

FINE-GRAINED DETERMINISTIC HARDNESS OF THE SHORTEST VECTOR PROBLEM

MARKUS HITTMEIR

ABSTRACT. Let γ -GapSVP $_p$ be the decision version of the shortest vector problem in the ℓ_p -norm with approximation factor γ , let n be the lattice dimension and $0 < \varepsilon \leq 1$. We prove that the following statements hold for infinitely many values of p .

- $(2 - \varepsilon)$ -GapSVP $_p$ is not in $O(2^{O(p)} \cdot n^{O(1)})$ -time, unless $P = NP$.
- $(2 - \varepsilon)$ -GapSVP $_p$ is not in $O(2^{2^{o(p)}} \cdot 2^{o(n)})$ -time, unless the Strong Exponential Time Hypothesis is false.

The proofs are based on a Karp reduction from a variant of the subset-sum problem that imposes restrictions on vectors orthogonal to the vector of its weights. While more extensive hardness results for the shortest vector problem in all ℓ_p -norms have already been established under randomized reductions, the results in this paper are fully deterministic.

1. INTRODUCTION

The Shortest Vector Problem (SVP) is a fundamental problem in computational mathematics, lattice theory and cryptography. The goal is to find the shortest non-zero vector (with respect to some ℓ_p -norm) that can be written as an integer linear combination of certain vectors over $\mathbb{R}^m$ (the set of these linear combinations is commonly referred to as *lattice*). SVP is the foundation of several public-key cryptosystems ([17]), and known to be NP-hard under randomized reductions ([2], [16]). However, the proof of its NP-hardness for finite ℓ_p -norms under a deterministic reduction is a notorious open problem ([5]). In this paper, we contribute to this problem by proving results on the *fine-grained* deterministic hardness of SVP, which consist in conditional lower bounds for its runtime complexity.

Let us briefly recall the formal definition of lattices and some fundamental notions related to them.

Definition 1.1. Let $n, m \in \mathbb{N}$. A *lattice* $\mathcal{L}$ is the set of all integer linear combinations of n linearly independent vectors $\mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{R}^m$, i.e.

$$\mathcal{L} = \mathcal{L}(\mathbf{b}_1, \dots, \mathbf{b}_n) := \left\{ \sum_{i=1}^n y_i \mathbf{b}_i : y_1, \dots, y_n \in \mathbb{Z} \right\}.$$

The ℓ_p -norm of $\mathbf{x} = (x_1, \dots, x_m) \in \mathbb{R}^m$ is defined as $\|\mathbf{x}\|_p := (\sum_{i=1}^m |x_i|^p)^{1/p}$ for $1 \leq p < \infty$, and as $\|\mathbf{x}\|_\infty := \max_{1 \leq i \leq m} |x_i|$ for $p = \infty$.

2010 *Mathematics Subject Classification.* 11H06, 11Y16.

This work was supported by the Research Council of Norway (grant 357539).

We define $\lambda_{1,p}(\mathcal{L})$ to be the *length of a shortest non-zero vector* in $\mathcal{L}$ in the ℓ_p -norm, i.e.,

$$\lambda_{1,p}(\mathcal{L}) := \min_{\mathbf{v} \in \mathcal{L} \setminus \{0\}} \|\mathbf{v}\|_p.$$

Definition 1.2. Let $p \geq 1$. We define the γ -approximative *decision version of the shortest vector problem* in the ℓ_p -norm, γ -GapSVP $_p$, as follows: Given a lattice $\mathcal{L}$, a distance threshold $\delta > 0$, and an approximation factor $\gamma \geq 1$, decide whether

- $\lambda_{1,p}(\mathcal{L}) \leq \delta$ (YES-instance)
- $\lambda_{1,p}(\mathcal{L}) > \gamma\delta$ (NO-instance)

For $\gamma = 1$, we denote the *exact* decision version of the shortest vector problem by GapSVP $_p$.

Hardness results for γ -GapSVP $_p$ have been studied extensively. The NP-hardness of GapSVP $_\infty$ has first been established in 1981 by van Emde Boas ([19]). Dinur ([8]) proved in 2002 that γ -GapSVP $_\infty$ is NP-hard for approximation factors γ that are almost polynomial in n . Hardness results for finite p started with Ajtai's work in 1998 ([2]), who proved that GapSVP $_2$ is NP-hard under a randomized reduction. Subsequently, the hardness has been extended to constant approximation factors for all finite p ([12], [16]). While these results are still based on randomized reductions (assuming $\text{NP} \not\subseteq \text{RP}$), Micciancio ([16]) also gave a deterministic polynomial-time reduction under the assumption of a number theoretic conjecture that concerns the distribution of square-free smooth numbers. Most hardness results for SVP are based on a reduction from the closest vector problem, using locally dense lattices. For a more detailed overview on results and open problems concerning the complexity of SVP, we refer the reader to Bennett's recent survey ([4]).

Another interesting research area is the *fine-grained* hardness of SVP. The main goal is to provide a lower bound for the runtime complexity of SVP based on an assumption such as the Strong Exponential Time Hypothesis (SETH). This hypothesis states that for every $\varepsilon > 0$ there is a $k \in \mathbb{N}$ such that the k -SAT problem on formulas with n variables is not in $O(2^{(1-\varepsilon)n})$ -time¹. Similar to the NP-hardness results for finite p discussed above, the current results providing such lower bounds for γ -GapSVP $_p$ for finite p are based on the randomized version of this hypothesis. In 2018, it has been proved ([1]) that for $p > p_0 \approx 2.1397$, $p \notin 2\mathbb{Z}$, GapSVP $_p$ is not in $O(2^{C_p n})$ -time for some constant $C_p \in (0, 1)$, unless randomized SETH is false. By using a gap-variant of SETH that states a complexity lower bound for the approximation of the number of satisfiable clauses in k -SAT, this has also been extended to γ -GapSVP $_p$ for some constant $\gamma > 1$ ([6]). In addition, a very recent preprint ([10]) by Hecht and Safra contains two hardness results for GapSVP $_p$ based on reductions that are subexponential-time, but deterministic. The first states that $(\sqrt{2} - o(1))$ -GapSVP $_p$ is not in polynomial-time for every $p > 2$, unless 3-SAT is in $O(2^{O(n^{2/3} \log n)})$ -time. The second statement improves upon the approximation factor γ for sufficiently large p , assuming $\text{NP} \not\subseteq \text{SUBEXP}$.

To summarize, there is a large variety of strong hardness results for GapSVP $_p$. But with few exceptions, the results for finite p rely on randomized reductions. In this paper, we will contribute to the problem of providing stronger hardness results for GapSVP that do not rely on randomization. These are our main results.

¹All cost statements in this paper treat arithmetic/word operations on $O(L)$ -bit words as unit-cost, where L is the maximum bit-length of any input (e.g., lattice vector coordinates).

Theorem 1.3. *Let $0 < \varepsilon \leq 1$. For every $N \in \mathbb{N}$ there is an integer $p > N$ such that $(2 - \varepsilon)$ -GapSVP _{p} is not in $O(2^{O(p)} \cdot n^{O(1)})$ -time, unless $P = NP$.*

Theorem 1.4. *Let $0 < \varepsilon \leq 1$. For every $N \in \mathbb{N}$ there is an integer $p > N$ such that $(2 - \varepsilon)$ -GapSVP _{p} is not in $O(2^{2^{o(p)}} \cdot 2^{o(n)})$ -time, unless SETH is false.*

These statements are based on a *deterministic polynomial-time reduction*. They provide both polynomial and subexponential complexity lower bounds for $(2 - \varepsilon)$ -GapSVP _{p} for infinitely many p , using the established hypotheses $P \neq NP$ and SETH. However, the proofs are not constructive, so they do not yield specific values of p for which the statements hold. We also note that the bound in Theorem 1.3 does not imply NP-hardness of GapSVP _{p} for any *fixed* p , since it does not cover all polynomial expressions of n when p is considered fixed. However, it has immediate consequences for another area of interest, namely the parameterized complexity ([9]) of the shortest vector problem. Consider the *fixed-parameter tractable* (FPT) complexity class consisting of problems that can be decided in running time

$$f(k) \cdot |x|^{O(1)}$$

where $|x|$ is the input size, $k \in \mathbb{N}$ is a parameter and f is an arbitrary (computable) function only depending on k . With regards to the shape of f , we can also define the classes for (sub-)exponential fixed-parameter tractability SUBEPT and EPT. One easily observes that $\text{SUBEPT} \subseteq \text{EPT} \subseteq \text{FPT}$. The following statement follows directly from Theorem 1.3.

Corollary 1.5. *Let $0 < \varepsilon \leq 1$. Then $(2 - \varepsilon)$ -GapSVP _{p} parameterized by $p \in \mathbb{N}$ is not in EPT, unless $P = NP$.*

Our proofs are substantially simpler than the other discussed contributions in the research area. They are based on a Karp reduction from a variant of the subset-sum problem to GapSVP. The standard version of subset-sum considers $a_1x_1 + \dots + a_nx_n = s$, where the a_i (the “weights”) and s (the “target sum”) are given non-negative integers. The problem is to find $(x_1, \dots, x_n) \in \{0, 1\}^n$ such that the equation is satisfied. Our variant adds restrictions for integer vectors that are orthogonal to both the vector of weights $(a_1, \dots, a_n) \in \mathbb{R}^n$ and to the vector $\mathbf{1}_n = (1, 1, \dots, 1) \in \mathbb{R}^n$. Namely, if no such non-zero vector that is short in the ℓ_p -norm exists, then the problem instance is reducible to an instance of GapSVP. We subsequently show that any standard subset-sum problem instance satisfies these restrictions with respect to some ℓ_p -norm for a sufficiently small value of p , allowing us to prove the bounds stated in our main results above.

The remainder of the paper is structured as follows: In Section 2, we will state lemmas and discuss related work, in particular the technique for solving subset-sum problems of low density. In Section 3, we will explain our method and prove Theorem 1.3 and Theorem 1.4. Section 4 provides concluding remarks and directions for future research.

2. PRELIMINARIES

Our main strategy is to exploit and elaborate on a relationship between the shortest vector and the subset-sum problem. We will discuss the core principle of this relationship later in this section. Let us start with some standard and/or easily obtainable results. The following lemma is well-known and can be derived as a special case from Hölder’s inequality.

Lemma 2.1. *Let $1 \leq q \leq p$, $n \in \mathbb{N}$ and $\mathbf{x} \in \mathbb{R}^n$. Then $\|\mathbf{x}\|_q \leq n^{1/q-1/p} \|\mathbf{x}\|_p$.*

As mentioned in the introduction, we will consider a variant of the subset-sum problem that imposes restrictions on the weights. In this context, we now introduce a notion that we will call p -dependency.

Definition 2.2. Let $0 < \alpha < 1$ and $p \geq 1$. We call non-negative integers $a_1, \dots, a_n$ p -dependent with respect to α if there exists $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{Z}^n$ with $\mathbf{x} \neq \mathbf{0}$ such that

$$\sum_{i=1}^n a_i x_i = 0 \wedge \sum_{i=1}^n x_i = 0 \wedge \|\mathbf{x}\|_p \leq \alpha \cdot n^{1/p}.$$

If the value of α is clear from the context, we simply call $a_1, \dots, a_n$ p -dependent.

Example 2.3. Let $n = 25$, $a_1 = 1$, $a_2 = 5$, $a_3 = 9$ and $a_4, \dots, a_n \in \mathbb{N}_0$. For $\mathbf{x} = (-1, 2, -1, 0, \dots, 0) \in \mathbb{Z}^n$, we have $\|\mathbf{x}\|_2 = \sqrt{6} \leq 5/2$, and the other conditions hold too. Hence, $a_1, \dots, a_n$ are 2-dependent with respect to $\alpha = 1/2$.

Example 2.4. Let $p \geq 1$, $n \in \mathbb{N}$ and $N := \lfloor \alpha \cdot n^{1/p} \rfloor + 1$. Then one can prove that $N, N^2, \dots, N^n$ are not p -dependent with respect to α .

Lemma 2.5. *Let $0 < \alpha < 1$, $p \geq 1$, $n, m \in \mathbb{N}$ and $a_1, \dots, a_n, b_1, \dots, b_m$ be non-negative integers. The following hold:*

- (1) $a_1, \dots, a_n$ are p -dependent iff $\lambda a_1, \dots, \lambda a_n$ are p -dependent for every $\lambda \in \mathbb{N}$.
- (2) If $a_1, \dots, a_n$ are p -dependent, then $a_1, \dots, a_n, b_1, \dots, b_m$ are p -dependent.
- (3) If $a_1, \dots, a_n$ are p -dependent, they are also q -dependent for every $q \in [1, p]$.
- (4) If $n < 2\alpha^{-p}$, then $a_1, \dots, a_n$ are not p -dependent.

Proof. One easily observes (1). For (2), let $\mathbf{x} = (x_1, \dots, x_n)$ be a vector that satisfies the conditions of Theorem 2.2 for $a_1, \dots, a_n$. Then $\mathbf{x}' = (x_1, \dots, x_n, 0, \dots, 0)$ in $\mathbb{Z}^{n+m}$ satisfies all conditions of Theorem 2.2 for $a_1, \dots, a_n, b_1, \dots, b_m$, in particular $\|\mathbf{x}'\|_p = \|\mathbf{x}\|_p \leq \alpha \cdot n^{1/p} \leq \alpha \cdot (n+m)^{1/p}$.

For proving (3), assume that there exists $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{Z}^n$ satisfying the condition of Theorem 2.2. Considering the third condition, Theorem 2.1 implies

$$\|\mathbf{x}\|_q \leq n^{1/q-1/p} \|\mathbf{x}\|_p \leq \alpha \cdot n^{1/q},$$

which proves the claim.

Let us now prove (4). Assume that $a_1, \dots, a_n$ are p -dependent. We first note that $\mathbf{x} = (x_1, \dots, x_n)$ must have at least two non-zero entries. The smallest possible combination (leading to the smallest ℓ_p -norm) would be 1 and -1 , which implies $n \geq \alpha^{-p} \|\mathbf{x}\|_p^p = 2\alpha^{-p}$. Clearly, for all other combinations of two or more non-zero entries, n must be even larger. We have thus established that, if $a_1, \dots, a_n$ are p -dependent, then $n \geq 2\alpha^{-p}$. The statement (4) follows. $\square$

Remark 2.6. While we will use the notion of p -dependency exactly as it is stated above, it may be interesting to note the following:

- (i) Define $\mathbf{a} = (a_1, \dots, a_n) \in \mathbb{Z}^n$ and $\mathbf{1}_n = (1, 1, \dots, 1) \in \mathbb{Z}^n$, then the vector $\mathbf{x}$ in Theorem 2.2 is orthogonal to both $\mathbf{a}$ and $\mathbf{1}_n$, i.e.,

$$\mathbf{a} \cdot \mathbf{x} = 0 \text{ and } \mathbf{1}_n \cdot \mathbf{x} = 0.$$

- (ii) Imposing certain restrictions on $a_1, \dots, a_n$ allows to increase the bound in Theorem 2.5 (4). For example, if the $a_1, \dots, a_n$ are distinct, one can prove the same statement for $n < 4\alpha^{-p}$.

Let us move on to explaining an approach for solving subset-sum problems via an oracle for the shortest vector problem. It is well known that the LLL-algorithm ([14]) for lattice reduction can be used to solve a certain class of knapsack and subset-sum problems in polynomial-time ([13], [18]). In particular, consider a subset-sum instance with weights a_i and target sum s , given by the equation

$$(2.1) \quad a_1x_1 + \cdots + a_nx_n = s.$$

The problem is to find $(x_1, \dots, x_n) \in \{0, 1\}^n$ such that (2.1) is satisfied. In the context of lattice reduction techniques, the efficient solvability of this problem depends on the density $d := n / \log_2(\max_i a_i)$. The best of these results has been proved by Coster et al. ([7]), where it is shown that an oracle for finding the shortest vector in a special lattice can be used to solve almost all subset-sum problems with $d < 0.9408$. In practice, this oracle is replaced by an application of the LLL-algorithm or other lattice reduction techniques. The procedure works as follows: For $N := \lfloor \frac{1}{2}\sqrt{n} \rfloor + 1$, we define the lattice $\mathcal{L} \subseteq \mathbb{Z}^{n+1}$ spanned by

$$\begin{aligned} \mathbf{b}_1 &= (1, 0, \dots, 0, 0, Na_1), \\ \mathbf{b}_2 &= (0, 1, \dots, 0, 0, Na_2), \\ &\vdots \\ \mathbf{b}_n &= (0, 0, \dots, 0, 1, Na_n), \\ \mathbf{b}_{n+1} &= \left(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2}, \frac{1}{2}, Ns \right). \end{aligned}$$

If $(x_1, \dots, x_n)$ is a solution to (2.1), we have

$$\mathbf{v} = \sum_{i=1}^n x_i \mathbf{b}_i - \mathbf{b}_{n+1} = (y_1, \dots, y_n, 0) \in \mathcal{L},$$

where $y_i \in \{-\frac{1}{2}, \frac{1}{2}\}$ and, hence, $\|\mathbf{v}\|_2 \leq \frac{1}{2}\sqrt{n}$. Knowing $\mathbf{v}$, it is easy to retrieve a solution to the subset-sum problem. Therefore, the hope of applying the LLL-algorithm is that $\mathbf{v}$ will occur in the reduced basis of $\mathcal{L}$.

In our approach (Theorem 3.9), we will consider a variation of the lattice defined above that also takes the number of non-zero elements in $(x_1, \dots, x_n)$ into account. Using this restriction together with the assumption of p -independency of the weights, we can rigorously prove that $\mathbf{v}$ is the shortest vector in the lattice, regardless of the density of the subset-sum problem.

3. MAIN RESULT

In this section, we will prove Theorem 1.3 and Theorem 1.4. We start by formally defining the decision version of the subset-sum problem, and consider some hardness results from the literature.

Definition 3.1. We define the *subset-sum problem*, SS , as follows: Given $s \in \mathbb{N}$ and non-negative integers $a_1, \dots, a_n$, decide whether

- $\exists (x_1, \dots, x_n) \in \{0, 1\}^n : \sum_{i=1}^n a_i x_i = s$ (YES-instance)
- $\nexists (x_1, \dots, x_n) \in \{0, 1\}^n : \sum_{i=1}^n a_i x_i = s$ (NO-instance)

For $\text{SS}(a_1, \dots, a_n, s)$, we will call witnesses $(x_1, \dots, x_n)$ for YES-instances *solutions* of $\text{SS}(a_1, \dots, a_n, s)$.

Lemma 3.2. *SS is NP-hard.*

Proof. Subset-sum is the eighteenth problem in Karp’s list of NP-complete problems from 1972 (see [11, p. 95], under “Knapsack”). The proof is by reduction from the partition problem. The only difference is that Karp’s original definition assumes all weights to be positive, whereas we also allow the weights to be 0. However, it is easy to see that this generalization preserves NP-hardness. $\square$

Lemma 3.3. *Assuming SETH, for any $\varepsilon > 0$ there exists $\delta > 0$ such that SS is not in time $O(s^{1-\varepsilon}2^{\delta n})$.*

Proof. This is Theorem 1.1 in [3]. The proof establishes a reduction from k -SAT to subset-sum instances that provides a much tighter lower bound than previous reductions, such as those from standard NP-hardness proofs. $\square$

Consider subset-sum problem instances $\text{SS}(a_1, \dots, a_n, s)$ as defined in Theorem 3.1. We start by showing that we can impose a restriction on the number of weights without losing the NP-hardness.

Definition 3.4. We define the $1/2$ -full subset-sum problem, $\text{SS}_{1/2}$, as follows: Given $s \in \mathbb{N}$ and non-negative integers $a_1, \dots, a_n$ with n being even, decide whether

- $\exists (x_1, \dots, x_n) \in \{0, 1\}^n : \sum_{i=1}^n a_i x_i = s \wedge |\{i : x_i = 1\}| = \frac{n}{2}$ (YES-instance)
- $\nexists (x_1, \dots, x_n) \in \{0, 1\}^n : \sum_{i=1}^n a_i x_i = s \wedge |\{i : x_i = 1\}| = \frac{n}{2}$ (NO-instance)

Lemma 3.5. *There is a deterministic polynomial-time reduction from SS to $\text{SS}_{1/2}$.*

Proof. Let $\text{SS}(a_1, \dots, a_n, s)$ be any subset-sum problem instance. Let

$$\text{SS}_{1/2}(a_1, \dots, a_n, 0, \dots, 0, s),$$

be an instance of $\text{SS}_{1/2}$ where n weights of 0s are added in addition to the n weights $a_1, \dots, a_n$ of the original problem. One easily observes that $\text{SS}(a_1, \dots, a_n, s)$ has a solution if and only if $\text{SS}_{1/2}(a_1, \dots, a_n, 0, \dots, 0, s)$ has a solution. $\square$

Definition 3.6. Let $c \in (0, 1)$ be a constant. We define the c -full subset-sum problem, SS_c , as follows: Given $s \in \mathbb{N}$ and non-negative integers $a_1, \dots, a_n$ such that $cn \in \mathbb{N}$, decide whether

- $\exists (x_1, \dots, x_n) \in \{0, 1\}^n : \sum_{i=1}^n a_i x_i = s \wedge |\{i : x_i = 1\}| = cn$ (YES-instance)
- $\nexists (x_1, \dots, x_n) \in \{0, 1\}^n : \sum_{i=1}^n a_i x_i = s \wedge |\{i : x_i = 1\}| = cn$ (NO-instance)

By $\text{SS}_c^{p, \alpha}$, we denote the c -full subset-sum problem that is restricted to inputs $a_1, \dots, a_n$ that are not p -dependent with respect to α (see Theorem 2.2).

Lemma 3.7. *Let $m \in \mathbb{N}$. For both $c = (m + 1/2)/(m + 1)$ and $c = 1/(2(m + 1))$, there is a deterministic polynomial-time reduction from $\text{SS}_{1/2}$ to SS_c .*

Proof. Let $\text{SS}_{1/2}(a_1, \dots, a_n, s)$ be any $\text{SS}_{1/2}$ instance. We start by considering the case $c := (m + 1/2)/(m + 1)$. Let $b_1, \dots, b_{mn}$ be any integers greater than $\sum_{i=1}^n a_i$. Denote $\beta := \sum_{i=1}^{mn} b_i$ and consider the target sum $s' := s + \beta$ together with the $mn + n$ integers $a_1, \dots, a_n, b_1, \dots, b_{mn}$. Note that

$$c(mn + n) = mn + n/2 \in \mathbb{N}.$$

Clearly, any solution of $\text{SS}_c(a_1, \dots, a_n, b_1, \dots, b_{mn}, s')$ needs to contain all the b_i and exactly $n/2$ weights among the a_i . As a consequence, one easily observes that it has a solution if and only if $\text{SS}_{1/2}(a_1, \dots, a_n, s)$ has a solution.

Let us now prove the case $c := 1/(2(m+1))$. We consider the same $mn+n$ integers $a_1, \dots, a_n, b_1, \dots, b_{mn}$, as above, but change the target to $s' := s$. Note that $c(mn+n) = n/2 \in \mathbb{N}$. Now any solution of $\text{SS}_c(a_1, \dots, a_n, b_1, \dots, b_{mn}, s')$ must omit the b_i and contain exactly $n/2$ weights among the a_i . Again we obtain that $\text{SS}_c(a_1, \dots, a_n, b_1, \dots, b_{mn}, s')$ has a solution if and only if $\text{SS}_{1/2}(a_1, \dots, a_n, s)$ has a solution. $\square$

Lemma 3.8. *Let $m \in \mathbb{N}$. For both $c = (m+1/2)/(m+1)$ and $c = 1/(2(m+1))$, the following holds.*

- (1) SS_c is NP-hard.
- (2) Assuming SETH, there exists $\delta > 0$ such that SS_c is not in time $O(2^{\delta n})$.

Proof. The first statement follows from the reductions in Theorem 3.5 and Theorem 3.7 and from Theorem 3.2.

For proving the second statement, assume to the contrary that, for all $\delta > 0$, SS_c can be solved in time $O(2^{\delta n})$. Fix some $0 < \varepsilon \leq 1$, let $\delta > 0$ be arbitrary, and let $\text{SS}(a_1, \dots, a_l, s)$ be any subset-sum problem instance. In the proofs of Theorem 3.5 and Theorem 3.7, one easily observes that this SS instance can be reduced to a SS_c instance with $n = Ml$ weights, where $M := 2(m+1)$ is a fixed constant. Setting $\delta_0 := \delta/M$, our assumption allows us to solve this SS_c instance and, thus, our initial SS instance, in time

$$O(2^{\delta_0 n}) = O(2^{\delta l}) \subseteq O(s^{1-\varepsilon} 2^{\delta l}).$$

Since $\delta > 0$ was arbitrary, we have shown that there exists $\varepsilon > 0$ such that, for all $\delta > 0$, SS is in time $O(s^{1-\varepsilon} 2^{\delta l})$. From Theorem 3.3 it follows that SETH is false, which we wanted to show. $\square$

The most crucial step towards our main results is the following lemma, which concerns a reduction from $\text{SS}_c^{p,\alpha}$ instances to GapSVP instances.

Lemma 3.9. *Let $0 < \varepsilon \leq 1$ and $m \in \mathbb{N}$ such that*

$$(3.1) \quad c := \frac{m+1/2}{m+1} > 1 - \frac{\varepsilon}{4} \quad \text{and} \quad \alpha := 1 - \frac{\varepsilon}{2}.$$

Let $\text{SS}_c^{p,\alpha}(a_1, \dots, a_n, s)$ be an instance of $\text{SS}_c^{p,\alpha}$, where $p \geq 1$ and $r := cn \in \mathbb{N}$. Define $N := \lfloor \alpha \cdot n^{1/p} \rfloor + 1$ and consider the lattice $\mathcal{L}_r \subseteq \mathbb{Z}^{n+2}$ spanned by the vectors

$$\begin{aligned} \mathbf{b}_1 &= (1, 0, \dots, 0, 0, N, Na_1), \\ \mathbf{b}_2 &= (0, 1, \dots, 0, 0, N, Na_2), \\ &\vdots \\ \mathbf{b}_n &= (0, 0, \dots, 0, 1, N, Na_n), \\ \mathbf{b}_{n+1} &= \left(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2}, Nr, Ns \right). \end{aligned}$$

If either $\varepsilon = 1$ or $n < (3^p - 1)/((2 - \varepsilon)^p - 1)$ for $\varepsilon < 1$, the following hold:

- (1) *If $(x_1, \dots, x_n)$ solves $\text{SS}_c^{p,\alpha}(a_1, \dots, a_n, s)$, then $(x_1 - \frac{1}{2}, \dots, x_n - \frac{1}{2}, 0, 0)$ is a shortest vector of $\mathcal{L}_r$, and it holds that $\lambda_{1,p}(\mathcal{L}_r) = n^{1/p}/2$.*
- (2) *If $\|\mathbf{v}\|_p \leq (2 - \varepsilon) \cdot n^{1/p}/2$ for some vector $\mathbf{v} = (v_1, \dots, v_{n+2}) \in \mathcal{L}_r$, then either $(v_1 + \frac{1}{2}, v_2 + \frac{1}{2}, \dots, v_n + \frac{1}{2})$ or $(-v_1 - \frac{1}{2}, -v_2 - \frac{1}{2}, \dots, -v_n - \frac{1}{2})$ is a solution of $\text{SS}_c^{p,\alpha}(a_1, \dots, a_n, s)$.*

Proof. By definition, any solution of $\text{SS}_c^{p,\alpha}(a_1, \dots, a_n, s)$ has exactly r non-zero entries in the corresponding vector $(x_1, \dots, x_n)$. One easily observes that $\mathcal{L}_r$ contains the vector $\mathbf{s} = (s_1, \dots, s_n, 0, 0)$, where $s_i = x_i - 1/2$. Note that $\|\mathbf{s}\|_p^p = n/2^p$. Let $\mathcal{S}$ be the set that, for all solutions of $\text{SS}_c^{p,\alpha}(a_1, \dots, a_n, s)$, contains $\mathbf{s}$ and $-\mathbf{s}$. Our goal is to show that $\mathcal{S}$ is the complete set of $(2 - \varepsilon)$ -approximate shortest vectors in $\mathcal{L}_r$. Consider all vectors $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_{n+2})$ which satisfy the three conditions

$$\begin{aligned} \|\hat{\mathbf{x}}\|_p &\leq (2 - \varepsilon) \cdot n^{1/p}/2 = \alpha \cdot n^{1/p}, \\ \hat{\mathbf{x}} &\in \mathcal{L}_r, \\ \hat{\mathbf{x}} &\notin \mathcal{S} \cup \{\mathbf{0}\}. \end{aligned}$$

In order to prove both claims (1) and (2), it is enough to show that no such vector exists. Assume to the contrary that it does. Then $N > \alpha \cdot n^{1/p}$ implies $\hat{x}_{n+1} = 0$ and $\hat{x}_{n+2} = 0$. Writing $\hat{\mathbf{x}} = \sum_{i=1}^n y_i \mathbf{b}_i + y \mathbf{b}_{n+1}$ establishes the following equations.

$$(3.2) \quad \hat{x}_i = y_i + y/2 \text{ for } 1 \leq i \leq n,$$

$$(3.3) \quad 0 = \hat{x}_{n+1} = N \left(yr + \sum_{i=1}^n y_i \right),$$

$$(3.4) \quad 0 = \hat{x}_{n+2} = N \left(\sum_{i=1}^n a_i y_i + ys \right).$$

We now prove that $|y| < 2$. We start by showing that, for all $\hat{\mathbf{x}}$ satisfying our conditions,

$$(3.5) \quad \left| \sum_{i=1}^n y_i \right| \leq \frac{n(|y| + 2 - \varepsilon)}{2}.$$

As a consequence of (3.2), we have $|y_i| = |\hat{x}_i - y/2| \leq |\hat{x}_i| + |y/2|$, from which it follows that

$$\left| \sum_{i=1}^n y_i \right| \leq \frac{n|y|}{2} + \sum_{i=1}^n |\hat{x}_i|.$$

Since $\|\hat{\mathbf{x}}\|_p \leq \alpha \cdot n^{1/p}$, Theorem 2.1 implies that

$$\sum_{i=1}^n |\hat{x}_i| = \|\hat{\mathbf{x}}\|_1 \leq n^{1-1/p} \|\hat{\mathbf{x}}\|_p \leq \alpha \cdot n = (2 - \varepsilon) \cdot n/2.$$

We conclude that (3.5) holds. Together with (3.3), we obtain

$$r = \frac{1}{|y|} \left| \sum_{i=1}^n y_i \right| \leq \frac{n}{2} \left(1 + \frac{2 - \varepsilon}{|y|} \right).$$

For $|y| \geq 2$, using (3.1) in this inequality yields $r \leq (1 - \varepsilon/4) \cdot n < cn = r$, a contradiction. We conclude that $|y|$ has to be smaller than 2.

We are hence left with the cases $y = 0, 1, -1$. Let $y = -1$ and note that $|\hat{x}_i| = |y_i - 1/2|$ due to (3.2). Assume first that

$$(3.6) \quad y_i = 0 \quad \vee \quad y_i = 1$$

holds for all $i \leq n$. We either have $\hat{x}_i = 0 - 1/2 = -1/2$ or $\hat{x}_i = 1 - 1/2 = 1/2$. In any case, $|\hat{x}_i|^p = 1/2^p$ for all i , and $\|\hat{\mathbf{x}}\|_p = n^{1/p}/2$. However, (3.4) yields a solution to $\text{SS}_c^{p,\alpha}$. Hence, the third condition $\hat{\mathbf{x}} \notin \mathcal{S} \cup \{\mathbf{0}\}$ is not satisfied.

Assume now that there is at least one i for which (3.6) is not true. It follows that $|\hat{x}_i| = |y_i - 1/2| \geq 3/2$ and, hence,

$$\|\hat{\mathbf{x}}\|_p \geq ((n-1)/2^p + (3/2)^p)^{1/p}.$$

We apply our assumptions to prove that $((n-1)/2^p + (3/2)^p)^{1/p} > (2-\varepsilon) \cdot n^{1/p}/2$. If $\varepsilon = 1$, it is easy to see that this is true. We may hence focus on $\varepsilon < 1$, where we assumed that $n < (3^p - 1)/((2-\varepsilon)^p - 1)$. This is equivalent to

$$\begin{aligned} 3^p &> (2-\varepsilon)^p \cdot n - (n-1) \\ \Leftrightarrow (3/2)^p + (n-1)/2^p &> (2-\varepsilon)^p \cdot n/2^p, \\ \Leftrightarrow ((n-1)/2^p + (3/2)^p)^{1/p} &> (2-\varepsilon) \cdot n^{1/p}/2. \end{aligned}$$

It follows that $\|\hat{\mathbf{x}}\|_p > (2-\varepsilon) \cdot n^{1/p}/2 = \alpha \cdot n^{1/p}$. Hence, for $y = -1$, no $\hat{\mathbf{x}}$ exists that satisfies the conditions. With very similar arguments, the same can be proved for $y = 1$.

Let us now deal with the final remaining case, $y = 0$. From (3.3) we may deduce that $\sum_{i=1}^n y_i = 0$. Define $\mathbf{y} = (y_1, y_2, \dots, y_n) \in \mathbb{Z}^n$, then (3.2) implies that

$$\|\mathbf{y}\|_p = \|\hat{\mathbf{x}}\|_p \leq (2-\varepsilon) \cdot n^{1/p}/2 = \alpha \cdot n^{1/p}.$$

Since $\hat{\mathbf{x}} \neq \mathbf{0}$ we also have $\mathbf{y} \neq \mathbf{0}$. Together with (3.4), this entails that $a_1, \dots, a_n$ are p -dependent with respect to α , a contradiction. So for $y = 0$, there is no $\hat{\mathbf{x}}$ satisfying the conditions. This finishes the proof. $\square$

Corollary 3.10. *Let $p \geq 1$, $c = 5/6$ and $\alpha = 1/2$. There exists a deterministic polynomial-time reduction from $\text{SS}_c^{p,\alpha}$ to GapSVP_p .*

Proof. Let $\text{SS}_c^{p,\alpha}(a_1, \dots, a_n, s)$ be any instance of $\text{SS}_c^{p,\alpha}$, and apply Theorem 3.9 with $\varepsilon = 1$ and $m = 2$. Note that the inequality in (3.1) is satisfied. We compute N and, for $r = 5n/6$, construct the lattice $\mathcal{L}_r$ spanned by the $\mathbf{b}_i$ in polynomial-time. Then the statements (1) and (2) guarantee that, for the distance threshold $\delta := n^{1/p}/2$, $\text{SS}_c^{p,\alpha}(a_1, \dots, a_n, s)$ is a YES-instance if and only if $\text{GapSVP}_p(\mathcal{L}_r, \delta)$ is a YES-instance. $\square$

Finally, we may prove our two main results on the hardness of GapSVP with approximation factors of the shape $2 - \varepsilon$.

Proof of Theorem 1.3. Let $0 < \varepsilon \leq 1$ be arbitrary. Set $\alpha := 1 - \varepsilon/2$ and, for a sufficiently large $m \in \mathbb{N}$, set $c := (m + 1/2)/(m + 1) > 1 - \varepsilon/4$. Assume to the contrary that there is $N_0 \in \mathbb{N}$ such that for all $p > N_0$ it holds that $(2-\varepsilon)\text{-GapSVP}_p$ can be solved in time $O(2^{O(p)} \cdot n^{O(1)})$. We show that this implies $\text{SS}_c \in \text{P}$.

Let $\text{SS}_c(a_1, \dots, a_n, s)$ be any instance of SS_c . If $n \leq 2^C$ for $C = \log(1/\alpha) \cdot N_0 + 1$, it can be solved in $O(n \cdot 2^n) = O(2^{2^C + C}) \subseteq O(1)$ arithmetic operations. We may hence focus on the case $n > 2^C$. For $p > (\log_2 n - 1)/\log_2(1/\alpha)$, Theorem 2.5 (4) implies that $\text{SS}_c(a_1, \dots, a_n, s) = \text{SS}_c^{p,\alpha}(a_1, \dots, a_n, s)$. We compute the slightly larger value $\hat{p} := \lceil \log_2(n+1)/\log_2(1/\alpha) \rceil$, and apply Theorem 3.9 with our values for ε , m and $\hat{p}$. We compute N and, for $r = cn$, construct the lattice $\mathcal{L}_r$ spanned by the $\mathbf{b}_i$ in polynomial-time. We have

$$\hat{p} \geq \log_2(n+1)/\log_2(2/(2-\varepsilon)) > \log_2(n+1)/\log_2(3/(2-\varepsilon)).$$

Hence, for $\varepsilon < 1$, it follows that

$$n < \frac{3^{\hat{p}}}{(2-\varepsilon)^{\hat{p}}} - 1 < \frac{3^{\hat{p}}}{(2-\varepsilon)^{\hat{p}}} - \frac{1}{(2-\varepsilon)^{\hat{p}}} < \frac{3^{\hat{p}} - 1}{(2-\varepsilon)^{\hat{p}} - 1}.$$

Consequently, all requirements for Theorem 3.9 are satisfied. The statements (1) and (2) guarantee that, for the distance threshold $\delta := n^{1/\hat{p}}/2$, $\text{SS}_c^{\hat{p},\alpha}(a_1, \dots, a_n, s)$ is a YES-instance if and only if $(2-\varepsilon)\text{-GapSVP}_{\hat{p}}(\mathcal{L}_r, \delta)$ is a YES-instance. But since $n > 2^C$ implies $\hat{p} > (\log_2 n - 1)/\log_2(1/\alpha) > N_0$, it follows from our assumption that we can solve $(2-\varepsilon)\text{-GapSVP}_{\hat{p}}$ instances in

$$O\left(2^{O(\hat{p})} \cdot n^{O(1)}\right) = O\left(2^{O(\log n)} \cdot n^{O(1)}\right) \subseteq O(n^{O(1)})$$

arithmetic operations. As a consequence of Theorem 3.9, the same is true for the considered SS_c instance. Since the instance was arbitrary, we obtain a polynomial-time algorithm for SS_c . From Theorem 3.8 (1) it follows that $\text{P} = \text{NP}$, which we wanted to show. $\square$

Proof of Theorem 1.4. The proof is similar to that of Theorem 1.3. Let $0 < \varepsilon \leq 1$, $\alpha := 1 - \varepsilon/2$ and c as above. Assume to the contrary that there is $N_0 \in \mathbb{N}$ such that for all $p > N_0$ it holds that $(2-\varepsilon)\text{-GapSVP}_p$ can be solved in time $O\left(2^{2^{o(p)}} \cdot 2^{o(n)}\right)$. We assume $n > 2^C$ for $C = \log(1/\alpha) \cdot N_0 + 1$. Theorem 2.5 (4) implies $\text{SS}_c(a_1, \dots, a_n, s) = \text{SS}_c^{\hat{p},\alpha}(a_1, \dots, a_n, s)$ for $\hat{p} := \lceil (\log_2(n+1))/\log_2(1/\alpha) \rceil$. We may apply Theorem 3.9 to translate this $\text{SS}_c^{\hat{p},\alpha}$ instance into a $(2-\varepsilon)\text{-GapSVP}_{\hat{p}}$ instance. Since $\hat{p} > N_0$, our assumption implies that the latter can be solved in time

$$O\left(2^{2^{o(\hat{p})}} \cdot 2^{o(n)}\right) = O\left(2^{2^{o(\log n)}} \cdot 2^{o(n)}\right) \subseteq O(2^{o(n)}).$$

Therefore, the considered SS_c instance can be solved in time $O(2^{o(n)})$. Since the instance was arbitrary, we obtain an algorithm for SS_c that runs in time $O(2^{o(n)})$. Clearly, for all $\delta > 0$ we have $O(2^{o(n)}) \subseteq O(2^{\delta n})$. Using Theorem 3.8 (2), it follows that SETH is violated, which we wanted to show. $\square$

4. DISCUSSION

Let us conclude by discussing some properties of the reduction proved in this paper, as well as possibilities to improve the current results, and other directions to explore in future research.

Remark 4.1. (Scalability of the reduction)

For the sake of convenience, we chose $\hat{p}$ slightly larger than we needed to in the proofs of Theorem 1.3 and Theorem 1.4. Considering $\varepsilon = 1$ and $c = 5/6$ like in Theorem 3.10, one observes that any instance $\text{SS}_{5/6}(a_1, \dots, a_n, s)$ can be reduced to an instance of GapSVP_p for $p > \log_2 n - 1$. Following back the reductions (Theorem 3.5 and Theorem 3.7) to an original $\text{SS}(a'_1, \dots, a'_m, s')$ instance, we see that $\text{SS}_{5/6}(a_1, \dots, a_n, s)$ has six times as many weights as the original instance, i.e., $n = 6m$. It follows that any subset-sum instance with m weights can be reduced to an instance of $\text{GapSVP}_{\hat{p}}$ for $\hat{p} := \lceil \log_2(6m) - 1 \rceil$. For example, all subset-sum problem instances with $m = 10000$ weights (or less) can be reduced to instances of GapSVP_{15} , and instances with $m = 10^{82}$ weights (or less) can be reduced to instances of GapSVP_{274} .

Remark 4.2. (Size of the next shortest vectors)

For even y (in particular, for the case $y = 0$), the next shortest vector must contain at least two entries of 1 and -1 in the components $\hat{x}_i$, leading to a total length of at least $2^{1/p}$. So for increasing p , we have $\lim_{p \rightarrow \infty} n^{1/p}/2 = 1/2$ for the shortest vector, but $\lim_{p \rightarrow \infty} 2^{1/p} = 1$ for the next shortest vector. For odd y (in particular for $y = \pm 1$), the next shortest vector must contain at least one entry $\pm 3/2$ in addition to the other entries of shape $\pm 1/2$, so for increasing p , we have

$$\lim_{p \rightarrow \infty} ((n-1)/2^p + (3/2)^p)^{1/p} = 3/2.$$

It thus seems like the approximation factor $\gamma = 2 - \varepsilon$ is optimal for our technique, at least in its currently applied form.

The idea of using $1/2$ in the components of $\mathbf{b}_{n+1}$ has first been introduced in Coster et al.'s improvement for solving low-density subset-sum instances ([7]). As it increases the gap between the shortest vector and the next shortest vectors, it is also a crucial element of our approach. The resulting factor of $1/2$ in the length of the shortest vector allows us to deduce an upper bound for n in Theorem 2.5 (4) that grows with p and can be used to prove our main results. However, since we are working with subset-sum instances SS_c with c close to 1, it is natural to consider $\mathbf{b}_{n+1} = (c, c, \dots, c, Nr, Ns)$ instead, which leads to a shortest vector of size

$$n^{1/p}(c(1-c)^p + c^p(1-c))^{1/p}.$$

For fixed p and $c \rightarrow 1$, this is indeed shorter than $n^{1/p}/2$ (although not short enough to prove NP-hardness of GapSVP_p directly). For growing p and fixed $c < 1$, however, this advantage vanishes. Unfortunately, it thus appears like our main results cannot be improved by using this replacement of $\mathbf{b}_{n+1}$.

Remark 4.3. (NP-hardness of GapSVP_p for finite p)

Besides improving the construction of the lattice, Theorem 3.10 gives rise to another direction for future research. It implies that we could show NP-hardness of GapSVP_p by proving NP-hardness of $\text{SS}_{5/6}^{p,1/2}$. We plan on investigating the potential of this approach and will study the notion of p -dependency in this context.

Open Question 4.4. Is $\text{SS}_{5/6}^{p,1/2}$ NP-hard for any $p \geq 1$?

REFERENCES

- [1] D. Aggarwal and N. Stephens-Davidowitz, *(Gap/S)ETH hardness of SVP*, In: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC), 228–238, 2018.
- [2] M. Ajtai, *The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract)*, In: Proceedings of the thirtieth annual ACM symposium on Theory of Computing, 10–19, 1998.
- [3] A. Abboud, K. Bringmann, D. Hermelin, and D. Shabtay, *SETH-based Lower Bounds for Subset Sum and Bicriteria Path*, ACM Trans. Algorithms, 18 (1), Article No. 6: 1–22, 2022.
- [4] H. Bennett, *The Complexity of the Shortest Vector Problem*, ACM SIGACT News, 54(1): 37–61, 2023.
- [5] H. Bennett, C. Peikert, *Hardness of the (approximate) shortest vector problem: A simple proof via Reed-Solomon codes*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 275: 37:1–37:20, 2023.
- [6] H. Bennett, C. Peikert, and Y. Tang, *Improved Hardness of BDD and SVP Under Gap-(S)ETH*, In: Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS), Vol. 215, Article No. 19, 2022.

- [7] M. J. Coster, A. Joux, B. A. LaMacchia, A. M. Odlyzko, C. Schnorr, J. Stern, *Improved low-density subset sum algorithms*, Computational Complexity, 2: 111–128, 1992.
- [8] I. Dinur, *Approximating $SV P_\infty$ to within almost-polynomial factors is NP-hard*, Theoretical Computer Science, 289(1): 55–71, 2002.
- [9] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science, Springer, 2006.
- [10] Y. Hecht and M. Safra, *Deterministic Hardness-of-Approximation of Unique-SVP and GapSVP in ℓ_p -norms for $p > 2$* , arXiv:2510.16991, Preprint, 2025.
- [11] R. M. Karp, *Reducibility among Combinatorial Problems*, In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (eds) Complexity of Computer Computations. The IBM Research Symposia Series, 85–103, 1972.
- [12] S. Khot, *Hardness of approximating the shortest vector problem in lattices*, J. ACM, 52(5): 789–808, 2005.
- [13] J. C. Lagarias and A. M. Odlyzko, *Solving low-density subset sum problems*, Journal of the ACM, 32(1): 229–246, 1985.
- [14] A. K. Lenstra, H. W. Lenstra, L. Lovász, *Factoring Polynomials with Rational Coefficients*, Ann. of Math., 261(4): 515–534, 1982.
- [15] Y. Lu, R. Zhang, L. Peng, D. Lin, *Solving Linear Equations Modulo Unknown Divisors: Revisited.*, In: Iwata, T., Cheon, J. (eds) Advances in Cryptology – ASIACRYPT 2015, Lecture Notes in Computer Science 9452, Springer, Berlin, Heidelberg, 2015.
- [16] D. Micciancio, *The shortest vector in a lattice is hard to approximate to within some constant*, SIAM J. Comput., 30(6): 2008–2035, 2001.
- [17] C. Peikert, *Public-key cryptosystems from the worst-case shortest vector problem: extended abstract*, In: Proceedings of the forty-first annual ACM Symposium on Theory of Computing (STOC), 333–342, 2009.
- [18] C. Schnorr, M. Euchner, *Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems*, Mathematical Programming, 66: 181–199, 1994.
- [19] P. van Emde Boas, *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*, Technical Report, 1981.

NORCE RESEARCH

Current address: Nygårdsgaten 112, 5008 Bergen, Norway

Email address: `mahi@norce-research.no`