Efficient reductions among lattice problems[*]

Daniele Micciancio[†]

October 9, 2007

Abstract

We give various deterministic polynomial time reductions among approximation problems on point lattices. Our reductions are both efficient and robust, in the sense that they preserve the rank of the lattice and approximation factor achieved. Our main result shows that for any $\gamma \geq 1$, approximating *all* the *successive minima* of a lattice (and, in particular, approximately solving the *Shortest Independent Vectors Problem*, SIVP_γ) within a factor γ reduces under deterministic polynomial time rank-preserving reductions to approximating the *Closest Vector Problem* (CVP) within the same factor γ. This solves an open problem posed by Blömer in (ICALP 2000). As an application, we obtain faster algorithms for the exact solution of SIVP that run in time $n! \cdot s^{O(1)}$ (where n is the rank of the lattice, and s the size of the input,) improving on the best previously known solution of Blömer (ICALP 2000) by a factor 3^n. We also show that SIVP, CVP and many other lattice problems are equivalent in their exact version under deterministic polynomial time rank-preserving reductions.

1 Introduction

A lattice L is the set of intersection points of an infinite n-dimensional grid. The *successive minima* $\lambda_i(L)$ (for $i = 1, \ldots, n$) are among the most fundamental parameters associated to a lattice, and are defined as the smallest values $\lambda_i(L)$ such that the sphere of radius $\lambda_i(L)$ centered around the origin contains at least i linearly independent lattice vectors. Lattice approximation problems have been widely investigated since the discovery of the basis reduction algorithm of Lenstra, Lenstra and Lovasz [LLL82], initially for their applications in cryptanalysis and combinatorial optimization [BO91, JS98, NS01] and more recently as a potential basis for cryptographic function design [Ajt04, GGH97, AD97, Cai03, Mic04, Reg04, MR07, Mic07]. The most important and widely studied lattice approximation problems (for approximation factor $\gamma \geq 1$) are:

- the *Shortest Vector Problem* (SVP): given a lattice, find an approximately shortest nonzero lattice vector, i.e., a vector of length at most $\gamma \cdot \lambda_1$,
- the *Closest Vector Problem* (CVP): given a lattice and a target point, find a lattice point approximately closest to the target, i.e., a lattice point at a distance from the target that is at most γ times the distance of the closest lattice point, and

[*] To appear in Proceedings of SODA 2008.

[†] Computer Science and Engineering Department, University of California at San Diego, La Jolla, CA 92093, USA. Email: daniele@cs.ucsd.edu.

- the *Shortest Independent Vectors Problem* (SIVP): given a lattice, find a maximal set of approximately shortest linearly independent lattice vectors, i.e., n linearly independent vectors (where n is the rank of the lattice) of length at most $\gamma\lambda_n$.

There is a wide gap between the small (constant or sub-polynomial in the lattice rank) approximation factors for which these problems are known to be intractable [Ajt98, Mic01, Kho05, BS99, DKRS03], and the the large (exponential) factors for which the problems are known to be solvable in polynomial time [LLL82, Bab86, Sch87, AKS01]. Given our limited understanding of the complexity of lattice approximation problems, it is natural to ask how these problems relate to each other for arbitrary approximation factors. Are these problems computationally equivalent? Are some harder than others? More specifically, we ask if there are polynomial time reductions among these problems that work for any approximation factor, and preserve both the rank of the lattice and the quality of approximation. The importance of preserving the rank of the lattice in reductions among lattice problems cannot be overemphasized. For example, since the best currently known solution to most lattice problems runs in time exponential in the rank, even doubling the rank would result in an exponential slow down in any algorithmic application of the reduction. To date, the only known polynomial time reduction that preserves both the rank and approximation is the one from SVP to CVP given by Goldreich, Micciancio, Safra and Seifert in [GMSS99]. The only other similar result we are aware of is a *non-deterministic* polynomial time reduction from SIVP_γ to CVP_γ given by Guruswami, Micciancio and Regev [GMR05] for the purpose of showing that SIVP_γ is not NP-hard, under standard complexity assumptions.

In this paper we give various dimension preserving deterministic polynomial time reductions between lattice approximation problems. The main result is a reduction from SIVP to CVP that preserves both the approximation factor and rank of the lattice. In fact, we prove something stronger: we give a reduction from the problem of finding linearly independent lattice vectors achieving *all* successive minima of the lattice to solving CVP. Specifically, we consider the following problem:

- the *successive minima problem* (SMP): given a lattice, find linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ (where n is the rank of the lattice) of length at most $\|\mathbf{v}_i\| \leq \gamma\lambda_i$ for all $i = 1, \ldots, n$.

This is a classic problem in the mathematical study of lattices that subsumes both SVP and SIVP as special cases. (Any solution to SMP_γ is also a solution to SIVP_γ, and the first vector in it is a solution to SVP_γ.) So, our result subsumes the reduction from SVP_γ to CVP_γ of [GMSS99] as a special case, and improves on the non-deterministic polynomial time reduction from SIVP to CVP of [GMR05].

Technically, we reduce SIVP to CVP by introducing a simple generalization of SVP that bears strong similarities with both SVP and CVP. In the shortest vector problem, given a lattice basis $\mathbf{B}$, we want to minimize the length $\|\mathbf{Bx}\|$ (where $\mathbf{x}$ is an integer vector) subject to the constraint that $x_i \neq 0$ for some i. We consider a variant SVP$'$ of the shortest vector problem where the index i is also given as part of the input, and the goal is to minimize $\|\mathbf{Bx}\|$ subject to the constraint that $x_i \neq 0$ for the given index i. The shortest vector in a lattice $\mathbf{B}$ can be easily found by solving all SVP$'$ instances $(\mathbf{B}, i)$ for $i = 1, \ldots, n$. One can also think of SVP$'$ as a variant of CVP, where the target vector $(\mathbf{b}_i)$ can be used multiple times: given a lattice $[\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_n]$ and target $\mathbf{t} = \mathbf{b}_i$, find the lattice point closest to a nonzero multiple of the target.

We give simple deterministic polynomial time reductions from SMP_γ to SVP'_γ and from SVP'_γ to CVP_γ. Our reductions preserve both the approximation factor and lattice rank. Moreover, the reductions work for any approximation factor γ and norm. As a result we get the claimed reduction from SIVP_γ to CVP_γ. An immediate application of our reduction is an algorithm to solve SIVP exactly in time $n!s^{O(1)}$ (where n is the rank of the lattice and s is the size of the input) by reducing it to CVP and then using the CVP algorithm of Blömer [Blö00]. The best previously known algorithm for the exact solution of SIVP (also given in [Blö00]) had running time $3^n \cdot n! \cdot s^{O(1)}$.

Next, we consider two other lattice problems recently introduced by Blömer and Naewe to design algorithms for the deterministic (resp. randomized) exact (resp. approximate) solution of SIVP. Specifically, we consider

- the *Generalized Closest Vector Problem* (GCVP), introduced by Blömer in [Blö00] for the design of deterministic algorithms that solve SIVP exactly, and
- the *Subspace Avoiding Problem* (SAP), a special case of GCVP recently introduced by Blömer and Naewe [BN07] to design and analyze randomized lattice approximation algorithms for SIVP and CVP.

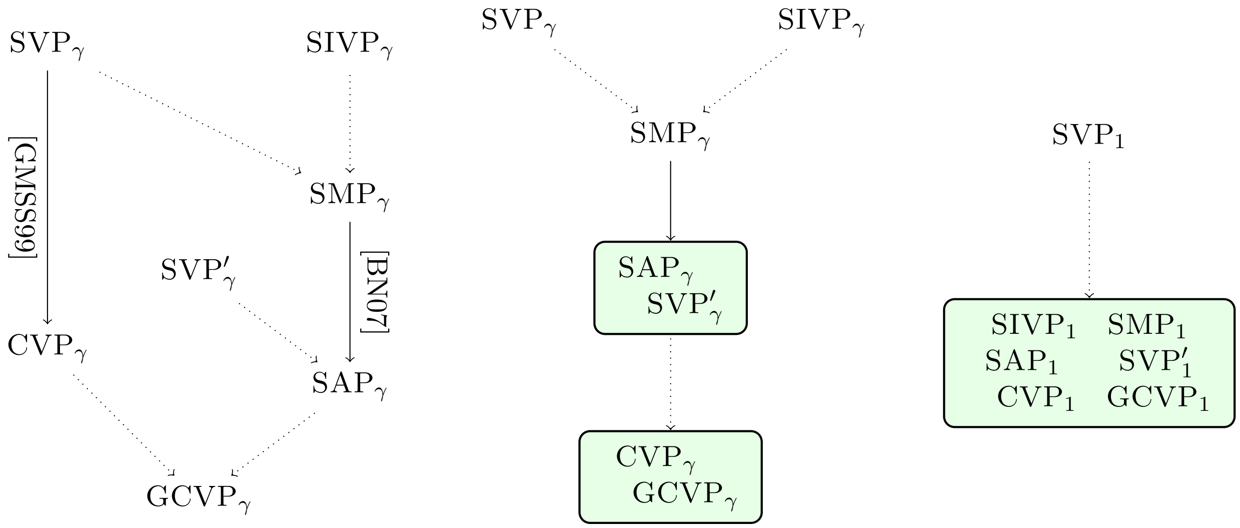
Using the same simple techniques from our main reduction, we are able to show that GCVP_γ and SAP_γ are equivalent to CVP_γ and SVP'_γ respectively, under polynomial time reductions that preserve both the approximation factor and rank of the lattice. Our equivalence results hold for any norm and approximation factor, and answer in the affirmative the open questions (posed by Blömer in [Blö00]) about the polynomial time reducibility of SIVP and GCVP to CVP.

Our results have interesting implications about the exact solvability of lattice problems (in the Euclidean norm) under randomized reductions. The fastest known deterministic algorithms for the solution of all lattice problems considered in this paper have running time $n^{O(n)}$. However, using randomization, Ajtai, Kumar and Sivakumar [AKS01] were able to improve the running time for the exact solution of SVP to a simple exponential function $2^{O(n)}$. Following [AKS01], there have been various attempts to generalize the randomized techniques of [AKS01] to the solution of other lattice problems, most notably CVP (as done in [AKS02]) and SIVP (as done in [BN07]). Albeit reducing the running time to $2^{O(n)}$, none of these attempts have led so far to an algorithm that solves CVP or SIVP (or any other lattice problems) exactly: [AKS02, BN07] only give $2^{O(n)}$-time randomized algorithms to solve CVP_γ and SIVP_γ within a constant factor $\gamma = 1 + \epsilon$, for arbitrary small $\epsilon > 0$. Our results imply that, in their exact versions (and at least in the Euclidean norm,) all lattice problems considered in this paper $\mathrm{SIVP}, \mathrm{SMP}, \mathrm{CVP}, \mathrm{SVP}', \mathrm{SAP}, \mathrm{GCVP}$ (with the only exception of SVP) are equivalent under deterministic polynomial-time rank-preserving reductions. So, exact solutions can be found in exponential time $2^{O(n)}$ either for *all* or for *none* of them. The shortest vector problem SVP reduces to any of them, but no reduction is known in the opposite direction. So, this perhaps explains why generalizing the randomized algorithm of [AKS01] to other lattice problems has, so far, led only to approximation algorithms.

The equivalence of lattice problems in the exact version is obtained by completing the sequence of approximation preserving reductions $\mathrm{SIVP}_\gamma \leq \mathrm{SMP}_\gamma \leq \mathrm{SVP}'_\gamma \leq \mathrm{SAP}_\gamma \leq \mathrm{GCVP}_\gamma \leq \mathrm{CVP}_\gamma$, with a new reduction from CVP to SIVP for the exact version of those problems. For simplicity, the last reduction is presented only for the Euclidean norm, though, with some care, the result can be generalized to many other norms. A polynomial time reduction from exact CVP to exact SIVP (in the Euclidean norm) had been previously given by Blömer and Seifert [BS99]. However, their reduction increases the rank of the lattice by 1. This has only a minor impact on the complexity of the problems, so for all practical purposes, the reduction of [BS99] can already be taken as a proof that exact CVP is not harder than exact SIVP. Still, the question remains if in order to reduce

CVP to SIVP, it is necessary to increase the lattice rank. Here we show that this is not the case, and that the reduction of [BS99] can be modified to make it rank preserving.

On the technical side, our modification of the reduction of [BS99] involves a new primal/dual dimension reduction method that might be of independent interest. Giving an approximation preserving reduction from CVP$_\gamma$ to SIVP$_\gamma$ would show that all lattice problems (except SVP) are equivalent not only in their exact version, but also in their approximate version, and it is left as an open problem. Known relations among lattice problems, prior to this paper, and including the new reductions, are summarized in Figure 1.



(a) Previously known relations.

(b) Relations among lattice approximation problems.

(c) Relations among exact lattice problems in ℓ_2.

Figure 1: Previously known relations among lattice problems, and relations including the results in this paper. Dotted arrows are trivial relations, where one problem is a special case of the other. Arrows indicate polynomial time reductions that preserve the lattice rank and approximation factor, and boxes enclose classes of equivalent problems under the same kind of reductions.

Outline In Section 2 we give some background about lattices. The main results of this paper are presented in Section 3 where we reduce SIVP and SMP to CVP. In Section 4 we present additional results establishing the computational equivalence of various lattice problems. Section 5 concludes with a discussion of the problems left open in this paper.

2 Preliminaries

In this section we give some general background about lattices, and formally define all lattice problems studied in this paper. A norm is a function $\mathbf{x} \mapsto \|\mathbf{x}\| \in \mathbb{R}^+$ such that $\|\mathbf{x}\| \geq 0$ with equality if and only if $\mathbf{x} = \mathbf{0}$, $\|a \cdot \mathbf{x}\| = |a| \cdot \|\mathbf{x}\|$, and $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$. Any norm induces a corresponding distance function $\operatorname{dist}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$. Lattice problems are typically formulated and studied in the Euclidean norm $\|\mathbf{x}\| = \sqrt{\sum_i x^2}$, but all definitions and most results in this paper hold for any norm, subject to some basic computability requirements. Specifically, we assume that

- given two vectors $\mathbf{x}$ and $\mathbf{y}$, one can efficiently determine if $\|\mathbf{x}\| \leq \|\mathbf{y}\|$
- given a vector $\mathbf{x}$ and a linear subspace S, one can efficiently determine (a lower bound to) the distance between $\mathbf{x}$ and S.

Throughout the paper, we refer to norms satisfying these two conditions as *efficiently computable*. The distance function is extended to sets in the customary way: $\operatorname{dist}(\mathbf{x}, S) = \min_{\mathbf{y} \in S} \operatorname{dist}(\mathbf{x}, \mathbf{y})$. The linear space spanned by a set of n vectors $\mathbf{S}$ is denoted $\operatorname{span}(\mathbf{S}) = \{\sum_i x_i \mathbf{s}_i : x_i \in \mathbb{R} \text{ for } 1 \leq i \leq n\}$.

We now describe some basic definitions related to lattices. For a more in-depth discussion, see [MG02]. A *lattice* is the set of all integer combinations

$$\left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \colon x_i \in \mathbb{Z} \text{ for } 1 \leq i \leq n \right\}$$

of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in $\mathbb{R}^m$. The number n is called the *rank* of the lattice, and the set of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is called a *basis*. A basis can be represented by the matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{m \times n}$ having the basis vectors as columns. The lattice generated by $\mathbf{B}$ is denoted $\mathcal{L}(\mathbf{B})$. Notice that $\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} \colon \mathbf{x} \in \mathbb{Z}^n\}$, where $\mathbf{Bx}$ is the usual matrix-vector multiplication. Two bases $\mathbf{B}, \tilde{\mathbf{B}}$ generate the same lattice $\mathcal{L}(\mathbf{B}) = \mathcal{L}(\tilde{\mathbf{B}})$ if and only if $\mathbf{B} = \tilde{\mathbf{B}}\mathbf{U}$ for some unimodular matrix $\mathbf{U}$. (A unimodular matrix is a square matrix with integer entries and determinant ±1.) For computational purposes, it is usually assumed that all lattice vectors have integers (or more generally rational) entries, so that the lattice can be represented by an integer (resp. rational) matrix $\mathbf{B} \in \mathbb{Z}^{m \times n}$.

The dual of a lattice Λ is the set

$$\Lambda^* = \{\mathbf{x} \in \operatorname{span}(\Lambda) \colon \forall \mathbf{y} \in \Lambda . \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\}$$

of all vectors that have integer scalar product ($\langle \mathbf{x}, \mathbf{y} \rangle = \sum_i x_i y_i$) with all lattice vectors. The dual of a lattice is a lattice, and if $\Lambda = \mathcal{L}(\mathbf{B})$ is the lattice generated by basis $\mathbf{B}$, then $\mathbf{B}^* = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}$ is a basis for the dual lattice, where $\mathbf{B}^T$ is the transpose of $\mathbf{B}$. A sub-lattice of $\mathcal{L}(\mathbf{B})$ is a lattice $\mathcal{L}(\mathbf{S})$ such that $\mathcal{L}(\mathbf{S}) \subseteq \mathcal{L}(\mathbf{B})$.

The *minimum distance* of a lattice Λ, denoted $\lambda_1(\Lambda)$, is the minimum distance between any two distinct lattice points, and equals the length of the shortest nonzero lattice vector:

$$\begin{aligned} \lambda_1(\Lambda) &= \min\{\operatorname{dist}(\mathbf{x}, \mathbf{y}) : \mathbf{x} \neq \mathbf{y} \in \Lambda\} \\ &= \min\{\|\mathbf{x}\| : \mathbf{x} \in \Lambda \setminus \{\mathbf{0}\}\} . \end{aligned}$$

This definition can be generalized to define the ith successive minimum as the smallest λ_i such that $\lambda_i \mathcal{B} = \{\mathbf{x} \colon \|\mathbf{x}\| \leq \lambda_i\}$ contains i linearly independent lattice points:

$$\lambda_i(\Lambda) = \min\{r \colon \dim(\operatorname{span}(\Lambda \cap r\mathcal{B})) \geq i\}.$$

Another important constant associated to a lattice is the *covering radius* $\nu(\Lambda)$, defined as

$$\nu(\Lambda) = \max_{\mathbf{x} \in \operatorname{span}(\Lambda)} \{\operatorname{dist}(\mathbf{x}, \Lambda)\}.$$

We often abuse notation and write $\lambda_1(\mathbf{B})$ instead of $\lambda_1(\mathcal{L}(\mathbf{B}))$ and similarly for other lattice parameters.

The following are among the most important and widely studied computational problems on point lattices.

Definition 1 (Shortest Vector Problem (SVP)) *Given a lattice $\mathbf{B} \in \mathbb{Z}^{m\times n}$, find a nonzero lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B}) \setminus \{\mathbf{0}\}$ such that $\|\mathbf{v}\| \leq \gamma\lambda_1(\mathbf{B})$.*

Definition 2 (Shortest Independent Vectors Problem (SIVP)) *Given a lattice $\mathbf{B} \in \mathbb{Z}^{m\times n}$, find n linearly independent lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v}_i\| \leq \gamma\lambda_n(\mathbf{B})$ for all $i = 1, \ldots, n$.*

Definition 3 (Closest Vector Problem (CVP)) *Given a lattice $\mathbf{B} \in \mathbb{Z}^{m\times n}$ and a target vector $\mathbf{t} \in \mathbb{R}^m$, find a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $dist(\mathbf{v}, \mathbf{t}) \leq \gamma\, dist(\mathbf{t}, \mathcal{L}(\mathbf{B}))$.*

SVP and SIVP are both special cases of the following classical mathematical problem.

Definition 4 (Successive Minima Problem (SMP)) *Given a lattice $\mathbf{B} \in \mathbb{Z}^{m\times n}$, find n linearly independent lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v}_i\| \leq \gamma\lambda_i(\mathbf{B})$ for all $i = 1, \ldots, n$.*

Clearly, SVP_γ and SIVP_γ reduce to SMP_γ. We now define three non-standard problems on lattices that will be used in this paper. The first is a simple variant of SVP. In the standard version of SVP, one is asked to minimize the norm $\|\mathbf{Bx}\|$ subject to the condition that $x_i \neq 0$ for some i. We consider a variant where the index i is given as part of the input.

Definition 5 (SVP') *Given a lattice $\mathbf{B} \in \mathbb{Z}^{m\times n}$ and an index $i \in \{1, \ldots, n\}$, find a lattice vector $\mathbf{Bx}$ with $x_i \neq 0$ such that $\|\mathbf{Bx}\| \leq \gamma \min\{\|\mathbf{Bx}\| \colon x_i \neq 0\}$.*

Clearly, SVP reduces to SVP' in polynomial time: on input a lattice $\mathbf{B}$, one can solve all SVP' instances $(\mathbf{B}, i)$ for $i = 1, \ldots, n$, and select the best answer. This problem will be instrumental in reducing SIVP and SMP to CVP.

The next problem was recently introduced in [BN07] to design randomized sampling algorithms for the approximate solution of SIVP and other lattice problems.

Definition 6 (Subspace Avoiding Problem (SAP)) *Given a lattice $\mathbf{B} \in \mathbb{Z}^{m\times n}$ and a linear subspace S, find a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B}) \setminus S$ such that $\|\mathbf{v}\| \leq \gamma\, dist(\mathbf{0}, \mathcal{L}(\mathbf{B}) \setminus S)$.*

Clearly, SVP is a special case of SAP, where $S = \{\mathbf{0}\}$. Also SVP' is a special case of SAP, where $S = \{\mathbf{x} \colon x_i = 0\}$ for some i. In turns, SAP is a special case of the following problem from [Blö00].

Definition 7 (Generalized Closest Vector Problem (GCVP)) *Given a lattice $\mathbf{B} \in \mathbb{Z}^{m\times n}$, a target vector $\mathbf{t} \in \mathbb{R}^n$ and an affine subspace S, find a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B}) \setminus S$ such that $dist(\mathbf{v}, \mathbf{t}) \leq \gamma\, dist(\mathbf{t}, \mathcal{L}(\mathbf{B}) \setminus S)$.*

It is easy to see that both CVP and SAP are special cases of GCVP, where $S = \emptyset$ and $\mathbf{t} = \mathbf{0}$ respectively. There are also distance estimation versions of these problems, where, for example, instead of finding a short nonzero vector in a lattice, it is enough to produce the approximate value of λ_1. These problems are usually formulated as decision (or promise) problems. For example, $\textsc{Gap}\mathrm{SVP}_\gamma$ is the problem of distinguishing pairs $(\mathbf{B}, d)$ where $\lambda_1(\mathbf{B}) \leq d$ from pairs where $\lambda_1(\mathbf{B}) > \gamma d$. It is easy to see that this promise problem is equivalent to computing an approximation $\tilde{\lambda}_1 \in [\lambda_1, \gamma\lambda_1]$ of the length of the shortest lattice vector. Similarly, $\textsc{Gap}\mathrm{SIVP}_\gamma$ and $\textsc{Gap}\mathrm{CVP}_\gamma$

are the promise problems associated to the task of estimating $\lambda_n(\mathbf{B})$ and $\operatorname{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$ within a factor γ. In the exact case (i.e., when $\gamma = 1$) and at least in the Euclidean norm, the distance estimation problems GAPSVP_1 and GAPCVP_1 are equivalent (under polynomial time reductions) to their corresponding search problems SVP_1 and CVP_1. (See [Kan87] and [MG02, Chapter 3].) It follows from the results in this paper that the same is true for $\mathrm{GAPSIVP}_1$ and SIVP_1. However, in the approximate case $\gamma > 1$, no such reduction is known.

3 Reducing SIVP and SMP to CVP

We show that the problem SVP'_γ is at least as hard as SIVP_γ (or, even SMP_γ), and no harder than CVP_γ. We point out that the main results presented in this section can also be obtained combining the reductions in Section 4 with trivial reductions and previously known results. We present the reductions in this section first in order to give a somehow simpler and more direct reduction from SMP to CVP. The reader not interested in the complexity of less standard problems (e.g., SAP, SVP$'$ and GCVP) can read this section and then skip most of Section 4.

The reduction from SMP_γ to SVP'_γ is based on the following simple lemma, which will be useful also later in the paper.

Lemma 1 *There is a polynomial time algorithm that on input a lattice basis $\mathbf{B} \in \mathbb{Q}^{m\times n}$ and a linear subspace S, outputs a new basis $\tilde{\mathbf{B}}$ for $\mathcal{L}(\mathbf{B})$ such that $\mathcal{L}(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_d) = S \cap \mathcal{L}(\mathbf{B})$, where d is the dimension of $S \cap$ span$(\mathbf{B})$.*

Proof. Assume S is represented by a system of linear equations, i.e., $S = \{\mathbf{x} \colon \mathbf{H}\mathbf{x} = \mathbf{0}\}$ for some given matrix $\mathbf{H} \in \mathbb{Q}^{h\times m}$. (If $S = \operatorname{span}(\mathbf{G})$ is given as a generating matrix $\mathbf{G}$, then $\mathbf{H}$ can be efficiently computed using linear algebra as the orthogonal complement of $\mathbf{G}$.) Consider the $h \times n$ matrix $\mathbf{HB}$, and extend it to a square $n \times n$ matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{HB} \\ \mathbf{O} \end{bmatrix}$$

by adding $n - h$ identically zero rows. Any square integer matrix $\mathbf{C}$ can be put into Smith Normal Form $\mathbf{S} = \mathbf{VCU}$, where

- $\mathbf{S}$ is a diagonal matrix, with non-negative diagonal entries such that $s_{i+1,i+1}$ divides $s_{i,i}$ for $i = 1, \ldots, n-1$. In particular, there is a d such that the first d diagonal elements of $\mathbf{S}$ are zero, and the remaining $n - d$ diagonal elements are nonzero.

- $\mathbf{U}$ and $\mathbf{V}$ are unimodular matrices.

(See [Coh93] for more information on the Smith Normal Form, as well as algorithms to compute the matrices $\mathbf{U}, \mathbf{V}$ and $\mathbf{S}$ in polynomial time.)

The output of the algorithm is $\tilde{\mathbf{B}} = \mathbf{BU}$. Clearly, $\tilde{\mathbf{B}}$ is a basis for $\mathcal{L}(\mathbf{B})$ because matrix $\mathbf{U}$ is unimodular. Now, consider the intersection

$$S \cap \mathcal{L}(\mathbf{B}) = S \cap \mathcal{L}(\tilde{\mathbf{B}}) = \{\tilde{\mathbf{B}}\mathbf{x} \colon \mathbf{x} \in \mathbb{Z}^n, \mathbf{H}\tilde{\mathbf{B}}\mathbf{x} = \mathbf{0}\},$$

and let $\mathbf{Q}$ the $n \times h$ matrix consisting of the first h columns of $\mathbf{V}$.

Since the columns of $\mathbf{Q}$ are linearly independent, the condition $\mathbf{H}\tilde{\mathbf{B}}\mathbf{x} = \mathbf{0}$ is equivalent to $\mathbf{QH}\tilde{\mathbf{B}}\mathbf{x} = \mathbf{0}$. But $\mathbf{QH}\tilde{\mathbf{B}} = \mathbf{QHBU} = \mathbf{VCU} = \mathbf{S}$. So, $\mathbf{QH}\tilde{\mathbf{B}}\mathbf{x} = \mathbf{Sx} = \mathbf{0}$ if and only if $x_i = 0$ for all $i > d$, and the set $S \cap \mathcal{L}(\mathbf{B})$ equals $\mathcal{L}(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_d)$. Finally, since $\tilde{\mathbf{B}}$ is a basis, the vectors $\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_d$ are linearly independent, and the dimension of $S \cap \mathcal{L}(\mathbf{B})$ is d. □

We can now give the first interesting reduction.

Theorem 2 *For any approximation factor γ, there is a polynomial time reduction from* SMP_γ *to* SVP'_γ*. The reduction preserves both the rank and dimension of the input lattice, and works for any efficiently computable norm. Moreover, on input a basis $\mathbf{B}$, the reduction makes only oracle calls of type* $\mathrm{SVP}'_\gamma(\tilde{\mathbf{B}}, i)$ *where $\mathcal{L}(\tilde{\mathbf{B}}) = \mathcal{L}(\mathbf{B})$.*

Proof. Let $\mathbf{B}$ be the input lattice. The reduction computes linearly independent lattice vectors $\mathbf{s}_i \in \mathcal{L}(\mathbf{B})$ of length at most $\|\mathbf{s}_i\| \leq \gamma\lambda_i(\mathbf{B})$ iteratively, one at a time. Assume we have found linearly independent lattice vectors $\mathbf{s}_1, \ldots, \mathbf{s}_k$ such that $\|\mathbf{s}_i\| \leq \gamma\lambda_i$ for $i = 1, \ldots, k$, and let $S = \mathrm{span}(\mathbf{s}_1, \ldots, \mathbf{s}_k)$. We want to find one more lattice vector $\mathbf{s}_{k+1} \notin S$ such that $\|\mathbf{s}_{k+1}\| \leq \gamma\lambda_{k+1}$.

Using Lemma 1 we can find a lattice basis $\tilde{\mathbf{B}}$ such that $\mathcal{L}(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_d) = S \cap \mathcal{L}(\mathbf{B})$. Notice that, since $\mathbf{s}_1, \ldots, \mathbf{s}_k \in \mathcal{L}(\mathbf{B})$ are linearly independent, $\dim(\mathcal{L}(\mathbf{B}) \cap S) = \dim(S) = k$. So, it must be $d = k$, and $\mathcal{L}(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_k) = S \cap \mathcal{L}(\mathbf{B})$. Make $n - k$ calls to $\mathrm{SVP}'(\tilde{\mathbf{B}}, i)$ for $i = k+1, \ldots, n$, and let $\mathbf{s}_{k+1} = \mathrm{SVP}'(\tilde{\mathbf{B}}, i)$ be the shortest vector returned by the oracle. Certainly, $\mathbf{s}_{k+1} \notin S$ because it uses $\tilde{\mathbf{b}}_i$ a nonzero number of times. We want to prove that $\|\mathbf{s}_{k+1}\| \leq \gamma\lambda_{k+1}$.

By definition of λ_{k+1}, there exist $k+1$ linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{k+1} \in \mathcal{L}(\mathbf{B}) = \mathcal{L}(\tilde{\mathbf{B}})$ such that $\|\mathbf{v}_i\| \leq \lambda_{k+1}$. Since they are linearly independent, they cannot all belong to S. Let $\mathbf{v}_j = \tilde{\mathbf{B}}\mathbf{x} \notin S$. It must be $x_i \neq 0$ for some $i > k$, for otherwise we would have $\tilde{\mathbf{B}}\mathbf{x} \in S$. Consider the oracle call $\mathrm{SVP}'(\tilde{\mathbf{B}}, i)$ where i is an index such that $x_i \neq 0$. Clearly, the optimal solution to the SVP' instance $(\tilde{\mathbf{B}}, i)$ has length at most $\|\tilde{\mathbf{B}}\mathbf{x}\| \leq \lambda_{k+1}$. So, the oracle returns a vector $\mathbf{s}_{k+1} = \mathrm{SVP}'(\tilde{\mathbf{B}}, i)$ of length at most $\|\mathbf{s}_{k+1}\| \leq \gamma\|\tilde{\mathbf{B}}\mathbf{x}\| \leq \gamma\lambda_{k+1}$. □

Next, we reduce SVP' to CVP. The idea underlying the reduction is that SVP' can be regarded as a variant of CVP where the target vector can be used multiple times.

Theorem 3 *For any approximation factor γ, there is a polynomial time reduction from* SVP'_γ *to* CVP_γ*. The reduction preserves both the rank and dimension of the input lattice, and works for any efficiently computable norm. Moreover, on input $(\mathbf{B}, i)$, all oracle calls made by the reduction are of type $(\tilde{\mathbf{B}}, \mathbf{t})$ where $\mathcal{L}(\tilde{\mathbf{B}}) \subset \mathcal{L}(\mathbf{B})$ and $\mathbf{t} \in \mathcal{L}(\mathbf{B})$.*

Proof. Let $(\mathbf{B}, i)$ be an SVP' input instance, and assume without loss of generality that $i = n$ equals the rank of $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$. Let $\mathbf{Ba}$ be an optimal solution to the input problem, i.e., a lattice vector $\mathbf{Ba}$ with $a_n \neq 0$ such that $\|\mathbf{Ba}\|$ is minimized. For $j = 0, \ldots, \lfloor \log_2 A \rfloor$ (where A is a sufficiently large bound to be determined) call the CVP_γ oracle on input $(\mathbf{B}^{(j)}, \mathbf{t}^{(j)})$ where

- $\mathbf{B}^{(j)} = [\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}, 2^{j+1}\mathbf{b}_n]$ is the matrix obtained multiplying the nth column of $\mathbf{B}$ by 2^{j+1}, and
- $\mathbf{t}^{(j)} = 2^j\mathbf{b}_n$.

Let $\mathbf{B}^{(j)}\mathbf{x}^{(j)}$ be the solution returned by the CVP oracle on input $(\mathbf{B}^{(j)}, \mathbf{t}^{(j)})$. The output of the reduction is the shortest among all vectors $\mathbf{B}^{(j)}\mathbf{x}^{(j)} - \mathbf{t}^{(j)}$. Notice that any such vector uses $\mathbf{b}_n$ a nonzero

$$2^{j+1}x_n^{(j)} - 2^j = 2^j \cdot (2x_n^{(j)} - 1) \neq 0$$

number of times. So, the output is a feasible solution to SVP$'$ instance $(\mathbf{B}, n)$. We need to prove that the output solution is within a factor γ from the shortest, i.e., $\|\mathbf{B}^{(j)}\mathbf{x}^{(j)} - \mathbf{t}^{(j)}\| \leq \gamma\|\mathbf{Ba}\|$ for some j.

Let j be the highest power of 2 such that 2^j divides a_n. Since a_n is nonzero, j is well defined and $a_n = 2^j \cdot (2a - 1)$ for some integer a. Consider the CVP instance $(\mathbf{B}^{(j)}, \mathbf{t}^{(j)})$. The lattice $\mathcal{L}(\mathbf{B}^{(j)})$ contains a vector $\mathbf{B}^{(j)}\mathbf{a}'$ (where $\mathbf{a}'$ is obtained from $\mathbf{a}$ replacing the nth entry a_n with a) at distance $\|\mathbf{Ba}\|$ from $\mathbf{t}^{(j)}$. So, the CVP oracle must find a lattice vector such that $\|\mathbf{B}^{(j)}\mathbf{x}^{(j)} - \mathbf{t}^{(j)}\|$ is at most $\gamma\|\mathbf{Ba}\|$ as desired.

It remains to determine a suitable bound A. Since $|a_n| = 2^j \cdot |2a - 1| \geq 2^j$, it is enough to set A to any number at least as big as $|a_n|$. Let α be (a lower bound to) the distance between vector $\mathbf{b}_n$ and the linear subspace S spanned by $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}$. (For the Euclidean norm, this is the length of the orthogonalized vector $\mathbf{b}_n^*$, i.e., the component of $\mathbf{b}_n$ orthogonal to $\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}$.) Since (for any $\mathbf{a}$) the vector $\mathbf{Ba}$ has norm at least $\alpha \cdot |a_n|$, and $\|\mathbf{Ba}\| \leq \|\mathbf{b}_n\|$ by the optimality of $\mathbf{a}$, it must be $|a_n| \leq \|\mathbf{b}_n\|/\alpha$ and we can set $A = \|\mathbf{b}_n\|/\alpha$. Since A can be computed in polynomial time, $\log_2 A$ is bounded by a polynomial in the input length, and the reduction runs in polynomial time. □

Combining the two theorems we get the following corollary.

Corollary 4 *There is a dimension, rank and approximation preserving reduction from* SMP$_\gamma$ *(and therefore* SIVP$_\gamma$*) to* CVP$_\gamma$*. The reduction works for any computable norm and approximation factor, and it has the additional property that on input a lattice* $\mathbf{B}$*, it makes only oracle calls of type* $(\tilde{\mathbf{B}}, \mathbf{t})$ *where* $\mathcal{L}(\tilde{\mathbf{B}}) \subset \mathcal{L}(\mathbf{B})$ *and* $\mathbf{t} \in \mathcal{L}(\mathbf{B})$*.*

4 Equivalence results

Now we turn to the proof that several problems on lattices are equivalent from a computational point of view under polynomial time rank-preserving reductions. We start by considering the subspace avoiding problem SAP of [BN07], where, on input a lattice $\mathbf{B}$ and linear subspace S, the goal is to find the shortest lattice vector outside S. Notice that our simple variant SVP$'$ of the shortest vector problem is a special case of SAP, where the subspace to avoid is $S = \{\mathbf{x}: x_i = 0\}$. So, there is a trivial reduction from SVP$'_\gamma$ to SAP$_\gamma$. The following corollary is an immediate application of Lemma 1, and shows that general SAP$_\gamma$ instances are not harder than SVP$'$.

Corollary 5 *For any efficiently computable norm and approximation factor* γ*, the problems* SAP$_\gamma$ *and* SVP$'_\gamma$ *are equivalent under deterministic polynomial time rank-preserving reductions.*

Proof. We already observed that SVP$'$ is a special case of SAP, so there is a trivial reduction from SVP$'_\gamma$ to SAP$_\gamma$. We reduce SAP$_\gamma$ to SVP$'_\gamma$. Let $(\mathbf{B}, S)$ be an instance of SAP$_\gamma$. We invoke Lemma 1 on input $(\mathbf{B}, S)$ to get an equivalent SAP$_\gamma$ instance $(\tilde{\mathbf{B}}, S)$ where $\mathcal{L}(\tilde{\mathbf{B}}) = \mathcal{L}(\mathbf{B})$ and $S = \text{span}(\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_d)$. Next, we call the SVP$'_\gamma$ oracle on input $(\tilde{\mathbf{B}}, i)$ for $i = d+1, \ldots, n$, and select the shortest of the vectors returned by the oracle calls. Clearly, all oracle answers belong to $\mathcal{L}(\mathbf{B}) \setminus S$ because they use some $\tilde{\mathbf{b}}_i \notin S$ (for $i > d$) a nonzero number of times. Moreover, the optimal solution to the SAP_γ input instance must use one of the vectors $\tilde{\mathbf{b}}_i$ (for $i > d$) a nonzero number of times, so the corresponding call $\text{SVP}'_\gamma(\tilde{\mathbf{B}}, i)$ returns a vector of length at most γ times longer than this optimal solution. □

The equivalence between CVP_γ and its generalization GCVP_γ introduced by [Blö00] is less trivial, but it can still be easily proved by adapting our reduction from SVP'_γ to CVP_γ.

Theorem 6 *For any efficiently computable norm and approximation factor γ, the problems CVP_γ and GCVP_γ are equivalent under deterministic polynomial time rank-preserving reductions.*

Proof. Clearly, CVP_γ is a special case of GCVP_γ (with $S = \emptyset$), so there is a trivial reduction from CVP_γ to GCVP_γ. We give a reduction in the opposite direction. Let $(\mathbf{B}, \mathbf{t}, S)$ be a GCVP_γ instance, where $\mathbf{B} \in \mathbb{Z}^{m \times n}$ is an m-dimensional lattice, $S \subset \mathbb{R}^m$ is an affine subspace, and $\mathbf{t} \in \mathbb{Q}^m$ is a target point. First of all, we determine if S intersects the lattice $\mathcal{L}(\mathbf{B})$, and if so, we find a lattice point in S. Both tasks can be efficiently performed using linear algebra. If S contains no lattice point, then we can immediately map it to CVP_γ instance $(\mathbf{B}, \mathbf{t})$. So, assume S contains a lattice point $\mathbf{v} \in S \cap \mathcal{L}(\mathbf{B})$. We can also assume, without loss of generality, that $\mathbf{v}$ is the origin, for, otherwise, we can consider an equivalent GCVP_γ instance $(\mathbf{B}, S - \mathbf{v}, \mathbf{t} - \mathbf{v})$, where $S - \mathbf{v}$ certainly contains the origin. To sum up, so far we have reduced general GCVP_γ instances to instances where S is a linear subspace.

Next, we use Lemma 1 to find a basis $\tilde{\mathbf{B}}$ for $\mathcal{L}(\mathbf{B})$ such that $S = \text{span}([\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_k])$. Using this basis $\tilde{\mathbf{B}}$, we define a collection of CVP_γ instances as follows. For any $i = k+1, \ldots, n$, and $j = 0 \ldots, \lfloor \log_2 A \rfloor$ (where A is a sufficiently large bound which can be determined as in the proof of Theorem 3), define

- $\tilde{\mathbf{B}}_i^j = [\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_{i-1}, 2^{j+1} \cdot \tilde{\mathbf{b}}_i, \tilde{\mathbf{b}}_{i+1}, \ldots, \tilde{\mathbf{b}}_n]$, and
- $\mathbf{t}_i^j = \mathbf{t} - 2^j \cdot \tilde{\mathbf{b}}_i$.

Call the CVP_γ oracle on all instances $(\tilde{\mathbf{B}}_i^j, \mathbf{t}_i^j)$ to get solutions $\mathbf{z}_i^j \in \mathcal{L}(\mathbf{B}_i^j)$. The output of the reduction is the vector $\mathbf{z}_i^j + 2^j \cdot \tilde{\mathbf{b}}_i$ closest to the target $\mathbf{t}$. This completes the description of the reduction from GCVP to CVP.

We need to show that the reduction is correct. Notice that the output of the reduction $\mathbf{z}_i^j + 2^j \cdot \tilde{\mathbf{b}}_i$ is a lattice point because $\mathbf{z}_i^j \in \mathcal{L}(\tilde{\mathbf{B}}_i^j) \subset \mathcal{L}(\mathbf{B})$ and $\tilde{\mathbf{b}}_i \in \mathcal{L}(\mathbf{B})$. Moreover, $\mathbf{z}_i^j + 2^j \cdot \tilde{\mathbf{b}}_i \notin S$ because it uses the basis vector $\tilde{\mathbf{b}}_i \notin S$ a nonzero (in fact, in $2^j(2\mathbb{Z}+1)$) number of times. So, the output of the reduction is a feasible solution to GCVP_γ instance $(\mathbf{B}, \mathbf{t}, S)$. All we need to do is to bound the quality of the approximation achieved by the reduction, i.e., show that for some i and j, the distance of the vector $\mathbf{z}_i^j + 2^j \cdot \tilde{\mathbf{b}}_i$ from the target $\mathbf{t}$ is within a factor γ from the optimal. Let $\tilde{\mathbf{B}}\mathbf{a}$ be the optimal solution to the GCVP_γ instance $(\mathbf{B}, \mathbf{t}, S)$ expressed in terms of the basis $\tilde{\mathbf{B}}$. We know that $[a_{k+1}, \ldots, a_n]$ is nonzero because $\tilde{\mathbf{B}}\mathbf{a} \notin S \supset \mathcal{L}([\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_k])$. In particular, there is an index $i > k$ such that $a_i \neq 0$. Let 2^j be the highest power of 2 that divides a_i, and consider CVP_γ instance $(\tilde{\mathbf{B}}_i^j, \mathbf{t}_i^j)$. Let $\mathbf{a}'$ be the vector obtained by replacing the ith coordinate of $\mathbf{a}$ by $a'_i = ((a_i/2^j) - 1)/2$. The distance of $\mathbf{t}_i^j$ from the lattice $\mathcal{L}(\tilde{\mathbf{B}}_i^j)$ is at most

$$\text{dist}(\mathbf{t}_i^j, \mathcal{L}(\tilde{\mathbf{B}}_i^j)) \quad \leq \quad \|\mathbf{t}_i^j - \tilde{\mathbf{B}}_i^j \mathbf{a}'\|$$

$$
\begin{aligned}
&= \|\mathbf{t} - 2^j \cdot \tilde{\mathbf{b}}_i - \sum_{h \neq i} \tilde{\mathbf{b}}_h a_h - 2^{j+1} \cdot \tilde{\mathbf{b}}_i a'_i\| \\
&= \|\mathbf{t} - \sum_{h} \tilde{\mathbf{b}}_h a_h\| = \|\mathbf{t} - \tilde{\mathbf{B}}\mathbf{a}\|.
\end{aligned}
$$

So, the CVP$_\gamma$ oracle will return a lattice vector $\mathbf{z}_i^j = \tilde{\mathbf{B}}_i^j \mathbf{x}$ within distance $\gamma \cdot \|\mathbf{t} - \tilde{\mathbf{B}}\mathbf{a}\|$ from the target $\mathbf{t}_i^j$, and the corresponding vector output by the reduction satisfies

$$
\|(\mathbf{z}_i^j + 2^j \cdot \tilde{\mathbf{b}}_i) - \mathbf{t}\| = \|\mathbf{z}_i^j - \mathbf{t}_i^j\| \leq \gamma \|\mathbf{t} - \tilde{\mathbf{B}}\mathbf{a}\|
$$

as required. □

We conclude this section showing that all lattice problems considered in this paper, with the only exception of SVP, are equivalent in their exact version under deterministic polynomial-time rank-preserving reductions. The result builds on a simple (rank increasing) reduction from CVP to SIVP in the Euclidean norm given in [BS99].

Our result can be generalized to other norms using the fact that the (Banach-Mazur) distance of any n-dimensional norm from the Euclidean norm ℓ_2 is bounded by a polynomial in n. The most delicate part of the adaptation to other norms is the generalization of the reduction in [BS99] to norms other than ℓ_2. For simplicity (and in order to use the reduction from [BS99]) we present the result only for the special case of the Euclidean norm.

Corollary 7 SIVP*,* SMP*,* SVP$'$ *and* CVP *in the Euclidean norm are equivalent in their exact version under polynomial time rank-preserving reductions, and* SVP *reduces to any of them.*

Proof. We have already given rank-preserving reductions SIVP$_\gamma \leq$ SMP$_\gamma \leq$ SVP$'_\gamma \leq$ SAP$_\gamma \leq$ GCVP$_\gamma \leq$ CVP$_\gamma$. In order to prove the corollary we need to give a rank-preserving reduction from CVP to SIVP. A reduction between these two problems is given by Blömer and Seifert in [BS99]. However, their reduction maps rank n instances of CVP to rank $(n+1)$ instances of SIVP. Here we use the transference theorems of Banaszczyk [Ban93] to modify their reduction into a rank preserving one. This involves a primal/dual rank reduction method that might be of independent interest. The idea is to use short vectors in the input lattice to find short vectors in the dual, and then use the short dual vectors to find closest vectors in the original lattice.

Let $(\mathbf{B}, \mathbf{t})$ be a CVP instance of rank n, and $\mathbf{w} \in \mathcal{L}(\mathbf{B})$ a lattice vector closest to $\mathbf{t}$. We want to find a lattice point at distance $\|\mathbf{w} - \mathbf{t}\|$ from $\mathbf{t}$, making a polynomial number of calls to an oracle that solves SIVP problems of rank n.

We first use the SIVP oracle to find a nonzero lattice vector of length at most $\|\mathbf{x}\| \leq \lambda_n(\mathbf{B})$. Vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ partitions the dual lattice $\mathcal{L}(\mathbf{B}^*) = \bigcup_{i \in \mathbb{Z}} S_i$ into subsets

$$
S_i = \{\mathbf{y} \in \mathcal{L}(\mathbf{B}^*) \colon \langle \mathbf{y}, \mathbf{x} \rangle = i\}
$$

lying on $(n-1)$-dimensional hyperplanes at distance $1/\|\mathbf{x}\|$ from each other. By the transference theorems we know that $\lambda_1(\mathbf{B}^*) \leq n/\lambda_n(\mathbf{B}) \leq n/\|\mathbf{x}\|$. So the shortest nonzero vector in the dual lattice belongs to subset S_i for some $i \in \{-n, \ldots, n\}$. In fact, since any lattice is symmetric about the origin, it is enough to consider $i \in \{0, \ldots, n\}$. For each i in this range, we can find the shortest nonzero vector in S_i as follows:

- For $i = 0$, this is a SVP instance of rank $n-1$, which can be solved by reducing it to CVP using the rank preserving reduction of Goldreich et al. [GMSS99], and then reducing the resulting CVP instance (which has rank $n-1$) to SIVP using the reduction of [BS99].
- For $i \neq 0$, we first find an arbitrary lattice point $\mathbf{v} \in S_i$, and the point $\mathbf{w}$ in the hyperplane $H_i = \{\mathbf{y} \in \operatorname{span}(\mathbf{B}) \colon \langle \mathbf{y}, \mathbf{x} \rangle = i\}$ closest to the origin. Next, we find the lattice point $\mathbf{u} \in S_0$ closest to the target $\mathbf{w} - \mathbf{v}$. This is a CVP instance of rank $(n-1)$ and it can be solved by reduction to SIVP as in the $i = 0$ case. The point $\mathbf{u} + \mathbf{v}$ belongs to S_i and it is the point in S_i of smallest norm.

Let $\mathbf{d}$ be the smallest of all the dual vectors found for $i \in \{0, \ldots, n\}$. Since the shortest vector in the dual lattice belongs to one of these sets S_i, we have $\|\mathbf{d}\| \leq \lambda_1(\mathbf{B}^*)$. This dual vector partitions the original lattice $\mathcal{L}(\mathbf{B}) = \bigcup_{i \in \mathbb{Z}} T_i$ into subsets

$$T_i = \{\mathbf{y} \in \mathcal{L}(\mathbf{B}) \colon \langle \mathbf{y}, \mathbf{d} \rangle = i\}$$

lying on $(n-1)$-dimensional hyperplanes at distance $1/\|\mathbf{d}\| = 1/\lambda_1(\mathbf{B}^*)$ from each other. We observe that the lattice vector $\mathbf{w}$ closest to $\mathbf{t}$ must belong to T_i for an index $i \in \langle \mathbf{t}, \mathbf{d} \rangle \pm n$ because

$$|\langle \mathbf{d}, \mathbf{w} \rangle - \langle \mathbf{d}, \mathbf{t} \rangle| = |\langle \mathbf{d}, \mathbf{w} - \mathbf{t} \rangle| \leq \|\mathbf{d}\| \cdot \|\mathbf{w} - \mathbf{t}\| \leq \lambda_1^*(\mathbf{B}) \nu(\mathbf{B})$$

which, by the transference theorem, is at most n. For each i, we can find the lattice vector in T_i closest to the target by solving a CVP problem of rank $n-1$. Each of these lower dimensional instances can be reduced to SIVP (of rank n) using the reduction from [BS99]. □

5 Open Problems

As Figure 1(b) shows, we have now a fairly good picture of the relations among the various lattice problems studied in this paper. The first question left open in this paper is to determine the relation between SVP_γ and SIVP_γ.

Open Problem 1 *Are there deterministic polynomial time reductions between* SVP_γ *and* SIVP_γ *that preserve both the rank of the lattice and quality of approximation? Are the two problems equivalent? Incomparable? Or one strictly harder than the other?*

We remark that for the exact ($\gamma = 1$) version of the problems in the Euclidean norm, there is a reduction from SVP to SIVP. (See Figure 1(c).) Also, the transference theorems of [Ban93] immediately give a polynomial time deterministic rank-preserving reduction from $\textsc{GapSVP}_{\gamma n}$ to $\textsc{GapSIVP}_\gamma$ for any factor γ. This supports the conjecture that SVP_γ is not harder than SIVP_γ. Proving a reduction in the opposite direction (from SIVP to SVP) appears harder (even just for the exact version of the problems) as it would imply the NP-hardness of SVP under deterministic polynomial time reductions. To date, all known NP-hardness proofs for SVP [Ajt98, Mic01, Kho05, HR07] (with the only exception of a deterministic reduction of [Mic01] based on a plausible, but unproven, number theoretic conjecture) are randomized. Moreover, they produce SVP instances with much higher rank than the size of the original problem. So, even finding a randomized rank-preserving reduction from SIVP to SVP would be an interesting result.

The other main open question is whether the hierarchy of problems shown in Figure 1(b) is strict, or some of the problems can be proved equivalent. In the case of the exact version in the Euclidean norm, Figure 1(c) shows that all problems, with the exception of SVP, collapse to the same class. This was proved giving a rank-preserving reduction from CVP to SIVP. Can the same be done for approximate versions of the problems?

Open Problem 2 *Is there a deterministic polynomial time reduction from* CVP_γ *to* SIVP_γ *that preserves the rank of the lattice and approximation factor?*

Answering the above question would show that all problems except SVP_γ are equivalent, and SVP_γ is not harder than any of them. Even reducing one of the easier problems (SMP_γ or SAP_γ) to SIVP_γ would be an interesting result, as it would collapse all the intermediate problems together, as well as establish a relation between SVP_γ and SIVP_γ.

Acknowledgments

The author thanks Stephen Checkoway for a useful conversation that led to the proof of Corollary 7, and the anonymous referees for useful comments. This research was supported in part by NSF grant CCF-0634909. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of STOC '97*, pages 284–293. ACM, May 1997.

[Ajt98] Miklós Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of STOC '98*, pages 10–19. ACM, May 1998.

[Ajt04] Miklós Ajtai. Generating hard instances of lattice problems. *Complexity of Computations and Proofs, Quaderni di Matematica*, 13:1–32, 2004. Preliminary version in STOC 1996.

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of STOC '01*, pages 266–275. ACM, July 2001.

[AKS02] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In *Proceedings of CCC '02*, pages 53–57. IEEE, May 2002.

[Bab86] László Babai. On Lovasz' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993.

[Blö00] Johannes Blömer. Closest vectors, successive minima and dual HKZ-bases of lattices. In *Proceedings of ICALP '00*, volume 1853 of *LNCS*, pages 248–259. Springer, July 2000.

[BN07] Johannes Blömer and Stefanie Naewe. Sampling methods for shortest vectors, closest vectors and successive minima. In *Proceedings of ICALP '07*, volume 4596 of *LNCS*, pages 65–77. Springer, July 2007.

[BO91] Ernest F. Brickell and Andrew M. Odlyzko. Cryptanalysis: A survey of recent results. In G. J. Simmons, editor, *Contemporary Cryptology*, chapter 10, pages 501–540. IEEE Press, 1991.

[BS99] Johannes Blömer and Jean-Pierre Seifert. On the complexity of computing short linearly independent vectors and short bases in a lattice. In *Proceedings of STOC '99*, pages 711–720. ACM, May 1999.

[Cai03] Jin-Yi Cai. A new transference theorem in the geometry of numbers and new bounds for Ajtai's connection factor. *Discrete Applied Mathematics*, 126(1):9–31, March 2003. Preliminary versions in CCC 1999 and COCOON 1999.

[Coh93] Henri Cohen. *A course in computational algebraic number theory*. Springer-Verlag, 1993.

[DKRS03] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003. Preliminary version in FOCS 1998.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Proceedings of CRYPTO '97*, volume 1294 of *LNCS*, pages 112–131. Springer, August 1997.

[GMR05] Venkatesan Guruswami, Daniele Micciancio, and Oded Regev. The complexity of the covering radius problem. *Computational Complexity*, 14(2):90–120, 2005. Preliminary version in CCC 2004.

[GMSS99] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.

[HR07] Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *Proceedings of STOC '07*, pages 469–477. ACM, June 2007.

[JS98] Antoine Joux and Jacques Stern. Lattice reduction: A toolbox for the cryptanalyst. *Journal of Cryptology*, 11(3):161–185, 1998.

[Kan87] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of operation research*, 12(3):415–440, August 1987.

[Kho05] Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM*, 52(5):789–808, September 2005. Preliminary version in FOCS 2004.

[LLL82] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):513–534, December 1982.

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.

[Mic01] Daniele Micciancio. The shortest vector problem is NP-hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, March 2001. Preliminary version in FOCS 1998.

[Mic04] Daniele Micciancio. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM Journal on Computing*, 34(1):118–169, 2004. Preliminary version in STOC 2002.

[Mic07] Daniele Micciancio. Generalized compact knapsaks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 2007. To appear in special issue on average-case complexity. Preliminary version in FOCS 2002.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measure. *SIAM Journal on Computing*, 37(1):267–302, 2007. Preliminary version in FOCS 2004.

[NS01] Phong Nguyen and Jacques Stern. The two faces of lattices in cryptology. In *Proceedings of CaLC '01*, volume 2146 of *LNCS*, pages 146–180. Springer, March 2001.

[Reg04] Oded Regev. New lattice based cryptographic constructions. *Journal of the ACM*, 51(6):899–942, 2004. Preliminary version in STOC 2003.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2–3):201–224, 1987.