The Hardness of the Closest Vector Problem with Preprocessing*

Daniele Micciancio†

Abstract

We give a new simple proof of the NP-hardness of the closest vector problem. In addition to being much simpler than all previously known proofs, the new proof yields new interesting results about the complexity of the closest vector problem with preprocessing. This is a variant of the closest vector problem in which the lattice is specified in advance, and can be preprocessed for an arbitrarily long amount of time before the target vector is revealed. We show that there are lattices for which the closest vector problem remains hard, regardless of the amount of preprocessing.

Keywords: Closest Vector Problem, Computational complexity, Cryptography, NP-hardness, Point lattices, Polynomial Hierarchy, Subset Sum.

1 Introduction

A lattice is the set of intersection points of a regular (but not necessarily orthogonal) n-dimensional grid. In the closest vector problem, given a lattice $\mathcal{L}$ and a target vector $\mathbf{y}$ one must find the point in the lattice $\mathcal{L}$ closest to the target $\mathbf{y}$. Lattices are used in coding theory for efficient signaling over band limited channels and vector quantization (see [1] for an overview), and more recently they have been used in cryptography to design encryption functions [2, 3, 4, 5]. In these applications, the lattice $\mathcal{L}$ usually represents the code or encryption function, while the target $\mathbf{y}$ is the received message. In this context the closest vector problem corresponds to the decoding or decryption process. Notice that the lattice $\mathcal{L}$ is usually fixed, and it is known long before transmission occurs. Therefore it makes sense to consider a variant of the closest vector problem in which the lattice is known in advance, and only the target vector $\mathbf{y}$ is specified as input to the problem. Moreover, essentially all known techniques to find (possibly approximate) solutions to the closest vector problem work as follows: (1) first a computationally intensive algorithm is run on the lattice to obtain some information useful for decoding (usually a reduced basis or a trellis); (2) then this information is used to solve the closest vector problem using some simple procedure (some form of rounding [6] for methods based on lattice reduction, or the Viterbi algorithm [7] for trellis based decoding). Trellis based decoding is very efficient, provided that a small trellis for the lattice exists. Unfortunately it has been demonstrated that minimal trellis size can grow exponentially with the dimension of the lattice (see [8, 9, 10]). In this paper we concentrate on methods where the result of preprocessing is always polynomially bounded in the size of the lattice description. Essentially all the preprocessing methods whose output is guaranteed to be small perform some sort of basis reduction (see section 2 for the definition of lattice basis), i.e. given any basis for the lattice, they produce a new basis consisting of short vectors. In certain cases the short basis can be computed in polynomial time, resulting in a polynomial time approximation algorithm for the closest vector problem. This is the case for example in [6] where LLL reduced bases (see [11]) are used, or [12] where block-KZ reduced bases (see [13]) are used, achieving $2^{O(n)}$ and $2^{o(n)}$ approximation factors. In other cases it is not known how to efficiently compute the good basis, but once this good basis is found, a much better approximation to the closest vector can be found in polynomial time. For example [14] shows how to achieve

*An edited version of this paper appears in *IEEE Transactions on Information Theory* **47**(3):1212-1215, Match 2001. This is the author's copy.

†The author is with the Department of Computer Science and Engineering, University of California, San Diego, 9500 Gilman Drive, Mail Code 0114, La Jolla, California 92093-0114. Email: daniele@cs.ucsd.edu

a $O(n^{1.5})$ approximation factor using dual KZ reduced basis[1]. The fastest currently known algorithms to solve the closest vector problem [16, 17] also use KZ reduced bases. However, even if the KZ-reduced basis is given, the running time of the algorithm remains exponential in the dimension of the lattice.

One natural question is whether it is possible to find optimal solutions to the closest vector problem (with preprocessing) in polynomial time, possibly using a different notion of reduced basis, or more generally using some other form of preprocessing with polynomially bounded output. In other words, we are asking if for every lattice $\mathcal{L}$ there exists some polynomial amount of information that makes the closest vector problem in $\mathcal{L}$ easily solvable.

In this paper we give a negative answer to this question, under standard complexity assumptions. In particular, we show that if the closest vector problem with preprocessing can be solved in polynomial time, then NP is contained in P/poly and the polynomial hierarchy collapses (see [18]). Our result is analogous to similar results for the nearest codeword problem [19] and the subset sum problem [20] and is based on a new proof of the NP-hardness of the closest vector problem.

A related result is presented in [17], where it is proved that for recursive cube search (RCS) algorithms (including the algorithm of Kannan [16]), the complexity of decoding *any* sequence of lattices with possible application in communications is exponential in the dimension, regardless of the amount of preprocessing. This result is somehow stronger than the one presented in this paper as it applies to a wide class of lattices, while our result is proved for a specific sequence of lattices. On the other hand, the lower bound in [17] is proved only for a specific class of decoding algorithms (RCS), while our result holds for any (known or yet to be discovered) polynomial time decoding procedure.

The rest of the paper is organized as follows: In section 2 we formally define the closest vector problem with preprocessing and other problems that will be used in this paper. In section 3 we give a new proof of the NP-hardness of the closest vector problem. In section 4 we use the new NP-hardness proof to derive a hardness result for the closest vector problem with preprocessing. Section 5 concludes with some remarks and open problems.

2 Definitions

Given n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$, the *lattice* generated by $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is the set of all integer linear combinations $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \{\sum \mathbf{b}_i x_i \mid x_i \in \mathbb{Z}\}$. The set $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is called a lattice *basis*, and it is usually represented as a matrix $\mathbf{B}$ with the basis vectors $\mathbf{b}_i$ as rows. Using matrix notation the lattice vectors can be represented as $\mathbf{xB}$, where $\mathbf{x}$ is an integer row vector.

The closest vector problem is defined as follows.

Definition 1 *In the closest vector problem (CVP) one is given a lattice basis* $\mathbf{B}$ *and a target vector* $\mathbf{y}$ *and must find a lattice vector* $\mathbf{xB}$ *(*$\mathbf{x} \in \mathbb{Z}^n$*) such that* $\|\mathbf{xB} - \mathbf{y}\|$ *is minimized. In the decisional version of CVP one is also given a real number* t*, and must decide whether there exist an integer vector* $\mathbf{x}$ *such that* $\|\mathbf{xB} - \mathbf{y}\| \leq t$*.*

The decisional and search version of CVP are easily proved equivalent. One direction is obviuos: given an algorithm to find the lattice point $\mathbf{xB}$ closest to $\mathbf{y}$, one can solve the decisional problem simply checking if $\|\mathbf{xB} - \mathbf{y}\| \leq t$. In the other direction, one can use the decision algorithm to recover the coeffients x_i, one at a time, as follows. Say we want to determine x_1. First we determine if x_1 is even or odd by checking if the distance of the target $\mathbf{y}$ from the sublattice $\mathcal{L}(2\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$ is the same as the distance from the original lattice. (Notice that the distance of the target from the lattice can be easily computed using the decision oracle and performing a binary search on t.) The other binary digits of x_1 can be determined repeating the above step after subtracting $\mathbf{b}_1$ from $\mathbf{y}$ if x_1 is odd, and multiplying $\mathbf{b}_1$ by 2. Notice that only a polynomial number of repetitions is needed because the size of the coefficients x_i is polynomial in the size of the input.

[1]The result in [14] is usually presented as a coNP $O(n^{1.5})$ approximation result for the closest vector problem, meaning that the KZ reduced basis constitutes an NP-proof that the target vector is not too close to the lattice. The $O(n^{1.5})$ approximation factor has been subsequently improved to $O(n)$ in [15] using techniques from harmonic analysis.

Once we have found the coefficient x_1 for some closest lattice vector $\mathbf{xB}$, we subtract $x_1\mathbf{b}_1$ from the target $\mathbf{y}$ and go on to the second coefficient, until all x_i have been determined.

We are interested in the following variant of the closest vector problem.

Definition 2 *The* closest vector problem with preprocessing *(CVPP) asks for a function P (the preprocessing function) and an algorithm D (the decoding algorithm) with the following properties:*

- *On input a lattice basis $\mathbf{B}$, $P(\mathbf{B})$ returns a new description L of the lattice $\mathcal{L}(\mathbf{B})$ whose size is polynomially related to the size of $\mathbf{B}$, i.e. there exists a constant c such that size$(L) <$ size$(\mathbf{B})^c$ for all bases $\mathbf{B}$ and $L = P(\mathbf{B})$.*
- *Given L and a target vector $\mathbf{y}$, $D(L, \mathbf{y})$ computes a lattice point $\mathbf{xB}$ closest to $\mathbf{y}$. In the decisional version of CVPP, D is also given a distance t, and $D(L, \mathbf{y}, t)$ decides whether there exists a lattice vector $\mathbf{xB}$ such that $\|\mathbf{xB} - \mathbf{y}\| \leq t$.*

Also for CVPP, the search and decision versions are equivalent: any algorithm to solve the search version also solves the decision version, and the search version can be reduced to the decision version evaluating the preprocessing function P on all lattices $(2^i\mathbf{b}_j, \mathbf{b}_{j+1}, \ldots, \mathbf{b}_n)$ with i bounded by a polynomial in the size of the input basis.

Notice that no complexity assumption is made on the preprocessing function P (other than the restriction on the size of the output). One may think of P as a preprocessing algorithm with unlimited computational resources. However, only the running of D is used to measure the complexity of the decoding process, i.e., we say that CVPP is solvable in polynomial time if there exists a function P and a polynomial time algorithm D such that $D(P(\mathbf{B}), \mathbf{y}, t)$ solves the CVP instance $(\mathbf{B}, \mathbf{y}, t)$.

We show that CVPP cannot be solved in polynomial time by giving a reduction from an NP-hard problem H to CVP with the property that any H instance M is mapped to a CVP instance $(\mathbf{B}, \mathbf{y}, t)$ where $\mathbf{B}$ is a lattice basis that depends only on the size of M. It immediately follows that if CVPP has a polynomial time solution, then the NP-hard problem H is solvable in P/poly, and consequently $NP \subseteq P/\text{poly}$.

In the rest of the paper we will make use of the following notoriously NP-hard problems [21].

Definition 3 *The* subset sum *problem (SS) is the following. Given $n+1$ integers $(a_1, \ldots, a_n, s)$, find a subset of the a_i's (if one exists) that adds up to s, or equivalently, find coefficients $x_i \in \{0, 1\}$ such that $\sum_i a_i x_i = s$. In the decision version of the problem one is given $(a_1, \ldots, a_n, s)$ and must decide if there exist coefficients $x_i \in \{0, 1\}$ such that $\sum_i a_i x_i = s$.*

Definition 4 Exact cover by 3-element sets *(X3C) is the following problem. Given a finite set M and a collection of three element subsets C, decide if there exists a sub-collection $C' \subseteq C$ such that each element of M is contained in exactly one element of C'.*

3 A New NP-hardness Proof for the Closest Vector Problem

The closest vector problem was proved NP-hard for the first time by van Emde Boas [22] in 1981. The original proof is rather complex. Subsequently, Kannan [16] gave a simpler proof by reduction from 3-dimensional matching (3DM, see [21]). In this section we give a new, even simpler, proof of the same NP-hardness result. The new proof will be used to derive a new hardness result for the closest vector problem with preprocessing.

Our proof is by reduction from the subset sum problem, and it is related to (a variant of) the Lagiarias-Odlyzko algorithm [23, 24] (see section 5 for further discussion). Given a subset sum instance $(a_1, \ldots, a_n, s)$ we define a lattice basis $\mathbf{B}$ with one row $\mathbf{b}_i$ for each subset sum coefficient a_i. Then we associate a target vector $\mathbf{y}$ to the sum s. Vectors $\mathbf{b}_i$ and $\mathbf{y}$ are defined as follows:

$$\mathbf{b}_i = [a_i, \overbrace{0, \ldots, 0}^{i-1}, 2, \overbrace{0, \ldots, 0}^{n-i}]$$

$$\mathbf{y} = [s, \overbrace{1, \ldots, 1}^{n}]$$

Notice that the basis $\mathbf{B}$ can be expressed in matrix notation as $\mathbf{B} = [\mathbf{a}|2\mathbf{I}_n]$ where $\mathbf{a}$ is the column vector $[a_1, \ldots, a_n]^T$ and $\mathbf{I}_n$ is the $n \times n$ identity matrix.

We now prove that the reduction is indeed correct. First assume that there exists a solution to the subset sum problem, i.e., there are $x_i \in \{0, 1\}$ such that $\sum_{i=1}^{n} x_i a_i = s$. Then

$$\begin{aligned} \|\mathbf{xB} - \mathbf{y}\|^2 &= \left\| \left[\sum_i a_i x_i - s, 2x_1 - 1, \ldots, 2x_n - 1 \right] \right\|^2 \\ &= \left(\sum_{i=1}^{n} a_i x_i - s \right)^2 + \sum_{i=1}^{n} (2x_i - 1)^2 = n \end{aligned}$$

because $\sum_{i=1}^{n} a_i x_i - s = 0$ and $2x_i - 1 = \pm 1$ for all i. This proves that the distance of $\mathbf{y}$ from $\mathcal{L}(\mathbf{B})$ is at most $\sqrt{n}$ and therefore the CVP instance $(\mathbf{B}, \mathbf{y}, \sqrt{n})$ has solution.

Conversely, assume that the distance of $\mathbf{y}$ from the lattice is at most $\sqrt{n}$ and let $\mathbf{x}$ be an integer vector such that $\|\mathbf{Bx} - \mathbf{y}\| \leq \sqrt{n}$. Notice that also in this case we have

$$\|\mathbf{xB} - \mathbf{y}\|^2 = \left(\sum_{i=1}^{n} a_i x_i - s \right)^2 + \sum_{i=1}^{n} (2x_i - 1)^2$$

and $\sum_{i=1}^{n} (2x_i - 1)^2 \geq n$ because all $2x_i - 1$ are odd integers. Therefore $\|\mathbf{xB} - \mathbf{y}\| \leq \sqrt{n}$ is possible only if $\sum_i a_i x_i - s = 0$ and $(2x_i - 1)^2 = 1$ for all i. This proves that $\sum_{i=1}^{n} a_i x_i = s$ and $x_i \in \{0, 1\}$ for all i, i.e. $\mathbf{x}$ is a solution to the subset sum problem.

4 The Closest Vector Problem with Preprocessing

The proof given in the previous section reduces a subset sum instance $(\mathbf{a}, s)$ to a CVP instance $(\mathbf{B}, \mathbf{y}, t)$ with the property that the lattice basis $\mathbf{B}$ only depends on the subset sum coefficients $\mathbf{a}$, while the target vector $\mathbf{y}$ depends on s.

Therefore we can use the hardness result for the subset sum problem with preprocessing [20] to derive a similar result for the closest vector problem. In particular, [20] proves that there exists a reduction from 3-dimensional matching (3DM) to subset sum (SS), such that 3DM instance M is mapped to a SS instance $(\mathbf{a}, s)$ where the subset sum coefficients $\mathbf{a}$ depend only on the size of M. Combining the result in [20] with our reduction we obtain the following theorem.

Theorem 5 *There exists a reduction from an NP-complete problem H to CVP such that any H instance M is mapped to a CVP instance $(\mathbf{B}, \mathbf{y}, t)$ where the lattice $\mathbf{B}$ depends only on the size of M.*

For completeness we now give a direct reduction from an NP-complete problem (X3C, see definition at the end of section 2) to CVP satisfying the conditions of the theorem. The following reduction essentially combines the ideas from [20] and our reduction from subset sum, but without the complications of using subset sum as an intermediate problem.

Fix some n, let $k = \binom{n}{3}$ and consider a matrix $\mathbf{T} \in \{0, 1\}^{k \times n}$ whose rows are all possible n-dimensional binary vectors containing exactly three ones. We identify the rows of $\mathbf{T}$ with the 3-element subsets of $\{1, \ldots, n\}$, and X3C instances with the corresponding characteristic vectors $\mathbf{m} \in \{0, 1\}^k$. X3C instance $\mathbf{m}$ is mapped to the following CVP instance:

$$\mathbf{B} = [\mathbf{T}|2\mathbf{I}_k] \qquad \mathbf{y} = [\mathbf{1}_n|\mathbf{m}] \qquad t = \|\mathbf{m}\|.$$

It is easy to see that X3C instance $\mathbf{m}$ has a solution if and only if CVP instance $(\mathbf{B}, \mathbf{y}, t)$ has a solution. Moreover, the lattice $\mathbf{B}$ depends only on the dimension k of $\mathbf{m}$.

5 Discussion

The reduction we gave from SS to CVP has obvious connections to the Lagarias-Odlyzko algorithm to solve subset sum (or more precisely the improved version in [24]). The (improved) Lagarias-Odlyzko algorithm works as follows: given a subset sum instance $(\mathbf{a}, s)$, build a lattice

$$\begin{bmatrix} c \cdot a_1 & 2 & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ c \cdot a_n & 0 & 0 & 2 \\ c \cdot s & 1 & \cdots & 1 \end{bmatrix}$$

where c is a sufficiently large constant, and look for a short non-zero vector in the lattice. Notice that if $\mathbf{x}$ is a solution to the subset sum problem, then the lattice has a vector of length $\sqrt{n}$, obtained by multiplying the last row by -1 and the other rows by $\mathbf{x}$. If the short vector found is of the form $\mathbf{xB}$ with $x_{n+1} = -1$ and $x_i \in \{0, 1\}$ for all other $i = 1, \ldots, n$, then $x_1, \ldots, x_n$ is a solution to the subset sum problem.

Notice that this algorithm can be succinctly described as follows:

1. Multiply the subset sum problem by some large constant c to obtain an equivalent instance $(c \cdot a_1, \ldots, c \cdot a_n, c \cdot s)$
2. Reduce $(c \cdot a_1, \ldots, c \cdot a_n, c \cdot s)$ to a CVP instance $(\mathbf{B}, \mathbf{y})$ using the reduction presented in section 3.
3. Solve the closest vector problem $(\mathbf{B}, \mathbf{y}, \sqrt{n})$ using the following heuristic technique: in order to find the lattice vector closest to $\mathbf{y}$, look for a short vector in the lattice generated by $[\mathbf{B}^T | \mathbf{y}^T]^T$. If this short vector is of the form $\mathbf{xB} - \mathbf{y}$, then it yields a vector $(\mathbf{xB}) \in \mathcal{L}(\mathbf{B})$ close to $\mathbf{y}$.

The reason the first column of the basis matrix is multiplied by a large constant c is that it is not known how to solve the shortest vector problem exactly, so in practice an approximation algorithm is used (e.g. LLL). If the first column in the matrix is multiplied by a large constant c, then even approximately shortest vectors must be zero in the first coordinate, and the coefficients $\mathbf{x}$ found by the approximation algorithm must satisfy $\sum a_i x_i = (-x_{n+1})s$. Obviously, there is no guarantee that the x_i are always 0 or 1, or that $x_{n+1} = -1$, and the Lagarias-Odlyzko algorithm succeeds only with high probability when the density of the lattice is sufficiently small.

The closest vector problem (as well as the decoding problem studied in [19]), are known to be NP-hard not only to solve exactly, but also when one seeks only an approximate solution [25, 26]. The arguments presented in this paper for lattices and in [19, 17] for the codes do not seem to extend to approximation versions of the problems. We leave as an open problem to prove that the closest vector problem with preprocessing and the nearest codeword problem with preprocessing are hard to approximate to within the same factors as in [25, 26].

References

[1] John H. Conway and Neil J. A. Sloane, *Sphere Packings, Lattices and Groups*, Springer Verlag, 3rd edition, 1998.

[2] Miklós Ajtai and Cynthia Dwork, "A public-key cryptosystem with worst-case/average-case equivalence," in *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, El Paso, Texas, 4–6 May 1997, pp. 284–293.

[3] Oded Goldreich, Shafi Goldwasser, and Shai Halevi, "Public-key cryptosystems from lattice reduction problems," in *Advances in Cryptology—CRYPTO '97*, Burton S. Kaliski Jr., Ed. 17–21 Aug. 1997, vol. 1294 of *Lecture Notes in Computer Science*, pp. 112–131, Springer-Verlag.

[4] Daniele Micciancio, "Lattice based cryptography: a global improvement," IACR Cryptology ePrint Archive [On-line], Report 1999/005, Mar. 1999, Available at http://eprint.iacr.org. (Formerly *Theory of Cryptography Library* Report 99-05).

[5] Roger Fischlin and Jean-Pierre Seifert, "Tensor-based trapdoors for CVP and their application to public key cryptography," in *7th IMA International Conference "Cryptography and Coding"*. 1999, vol. 1746 of *Lecture Notes in Computer Science*, pp. 244–257, Springer-Verlag.

[6] László Babai, "On Lovasz' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[7] G. David Forney Jr., "The Viterbi algorithm," in *Proc. IEEE*. IEEE, 1973, vol. 61, pp. 268–278.

[8] G. David Forney Jr., "Density/length profiles and trellis complexity of lattices," *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 1753–1772, Nov. 1994.

[9] Vahid Tarokh and Ian F. Blake, "Trellis complexity versus coding gain of lattices I," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1796–1807, Nov. 1996.

[10] Vahid Tarokh and Ian F. Blake, "Trellis complexity versus coding gain of lattices II," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1808–1816, Nov. 1996.

[11] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, pp. 513–534, 1982.

[12] Ravi Kannan, *Annual Reviews of Computer Science*, vol. 2, chapter Algorithmic Geometry of numbers, pp. 231–267, Annual Review Inc., Palo Alto, California, 1987.

[13] Claus-Peter Schnorr, "A hierarchy of polynomial time lattice basis reduction algorithms," *Theoretical Computer Science*, vol. 53, no. 2–3, pp. 201–224, 1987.

[14] Jeffrey C. Lagarias, Hendrik W. Lenstra, Jr., and Claus-Peter Schnorr, "Korkine-Zolotarev bases and successive minima of a lattice and its reciprocal lattice," *Combinatorica*, vol. 10, no. 4, pp. 333–348, 1990.

[15] Wojciech Banaszczyk, "New bounds in some transference theorems in the geometry of numbers," *Mathematische Annalen*, vol. 296, pp. 625–635, 1993.

[16] Ravi Kannan, "Minkowski's convex body theorem and integer programming," *Mathematics of operation research*, vol. 12, no. 3, pp. 415–440, Aug. 1987.

[17] Amir H. Banihashemi and Amir K. Khandani, "On the complexity of decoding lattices using the Korkin-Zolotarev reduced basis," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 162–171, Jan. 1998.

[18] Richard M. Karp and Richard J. Lipton, "Some connections between nonuniform and uniform complexity classes," in *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, Los Angeles, California, 28–30 Apr. 1980, pp. 28–30, Appeared in journal form as: R.M. Karp and R.J. Lipton, Turing machines that take advice, Enseign. Math. **28** (1982) 191–209.

[19] Jehoshua Bruck and Moni Naor, "The hardness of decoding linear codes with preprocessing," *IEEE Transactions on Information Theory*, vol. 36, no. 2, pp. 381–385, Mar. 1990.

[20] Antoine Lobstein, "The hardness of solving subset sum with preprocessing," *IEEE Transactions on Information Theory*, vol. 36, no. 4, pp. 943–946, July 1990.

[21] Michael R. Garey and David S. Johnson, *Computers and Intractability, a guide to the theory of NP-completeness*, A Series of books in the mathematical sciences. W. H. Freeman, San Francisco, 1979.

[22] Peter van Emde Boas, "Another NP-complete problem and the complexity of computing short vectors in a lattice," Tech. Rep. 81-04, Mathematische Instituut, Universiry of Amsterdam, 1981, Available on-line at URL `http://turing.wins.uva.nl/~peter/`.

[23] Jeffrey C. Lagarias and Andrew M. Odlyzko, "Solving low-density subset sum problems," *Journal of the ACM*, vol. 32, no. 1, pp. 229–246, Jan. 1985.

[24] Matthijs J. Coster, Antoine Joux, Brian A. LaMacchia, Andrew M. Odlyzko, Claus-Peter Schnorr, and Jacques Stern, "Improved low-density subset sum algorithms," *Computational Complexity*, vol. 2, no. 2, pp. 111–128, 1992.

[25] Sanjeev Arora, László Babai, Jacques Stern, and Elizabeth Z Sweedyk, "The hardness of approximate optima in lattices, codes, and systems of linear equations," *Journal of Computer and System Sciences*, vol. 54, no. 2, pp. 317–331, Apr. 1997, Preliminary version in FOCS'93.

[26] Irit Dinur, Guy Kindler, and Shmuel Safra, "Approximating CVP to within almost-polynomial factors is NP-hard," in *39th Annual Symposium on Foundations of Computer Science*, Palo Alto, California, 7–10 Nov. 1998, IEEE.