mn)$ operations. So we now have $a_{ij} \in \mathbb{R}$ such that $\boldsymbol{t} = \sum_{i,j} a_{ij}\boldsymbol{b}^{ij}$.

Let $\boldsymbol{x} := \sum_{i,j} \lfloor a_{ij} \rceil \boldsymbol{b}^{ij} \in A_m \otimes A_n$. Again it is clear that $\boldsymbol{x}$ can be calculated in $O(mn)$ operations as every $\boldsymbol{b}^{ij}$ has only 4 nonzero coefficients. Now note that

$$\|\boldsymbol{x} - \boldsymbol{t}\| = \left\| \sum_{i,j} (\lfloor a_{ij} \rceil - a_{ij}) \boldsymbol{b}^{ij} \right\| \leq \sqrt{m'n' \cdot (4 \cdot \frac{1}{2})^2} = 2\sqrt{m'n'}$$

which is the case because the $(kl)$-th coefficient is nonzero in at most 4 basis vectors $\boldsymbol{b}^{ij}$ and combining this with the fact that $|\lfloor a_{ij} \rceil - a_{ij}| \leq \frac{1}{2}$ gives us that the $(kl)$-th coefficient of $\boldsymbol{x} - \boldsymbol{t}$ is bounded in absolute value by $4 \cdot \frac{1}{2} = 2$ for all $k = 1, \dots, m'$ and $l = 1, \dots, n'$. □

We finally have all the ingredients to construct a polynomial iterative CVP algorithm for the lattice $A_m \otimes A_n$. To achieve a bound on the number of iterations we will round our target to a grid. In this way we can give a lower bound on the improvement in squared distance made to our target in each iteration and thus bound the number of iteration from above. This grid will successively be made finer until our target lies in it and a closest vector is found.

**Algorithm 1** A polynomial CVP algorithm for the lattice $A_m \otimes A_n$.

**Input** : $m, n, l \geq 1$ and $t = \sum_{i,j} a_{ij}\boldsymbol{b}^{ij} \in \text{span}(A_m \otimes A_n)$ with $a_{ij} \in 2^{-l}\mathbb{Z}$
**Output**: a closest vector to $\boldsymbol{t}$ in $A_m \otimes A_n$

Find $(a_{kl})_{k,l}$ such that $\boldsymbol{t} = \sum_{k,l} a_{kl}\boldsymbol{b}^{kl}$ ;
$\boldsymbol{a} = \sum_{k,l} \lfloor a_{kl} \rceil \boldsymbol{b}^{kl}$;
**for** $i = 0, \dots, l$ **do**
  `// Outer loop`
  $\boldsymbol{t}_i = \sum_{k,l} 2^{-i} \lfloor 2^i \cdot a_{kl} \rceil \boldsymbol{b}^{kl}$;
  **while** *$K_{m',n'}(\boldsymbol{a} - \boldsymbol{t}_i)$ has a negative cycle $G_{\boldsymbol{v}}$* **do**
    `// Inner loop`
    $\boldsymbol{a} = \boldsymbol{a} + \boldsymbol{v}$;
  $\boldsymbol{x}_i = \boldsymbol{a}$;
**return** $\boldsymbol{x}_l$;

**Theorem 3** *Given a target $\boldsymbol{t} = \sum_{i,j} a_{ij}\boldsymbol{b}^{ij} \in \text{span}(A_m \otimes A_n)$ with all $a_{ij} \in 2^{-l}\mathbb{Z}$ and with $l \geq 1$ we can find a closest vector to $\boldsymbol{t}$ in $A_m \otimes A_n$ in $O(l \cdot (mn)^2 \min\{m, n\})$ arithmetic operations with Algorithm 1.*

*Proof* First note that by Lemmas 1 and 7 it is clear that after each outer loop $\boldsymbol{x}_i$ is a closest vector to $\boldsymbol{t}_i$. Therefore we will focus on the complexity. First let $a_{kl} \in 2^{-l}\mathbb{Z}$ such that $\boldsymbol{t} = \sum_{k,l} a_{kl}\boldsymbol{b}^{kl} \in 2^{-l}\mathbb{Z}^{m'n'}$. Recall that this can be done in time $O(mn)$. Let $\boldsymbol{t}_i := \sum_{k,l} 2^{-i}\lceil 2^i \cdot a_{kl}\rfloor \boldsymbol{b}^{kl}$ for $i = 0, \ldots, l$, so $\boldsymbol{t}_l = \boldsymbol{t}$. Recall that these can also be computed in time $O(mn)$ each as each $\boldsymbol{b}^{kl}$ has only 4 nonzero coefficients. Let $\boldsymbol{x}_i$ be the closest vector to $\boldsymbol{t}_i$ as obtained by the algorithm for $i = 0, \ldots, l$. Let $\boldsymbol{e}_i = \sum_{k,l} a'_{kl}\boldsymbol{b}^{kl} := \boldsymbol{t}_i - \boldsymbol{t}_{i-1}$ and note that $\|\boldsymbol{t}_i - \boldsymbol{t}_{i-1}\| = \|\boldsymbol{e}_i\| \leq 4 \cdot 2^{-i}\sqrt{m'n'}$ as every $|a'_{kl}| \leq 2^{-i}$ and for every coefficient there are at most 4 basis elements that are nonzero there.

Note that if our current target is $\boldsymbol{t}_i$ and our current best approximation is $a \in A_m \otimes A_n$ we will improve in every iteration with at least $2^{-i+1}$ between squared distances if we improve at all as for a relevant vector $\boldsymbol{v} \in RV(A_m \otimes A_n)$ we have

$$\|\boldsymbol{a} + \boldsymbol{v} - \boldsymbol{t}_i\|^2 - \|\boldsymbol{a} - \boldsymbol{t}_i\|^2 = 2\langle \boldsymbol{a} - \boldsymbol{t}_i, \boldsymbol{v}\rangle + \langle \boldsymbol{v}, \boldsymbol{v}\rangle \in 2^{-i+1}\mathbb{Z}^{m'n'}$$

because $\boldsymbol{a}$ and $\boldsymbol{v}$ are integer vectors and $\boldsymbol{t}_i \in 2^{-i}\mathbb{Z}^{m'n'}$.

When searching a closest vector to $\boldsymbol{t}_i$ we start with the approximation $\boldsymbol{x}_{i-1}$. To bound the number of iterations of the inner loop to get to $\boldsymbol{x}_i$ we need the following bound for $i \geq 1$:

$$\begin{aligned}
&\|\boldsymbol{t}_i - \boldsymbol{x}_{i-1}\|^2 - \|\boldsymbol{t}_i - \boldsymbol{x}_i\|^2 \\
&= (\|\boldsymbol{t}_i - \boldsymbol{x}_{i-1}\| + \|\boldsymbol{t}_i - \boldsymbol{x}_i\|)(\|\boldsymbol{t}_i - \boldsymbol{x}_{i-1}\| - \|\boldsymbol{t}_i - \boldsymbol{x}_i\|) \\
&\leq (\|\boldsymbol{t}_{i-1} - \boldsymbol{x}_{i-1}\| + \|\boldsymbol{e}_i\| + \|\boldsymbol{t}_i - \boldsymbol{x}_i\|)\,(\|\boldsymbol{t}_{i-1} - \boldsymbol{x}_{i-1}\| + \|\boldsymbol{e}_i\| - \|\boldsymbol{t}_i - \boldsymbol{x}_i\|)
\end{aligned}$$

Note that by Lemma 8 be have that $\|\boldsymbol{t}_i - \boldsymbol{x}_i\| \leq 2\sqrt{m'n'}$ for all $i \geq 0$. Therefore:

$$\begin{aligned}
&\leq \left(4 + 2^{-i+2}\right)\sqrt{m'n'}\left(2^{-i+2}\sqrt{m'n'} + \operatorname{dist}(\boldsymbol{t}_{i-1}, A_m \otimes A_n) - \operatorname{dist}(\boldsymbol{t}_i, A_m \otimes A_n)\right) \\
&\leq \left(4 + 2^{-i+2}\right)\sqrt{m'n'}\left(2^{-i+2}\sqrt{m'n'} + \|\boldsymbol{t}_{i-1} - \boldsymbol{t}_i\|\right) \\
&\leq \left(4 + 2^{-i+2}\right)\sqrt{m'n'}\left(2^{-i+2}\sqrt{m'n'} + 2^{-i}\sqrt{m'n'}\right) = 10 \cdot 2^{-i+1}\left(1 + 2^{-i}\right)m'n'
\end{aligned}$$

So for fixed $i$ the inner loop starts with $\boldsymbol{a} = \boldsymbol{x}_{i-1}$ and improves this approximation until $\|\boldsymbol{t}_i - \boldsymbol{a}_s\| = \|\boldsymbol{t}_i - \boldsymbol{x}_i\|$. So we get the following

$$\|\boldsymbol{t}_i - \boldsymbol{x}_{i-1}\|^2 = \|\boldsymbol{t}_i - \boldsymbol{a}\|^2 < \|\boldsymbol{t}_i - \boldsymbol{a}_1\|^2 < \ldots < \|\boldsymbol{t}_i - \boldsymbol{a}_s\|^2 = \|\boldsymbol{t}_i - \boldsymbol{x}_i\|^2$$

and because $\|\boldsymbol{t}_i - \boldsymbol{x}_{i-1}\|^2 - \|\boldsymbol{t}_i - \boldsymbol{x}_i\|^2 \leq 10 \cdot 2^{-i+1}\left(1 + 2^{-i}\right)m'n'$ and in every iteration this decreases with at least $2^{-i+1}$ there can be at most $10 \cdot \left(1 + 2^{-i}\right)m'n' + 1$ iterations ($+1$ for the final check) for every $i \geq 1$. So given a closest vector $\boldsymbol{x}_{i-1}$ to $\boldsymbol{t}_{i-1}$ we can find a closest vector $\boldsymbol{x}_i$ to $\boldsymbol{t}_i$ in $O(mn)$ iterations. By Lemma 7 each iteration takes $O(mn\min\{m,n\})$ operations. So in total we need $O((mn)^2\min\{m,n\})$ operations to go from $\boldsymbol{x}_{i-1}$ to $\boldsymbol{x}_i$ for $i \geq 1$. So given $\boldsymbol{x}_0$ we can find $\boldsymbol{x}_l$ in $O(l \cdot (mn)^2\min\{m,n\})$ operations. By Lemma 8 we can find an $\boldsymbol{a} \in A_m \otimes A_n$ such that $\|\boldsymbol{t}_0 - \boldsymbol{a}\|^2 \leq 4m'n'$ and thus

$$\|\boldsymbol{t}_0 - \boldsymbol{a}\|^2 - \|\boldsymbol{t}_0 - \boldsymbol{x}_0\|^2 \leq 4m'n'$$

and as this difference decreases with at least $2^{-0+1} = 2$ every iteration the number of iterations to obtain $\boldsymbol{x}_0$ from the first approximation is also in $O(mn)$ and thus the total number of operations to find $\boldsymbol{x}_0$ is in $O((mn)^2 \min\{m, n\})$. This changes nothing to the total complexity and thus we can find a closest vector to $\boldsymbol{t}_l = \boldsymbol{t}$ in $A_m \otimes A_n$ in $O(l \cdot (mn)^2 \min\{m, n\})$ operations. □

## References


1. Babai, L.: On lovász' lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (1986). DOI 10.1007/BF02579403. URL `http://dx.doi.org/10.1007/BF02579403`
2. Bonifas, N., Dadush, D.: Short paths on the voronoi graph and the closest vector problem with preprocessing. CoRR (2014). URL `http://arxiv.org/abs/1412.6168`
3. Conway, J., Sloane, N.: Fast quantizing and decoding and algorithms for lattice quantizers and codes. IEEE Transactions on Information Theory **28**(2), 227–232 (1982). DOI 10.1109/TIT.1982.1056484
4. Conway, J., Sloane, N.: Voronoi regions of lattices, second moments of polytopes, and quantization. IEEE Transactions on Information Theory **28**(2), 211–226 (1982)
5. Conway, J., Sloane, N.: Sphere Packings, Lattices and Groups. Grundlehren der mathematischen Wissenschaften. Springer New York (1998)
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Third Edition, 3rd edn. The MIT Press (2009)
7. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings, pp. 1–23. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). DOI 10.1007/978-3-642-13190-5_1. URL `http://dx.doi.org/10.1007/978-3-642-13190-5_1`
8. Lyubashevsky, V., Peikert, C., Regev, O.: A Toolkit for Ring-LWE Cryptography, pp. 35–54. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). DOI 10.1007/978-3-642-38348-9_3. URL `http://dx.doi.org/10.1007/978-3-642-38348-9_3`
9. McKilliam, R.G., Clarkson, I.V.L., Quinn, B.G.: An algorithm to compute the nearest point in the lattice $A_n^*$. CoRR (2008). URL `http://arxiv.org/abs/0801.1364`
10. McKilliam, R.G., Clarkson, I.V.L., Smith, W.D., Quinn, B.G.: A linear-time nearest point algorithm for the lattice $A_n^*$. In: Information Theory and Its Applications, 2008. ISITA 2008. International Symposium on, pp. 1–5 (2008). DOI 10.1109/ISITA.2008.4895596
11. Micciancio, D., Voulgaris, P.: A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In: Proceedings of the Forty-second ACM Symposium on Theory of Computing, STOC '10, pp. 351–358. ACM, New York, NY, USA (2010). DOI 10.1145/1806689.1806739. URL `http://doi.acm.org/10.1145/1806689.1806739`
12. Oggier, F., Viterbo, E.: Algebraic number theory and code design for Rayleigh fading channels. Now Publishers Inc (2004)
13. Voronoi, G.: Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs. Journal für die reine und angewandte Mathematik **134**, 198–287 (1908). URL `http://eudml.org/doc/149291`