ELSEVIER
Theoretical Computer Science 285 (2002) 55–71
Theoretical Computer Science
www.elsevier.com/locate/tcs

Approximating SVP$_\infty$ to within almost-polynomial factors is NP-hard

Irit Dinur

Dept. of Computer Science, School of Mathematical Sciences, Ramat Aviv, 69978 Tel Aviv, Israel

Received 14 May 2001

Abstract

We show SVP$_\infty$ and CVP$_\infty$ to be NP-hard to approximate to within $n^{c/\log\log n}$ for some constant $c>0$. We show a direct reduction from SAT to these problems, that combines ideas from Arora et al. (Proc. 34th IEEE Symp. on Foundations of Computer Science, 1993, p. 724) and Dinur et al. (Approximating-CVP to within almost-polynomial factors is NP-hard, manuscript, 1999), along with some modifications. Our result is obtained without relying on the PCP characterization of NP, although some of our techniques are derived from the proof of the PCP characterization itself (STOC: ACM Symposium on Theory of Computing (STOC), 1999). © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Shortest vector; Lattice problems; Hardness of approximation

1. Introduction

1.1. Background

A lattice $L=L(v_1,\ldots,v_n)$, for linearly independent vectors $v_1,\ldots,v_n\in R^k$ is the additive group generated by the basis vectors, i.e. the set $L=\{\sum a_iv_i \mid a_i\in\mathbb{Z}\}$. Given L, the shortest vector problem (SVP$_p$) is to find the shortest non-zero vector in L. The length is measured in Euclidean l_p norm $(1\leqslant p\leqslant\infty)$. The closest vector problem (CVP$_p$) is the non-homogeneous analog, i.e. given L and a vector y, find a vector in L, closest to y.

These lattice problems have been introduced in the 19th century, and have been studied since. Minkowsky and Dirichlet tried, with little success, to come up with approximation algorithms for these problems. It was much later that the lattice reduction algorithm was presented by Lenstra et al. [14], achieving a polynomial-time algorithm approximating the shortest lattice vector to within the exponential factor $2^{n/2}$, where n is

E-mail address: dinuri@tau.ac.il (I. Dinur).

PII: S0304-3975(01)00290-0

the dimension of the lattice. Babai [4] applied LLL's methods to present an algorithm that approximates CVP to within a similar factor. Schnorr [17] improved on LLL's technique, reducing the factor of approximation to $(1+\varepsilon)^n$, for any constant $\varepsilon > 0$, for both CVP and SVP. These positive approximation results hold for l_p norm for any $p \geqslant 1$ yet are quite weak, achieving only extremely large (exponential) approximation factors. The shortest vector problem is particularly important, quoting [3], because even the above relatively weak approximation algorithms have been used in a host of applications, including integer programming, solving low-density subset-sum problems and breaking knapsack based codes [13], simultaneous diophantine approximation and factoring polynomials over the rationals [14], and strongly polynomial-time algorithms in combinatorial optimization [11].

Interest in lattice problems has been recently renewed due to a result of Ajtai [1], showing a reduction, from a version of SVP, to the *average-case* of the same problem.

Only recently [2] showed a randomized reduction from the NP-complete problem subset-sum to SVP. This has been improved [6], showing approximation hardness for some small factor $(1 + 1/n^{\varepsilon})$. Micciancio [16] has then significantly strengthened Ajtai's result, showing SVP hard to approximate to within some constant factor.

The above results all apply to SVP_p, for finite p. SVP with the maximum norm l_∞, appears to be a harder problem. Lagarias showed SVP_∞ to be NP-hard in its exact decision version. Arora et al. [3] utilized the PCP characterization of NP to show that both CVP (for l_p norm for any p) and SVP_∞ are quasi-NP-hard to approximate to within $2^{(\log n)^{1-\varepsilon}}$ for any constant $\varepsilon > 0$. Recently, the hardness result for approximating CVP has been strengthened [10, 9] showing that it is NP-hard to approximate to within a factor of $n^{c/\log\log n}$ (where n is the lattice dimension, and $c > 0$ is some constant). In this paper we similarly strengthen the hardness result for approximating SVP_∞.

A g-approximation algorithm for SVP_2 implies a $\sqrt{n}g$-approximation algorithm for SVP_∞, since for every vector v, $\|v\|_\infty \leqslant \|v\|_2 \leqslant \|v\|_\infty\sqrt{n}$. Thus hardness for approximating SVP_∞ to within a factor $\sqrt{n}g$ will imply the hardness for approximating SVP_2 to within factor g.

So far there is still a huge gap between the positive results, showing approximations for SVP and CVP with exponential factors, and the above hardness results. Nevertheless, some other results provide a discouraging indication for improving the hardness result beyond a certain factor. Goldreich and Goldwasser [12] showed that approximating both SVP_2 and CVP_2 to within $\sqrt{n}$ and approximating SVP_∞ and CVP_∞ to within $n/\mathrm{O}(\log n)$ is in NP $\cap$ co-AM. Hence it is unlikely for any of these problems to be NP-hard.

1.2. Our result

We prove that approximating SVP_∞ and CVP_∞ to within a factor of $n^{c/\log\log n}$ is NP-hard (where n is the lattice dimension and $c > 0$ is some arbitrary constant).

1.3. Technique

We obtain our result by modifying (and slightly simplifying) the framework of Dimur et al. [10, 9]. Starting out from SAT, we construct a new SAT instance that has the additional property that it is either totally satisfiable, or, not even weakly-satisfiable in some specific sense (to be elaborated upon below). We refer to such a SAT instance as an $\mathcal{SSAT}_\infty$ instance (this is a variant of [10] $\mathcal{SSAT}$). The construction reducing SAT to $\mathcal{SSAT}_\infty$ is the main part of the paper. The construction has a tree-like recursive structure that is a simplification of techniques from [10, 9], along with some additional observations tailored to the l_∞ norm.

We finally obtain our result by reducing $\mathcal{SSAT}_\infty$ to SVP$_\infty$ and to CVP$_\infty$. These reductions are combinatorial and utilize an additional idea from [3].

Hardness-of-approximation results are naturally divided into those that are obtained via reduction from PCP, and those that are not. Although the best previous hardness result for SVP$_\infty$ [3] relies on the PCP characterization of NP, our proof does not. We do, however, utilize some techniques similar to those used in the proof of the PCP characterization of NP itself. In fact, the nature of the SVP$_\infty$ problem eliminates some of the technical complications from [8–10]. Thus, we believe that SVP$_\infty$ makes a good candidate (out of all of the lattice problems) for pushing the hardness-of-approximation factor to within polynomial range.

1.4. Structure of the paper

In Section 2, we present a variant of the $\mathcal{SSAT}$ problem from [10] which we call $\mathcal{SSAT}_\infty$. We then proceed with some standard (and not so standard) definitions. In Section 3, we give the reduction from SAT to $\mathcal{SSAT}_\infty$, whose correctness is proven in Section 4. Finally, in Section 5 we describe the (simple) reduction from $\mathcal{SSAT}_\infty$ to SVP$_\infty$ and to CVP$_\infty$, establishing the hardness of approximating SVP$_\infty$ and CVP$_\infty$.

2. Definitions

2.1. $\mathcal{SSAT}_\infty$

A SAT instance is a set $\Psi = \{\psi_1, \ldots, \psi_n\}$ of *tests* (Boolean functions) over variables $V = \{v_1, \ldots, v_m\}$. We denote the range of the variables by $\mathcal{F}$, and the set of satisfying assignments for $\psi_i \in \Psi$ by $\mathcal{R}_{\psi_i}$. The Cook–Levin [7, 15] theorem states that it is NP-hard to distinguish whether or not the system is satisfiable (i.e. whether there is an assignment to the variables that satisfies all of the tests). We next define $\mathcal{SSAT}_\infty$, a version of SAT that has the additional property that when the instance is not satisfiable, it is not even 'weakly-satisfiable' in a sense that will be formally defined below.

We recall the following definitions (Definitions 1–3) from [10].

Definition 1 (*super-assignment to tests*). A super-assignment is a function S mapping to each $\psi \in \Psi$ a value from $\mathbb{Z}^{\mathcal{R}_\psi}$. In other words $S(\psi)$ is a vector of integer coefficients, one for each value $r \in \mathcal{R}_\psi$. Denote by $S(\psi)[r]$ for $r \in \mathcal{R}_\psi$ the rth coordinate of $S(\psi)$.

If $S(\psi) = \vec{0}$ we say that $S(\psi)$ is trivial. If $S(\psi)[r] \neq 0$, we say that the value r appears in $S(\psi)$. A *natural assignment* (an assignment in the usual sense) is identified with a super-assignment that assigns each $\psi \in \Psi$ a unit vector with a 1 in the corresponding coordinate. In this case, exactly one value appears in each $S(\psi)$.

We next define the projection of a super-assignment to a test onto each of its variables. Consistency between tests will amount to equality of projections on mutual variables.

Definition 2 (*projection*). Let S be a super-assignment to the tests. We define the projection of $S(\psi)$ on a variable x of ψ, $\pi_x(S(\psi)) \in \mathbb{Z}^{|\mathcal{F}|}$, in the natural way:

$$\forall a \in \mathcal{F}: \quad \pi_x(S(\psi))[a] \stackrel{\text{def}}{=} \sum_{r \in \mathcal{R}_\psi, r|_x = a} S(\psi)[r].$$

We shall now proceed to define the notion of consistency between tests. If the projections of two tests on each mutual variable x are equal (in other words, they both give x the same super-assignment), we say that the super-assignments of the tests are consistent.

Definition 3 (*consistency*). Let S be a super-assignment to the tests in Ψ. S is consistent if for every pair of tests ψ_i and ψ_j with a mutual variable x,

$$\pi_x(S(\psi_i)) = \pi_x(S(\psi_j)).$$

Given a system $\Psi = \{\psi_1, \ldots, \psi_n\}$, a super-assignment $S : \Psi \to \mathbb{Z}^{\mathcal{R}}$ is called *not-all-zero* if there is at least one test $\psi \in \Psi$ for which $S(\psi) \neq \vec{0}$. The *norm* of a super-assignment S is defined

$$\|S\| \stackrel{\text{def}}{=} \max_{\psi \in \Psi}(\|S(\psi)\|_1),$$

where $\|S(\psi)\|_1$ is the standard l_1 norm. Note that the norm of a natural super-assignment is 1.

The gap of $\mathcal{SSAT}_\infty$ is formulated in terms of the norm of the minimal super-assignment that maintains consistency.

Definition 4 (g-$\mathcal{SSAT}_\infty$). An instance of $\mathcal{SSAT}_\infty$ with parameter g

$$I = \langle \Psi = \{\psi_1, \ldots, \psi_n\},\ V = \{v_1, \ldots, v_m\}, \{\mathcal{R}_{\psi_1}, \ldots, \mathcal{R}_{\psi_n}\} \rangle$$

consists of a set Ψ of *tests* over a common set V of variables that take values in a field $\mathcal{F}$. The parameters m and $|\mathcal{F}|$ are always bounded by some polynomial in n. Each test $\psi \in \Psi$ has associated with it a list $\mathcal{R}_\psi$ of assignments to its variables, called the *satisfying assignments* or the *range* of the test ψ. The problem is to distinguish between the following two cases,

Yes: There is a consistent natural assignment for Ψ.
No: Every not-all-zero consistent super-assignment is of norm $>g$.

Remark. The definition of $\mathcal{SSAT}_\infty$ differs from that of $\mathcal{SSAT}$ [10] only in the characterization of when a super-assignment falls into the 'no' category. On one hand, $\mathcal{SSAT}_\infty$ imposes a weaker requirement of not-all-zero rather than the non-triviality of $\mathcal{SSAT}$. On the other hand, the norm of a super assignment S is measured by a 'stronger' measure, taking the maximum of $\|S(\psi)\|_1$ over all ψ, rather than the average as in $\mathcal{SSAT}$.

Theorem 5 ($\mathcal{SSAT}_\infty$ Theorem). *$\mathcal{SSAT}_\infty$ is NP-hard for $g=n^{c/\log\log n}$ for some $c>0$.*

We conjecture that a stronger statement is true, which would imply that SVP_∞ is NP-hard to approximate to within a *constant power* of the dimension.

Conjecture 6. *$\mathcal{SSAT}_\infty$ is NP-hard for $g=n^c$ for some constant $c>0$.*

2.2. LDFs, super-LDFs

Throughout the paper, let $\mathcal{F}$ denote a finite field $\mathcal{F}=\mathbb{Z}_p$ for some prime number $p>1$. We will need the following definitions.

Definition 7 (*low degree function*—*$[r,d]$-LDF*). A function $f:\mathcal{F}^d\to\mathcal{F}$ is said to have degree r if its values are the point evaluation of a polynomial on $\mathcal{F}^d$ with degree $\leqslant r$ in each variable. In this case we say that f is an $[r,d]$-LDF, or $f\in\mathrm{LDF}_{r,d}$.

Sometimes we omit the parameters and refer simply to an LDF.

Definition 8 (*low degree extension*). Let m,d be natural numbers, and let $\mathcal{H}\subset\mathcal{F}$ such that $|\mathcal{H}^d|=m$. A vector $(a_0,\dots,a_{m-1})\in\mathcal{F}^m$ can be naturally identified with a function $A:\mathcal{H}^d\to\mathcal{F}$ by viewing points in $\mathcal{H}^d$ as representing numbers in base $|\mathcal{H}|$.

There exists exactly one $[|\mathcal{H}|-1,d]$-LDF $\hat{A}:\mathcal{F}^d\to\mathcal{F}$ that extends A. $\hat{A}$ is called the $|\mathcal{H}|-1$ degree extension of A in $\mathcal{F}$.

A D-dimensional affine subspace (*D-cube* for short) $\mathcal{C}\subset\mathcal{F}^d$ is said to be *parallel* to the axises if it can be written as $\mathcal{C}=x+\mathrm{spn}(e_{i_1},\dots,e_{i_D})$, where $x\in\mathcal{F}^d$ and $e_i\in\mathcal{F}^d$ is the ith axis vector, $e_i=(0,\dots,1,\dots,0)$. We write the parameterization of the cube $\mathcal{C}$ as follows:

$$\mathcal{C}(\bar{t})\stackrel{\text{def}}{=}x+\sum_{j=1}^{D}t_je_{i_j}\in\mathcal{F}^d \quad \text{for } \bar{t}=(t_1,\dots,t_D)\in\mathcal{F}^D.$$

While for a general (non-parallel) cube, the restriction of an $[r,d]$-LDF to a D-cube in $\mathscr{F}^d$ is an $[rD,D]$-LDF, its restriction to a parallel D-cube is an $[r,D]$-LDF. We will need the following (simple) proposition,

Proposition 9. *Let $f:\mathscr{F}^d\to\mathscr{F}$. Suppose, for every parallel D-cube $\mathscr{C}\subset\mathscr{F}^d$ the function $f|_{\mathscr{C}}:\mathscr{F}^D\to\mathscr{F}$ defined by*

$$\forall x\in\mathscr{F}^D\quad f|_{\mathscr{C}}(x)=f(\mathscr{C}(x))$$

is an $[r,D]$-LDF. Then f is an $[r,d]$-LDF.

Similar to the definition of super-assignments, we define a *super*-$[r,d]$-LDF (or a super-LDF for short) $\mathscr{G}\in\mathbb{Z}^{\mathrm{LDF}_{r,d}}$ to be a vector of integer coefficient $\mathscr{G}[P]$ per LDF $P\in\mathrm{LDF}_{r,d}$. This definition arises naturally from the fact that the tests in our final construction will range over LDFs. We further define the *norm* of a super-LDF to be the l_1 norm of the corresponding coefficient vector.

We say that an LDF $P\in\mathrm{LDF}_{r,d}$ *appears* in $\mathscr{G}$ iff $\mathscr{G}[P]\neq 0$. A point x is called *ambiguous* for a super-LDF $\mathscr{G}$, if there are two LDFs P_1,P_2 appearing in $\mathscr{G}$ such that $P_1(x)=P_2(x)$. The following (simple) property of *low-norm* super-LDFs is heavily used in this paper.

Proposition 10 (low ambiguity). *Let $\mathscr{G}$ be a super-$[r,d]$-LDF of norm $\leqslant g$. The fraction of ambiguous points for $\mathscr{G}$ is $\leqslant\mathrm{amb}(r,d,g)\overset{\mathrm{def}}{=}\binom{g}{2}rd/|\mathscr{F}|$.*

Proof. The number of non-zero coordinates in an integer vector whose l_1 norm is g is $\leqslant g$. There are $\leqslant\binom{g}{2}$ pairs of LDFs appearing in $\mathscr{G}$, and each pair agrees on at most $rd/|\mathscr{F}|$ of the points in $\mathscr{F}^d$. □

The following embedding-extension technique taken from [8] is used in our construction.

Definition 11 (*embedding extension*). Let $b,k>1$ and t be natural numbers. We define the *embedding extension mapping* $E_b:\mathscr{F}^t\to\mathscr{F}^{tk}$ as follows. E_b maps any point $x=(\xi_1,\ldots,\xi_t)\in\mathscr{F}^t$ to $y\in\mathscr{F}^{tk}$, $y=E_b(x)=(\eta_1,\ldots,\eta_{tk})$ by

$$E_b(\xi_1,\ldots,\xi_t)\overset{\mathrm{def}}{=}(\xi_1,(\xi_1)^b,(\xi_1)^{b^2},\ldots,(\xi_1)^{b^{k-1}},\ldots,\xi_t,(\xi_t)^b,(\xi_t)^{b^2},\ldots,(\xi_t)^{b^{k-1}}).$$

The following (simple) proposition, shows that any LDF on $\mathscr{F}^t$ can be represented by an LDF on $\mathscr{F}^{tk}$ with significantly lower degree:

Proposition 12. *Let $f:\mathscr{F}^t\to\mathscr{F}$ be a $[b^k-1,t]$-LDF, for integers $t>0$, $b>1$, $k>1$. There is a $[b-1,tk]$-LDF $f_{\mathrm{ext}}:\mathscr{F}^{tk}\to\mathscr{F}$ such that*

$$\forall x\in\mathscr{F}^t:\quad f(x)=f_{\mathrm{ext}}(E_b(x)).$$

For any $[b-1,kt]$-LDF f, its ‘restriction’ to the manifold $f|_{E_b}:\mathscr{F}^t\to\mathscr{F}$ is defined as

$$\forall x\in\mathscr{F}^t \quad f|_{E_b}(x)\overset{\text{def}}{=}f(E_b(x))$$

and is a $[b^k-1,t]$-LDF (the degree in a variable ξ_i of $f|_{E_b}$ is $\leqslant(b-1)(b^0+b^1+\cdots+b^{k-1})=b^k-1$).

Let $\tilde{\mathscr{G}}$ be a super-$[b^k-1,t]$-LDF (i.e. a vector in $\mathbb{Z}^{\mathrm{LDF}_{b^k-1,t}}$). Its *embedding-extension* is the super-$[b-1,tk]$-LDF $\mathscr{G}$ defined by

$$\forall f\in \mathrm{LDF}_{b-1,tk} \quad \mathscr{G}[f]\overset{\text{def}}{=}\tilde{\mathscr{G}}[f|_{E_b}].$$

In a similar manner, the *restriction* of a super-$[b-1,tk]$-LDF $\mathscr{G}$ is a super-$[b^k-1,t]$-LDF $\tilde{\mathscr{G}}$ defined by

$$\forall f\in \mathrm{LDF}_{b^k-1,t} \quad \tilde{\mathscr{G}}[f]\overset{\text{def}}{=}\mathscr{G}[f_{\text{ext}}].$$

The following proposition holds (e.g. by a counting argument).

Proposition 13. *Let $\mathscr{G}_1,\mathscr{G}_2$ be two super-$[b-1,tk]$-LDFs, and let $\tilde{\mathscr{G}}_1,\tilde{\mathscr{G}}_2$ be their respective restrictions (with parameter b). $\tilde{\mathscr{G}}_1=\tilde{\mathscr{G}}_2$ if and only if $\mathscr{G}_1=\mathscr{G}_2$.*

3. The construction

We prove that $\mathscr{SSAT}_\infty$ is NP-hard via a reduction from SAT, described herein. We adopt the whole framework of the construction from [9], and refer the reader there for a more detailed exposition.

Let $\Phi=\{\varphi_1,\dots,\varphi_n\}$ be an instance of SAT, viewed as a set of Boolean *tests* over Boolean variables $V_\Phi=\{\mathtt{x}_1,\dots,\mathtt{x}_m\}$, ($m=n^{c'}$ for some constant $c'>0$) such that each test depends on $D=3$ variables. Cook's theorem [7] states that it is NP-hard to decide whether there is an assignment for V_Φ satisfying all of the tests in Φ.

Starting from Φ, we shall construct an $\mathscr{SSAT}_\infty$ test-system Ψ over variables $V_\Psi\supset V_\Phi$. Our new variables V_Ψ will be non-Boolean, ranging over a field $\mathscr{F}$, with $|\mathscr{F}|=n^{c/\log\log n}$ for some constant $c>0$. An assignment to V_Ψ will be interpreted as an assignment to V_Φ by identifying the value $0\in\mathscr{F}$ with the Boolean value `true` and any other non-zero value with `false`.

3.1. Constructing the CR-forest

In order to construct the $\mathscr{SSAT}_\infty$ instance $I=\langle\Psi,V,\{\mathscr{R}_{\psi_1},\dots,\mathscr{R}_{\psi_n}\}\rangle$ we need to describe for each test $\psi\in\Psi$, which variables it depends on, and its satisfying assignments $\mathscr{R}_\psi$. We begin by constructing the CR-forest, which is a combinatorial object holding the underlying structure of Ψ. The forest $\mathbf{F}_n(\Phi)$ will have a tree $\mathbf{T}_\varphi$ for every test $\varphi\in\Phi$. Each node in the forest will have a set of variables associated with it. For every leaf there will be one test depending on the variables associated with that leaf.

Let us (briefly) describe one tree $\mathbf{T}_\varphi$ in the forest $\mathbf{F}_n(\Phi)$.

Every tree will be of depth $K \leqslant \log\log n$ (however, not all of the leaves will be at the bottom level).

Each node v in the tree will have a domain $\mathbf{dom}_v = \mathscr{F}^d$ of points ($\mathbf{dom}_v = \mathscr{F}^{d_0}$ in case v is the root node) associated with it. We set $d_0 = \log\log n$ and $d = a(D+2)$ and fix $a = 4$.

The offsprings of a non-leaf node v will be labeled each by a distinct $(D+2)$-cube $\mathscr{C}_v$ of $\mathbf{dom}_v$ (this part is slightly simpler than in [9]),

$$\mathbf{labels}(v) \overset{\text{def}}{=} \{\mathscr{C} \mid \mathscr{C} \text{is a } (D+2)\text{-cube in } \mathbf{dom}_v\}.$$

The points in the domain $\mathbf{dom}_v$ of each node v will be mapped to some of Ψ's variables, by the injection $\mathbf{var}_v : \mathbf{dom}_v \to V_\Psi$. This mapping is defined inductively as follows. For each node v, we denote by V_v the set of 'fresh new' variables mapped from $\mathbf{dom}_v$ (i.e. none of the nodes defined inductively so far have points mapped to these variables). Altogether

$$V \overset{\text{def}}{=} V_\Psi = \bigcup_{\substack{v \in \mathbf{T}_\varphi \\ \varphi \in \Phi}} V_v.$$

For the root node, $\mathbf{var}_{\text{root}_\varphi} : \mathbf{dom}_{\text{root}_\varphi} \to V_\Psi$ is defined (exactly as in [9]) by mapping $\mathscr{H}^{d_0} \subseteq \mathbf{dom}_{\text{root}_\varphi} = \mathscr{F}^{d_0}$ to V_Φ and the rest of the points to the rest of $V_{\text{root}_\varphi} \overset{\text{def}}{=} \hat{V}_\Phi \subset V_\Psi$ (i.e. the low-degree-extension of V_Φ). It is important that $\mathbf{var}_{\text{root}_\varphi}$ is defined independently of φ.

For a non-root node v with parent u, the points of the cube $\mathscr{C}_v \in \mathbf{labels}(u)$ labeling v are mapped into the domain $\mathbf{dom}_v$ by the embedding extension mapping, $E_{b_v} : \mathscr{C}_v \to \mathbf{dom}_v$, defined above in Section 2.2 (the parameter b_v specified below depends on the specific node v, rather than just on v's level as in [9]). These points are u's points that are 'passed on' to the offspring v. We think of the point $y = E_{b_v}(x) \in \mathbf{dom}_v$ as 'representing' the point $x \in \mathscr{C}_v \subset \mathbf{dom}_u$, and define $\mathbf{var}_v : \mathbf{dom}_v \to V_\Psi$ as follows,

Definition 14 ($\mathbf{var}_v$, *for a non-root node* v). Let v be a non-root node, let u be v's parent, and let $\mathscr{C}_v \subset \mathbf{dom}_u$ be the label attached to v. For each point $y \in E_{b_v}(\mathscr{C}_v) \subset \mathbf{dom}_v$ define $\mathbf{var}_v(y) \overset{\text{def}}{=} \mathbf{var}_u(E_{b_v}^{-1}(y))$, i.e. points that 'originated' from $\mathscr{C}_v$ are mapped to the previous-level variables, that their pre-images in $\mathscr{C}_v$ were mapped to. For each 'new' point $y \in \mathbf{dom}_v \backslash E_{b_v}(\mathscr{C}_v)$ we define $\mathbf{var}_v(y)$ to be a distinct 'fresh' variable from V_v.

The parameters used for the embedding extension mappings E_{b_v} are $t = D+2$, $k = d/t = a$. We set the degree of the root node $r_{\text{root}_\varphi} = |\mathscr{H}| = |\mathscr{F}|^{1/10}$ and r_v and b_v (for non-root nodes v) are defined by the following recursive formulas:

$$b_v = \begin{cases} \sqrt[a]{r_u + 1} & \mathscr{C}_v \text{ is parallel to the axises,} \\ \sqrt[a]{r_u(D+2)+1} & \text{otherwise,} \end{cases}$$

$$r_v = b_v - 1.$$

We stop the recursion and define a node to be a leaf (i.e. define its labels to be empty) whenever $r_v \leqslant 2(D+2)$.

We will show below that b_v, r_v decrease with the level of v until for some level $K < \log\log n$, $r_v \leqslant 2(D+2) = \mathrm{O}(1)$. (This may happen to some nodes sooner than others, therefore not all of the leaves reside in level K.)

We now complete the construction by describing the tests and their satisfying assignments.

Definition 15 (*tests*). Ψ will have one test ψ_v for each leaf v in the forest. ψ_v will depend on the $|\mathscr{F}|^d$ variables in $\mathbf{var}_v(\mathbf{dom}_v)$. The set of satisfying assignments for ψ_v's variables, $\mathscr{R}_{\psi_v}$, will consist of assignments A that satisfy the following two conditions:
(1) A is an $[r_v, d]$-LDF on $\mathbf{var}_v(\mathbf{dom}_v)$.
(2) If $v \in \mathbf{T}_\varphi$ for $\varphi \in \Phi$ and φ's variables appear in $\mathbf{var}_v(\mathbf{dom}_v)$, then A must satisfy φ.

3.2. Construction size

We assume, for simplicity, that all parameters K, d_0, d, b_v, r_v are natural numbers. Recall that we defined $d_0 \overset{\text{def}}{=} \log\log n$ and $d \overset{\text{def}}{=} a(D+2)$ for $a=4$. We also set $r_{\mathrm{root}_\varphi} = |\mathscr{F}|^{1/10} = n^{c/\log\log n}$.

We claim that the forest's depth is bounded by $K \leqslant \log\log n$. Suppose to the contrary that there's a node v of depth K all of whose ancestors u have $r_u > a(D+2)$.

For this purpose we prove by simple induction that every node v of level i obeys $r_v \leqslant \max((r_{\mathrm{root}_\varphi})^{1/2^i}, 2(D+2))$. For $r_{\mathrm{root}_\varphi}$ this indeed holds. Assume by induction that it holds for nodes u of level $\leqslant i$. Let v be a node of level $i+1$ with parent u. Thus, $r_u > 2(D+2)$ (otherwise u would have been a leaf) and so

$$r_v < b_v \leqslant \sqrt[a]{(D+2)r_u + 1} < \sqrt[a]{2r_u(D+2)} \leqslant (r_u)^{2/a} \leqslant \sqrt{r_u} \leqslant (r_{\mathrm{root}_\varphi})^{1/2^{i+1}}.$$

We set K to be the minimal i for which $(r_{\mathrm{root}_\varphi})^{1/2^{i+1}} \leqslant 2(D+2) = \mathrm{O}(1)$. Since $r_{\mathrm{root}_\varphi} = 2^{c\log n/\log\log n}$, $K \leqslant \lfloor \log(c\log n/\log\log n)\rfloor + 1 < \log\log n$. This completes the induction.

The range of the tests. The tests of the test-system range over $[r,d]$-LDFs for $r \leqslant 2(D+2) = \mathrm{O}(1)$. The number of monomials of degree r, and dimension $d = a(D+2) = \mathrm{O}(1)$ is bounded by $(r+1)^d = \mathrm{O}(1)$. The number of $[r,d]$-LDFs is hence bounded by $|\mathscr{F}|^{\mathrm{O}(1)} < \mathrm{O}(n)$ and therefore the range of the tests is polynomial in n.

The number of tests and variables. It is only left to verify that the size of the forest is polynomial. We have $|\Phi| = n$ trees, so let's verify that the number of nodes in each tree is polynomially-bounded.

Consider a tree $\mathbf{T} = \mathbf{T}_\varphi \in \mathbf{F}_n(\Phi)$. root_φ has $\leqslant (|\mathscr{F}|^{d_0})^{D+3} = n^{\mathrm{O}(1)}$ offsprings and each node in level i $(0 < i < K)$ has $\leqslant (|\mathscr{F}|^d)^{D+3} = |\mathscr{F}|^{\mathrm{O}(1)}$ offsprings. Altogether the number of nodes in $\mathbf{T}$ is bounded by

$$n^{\mathrm{O}(1)} \prod_{i=1}^{K} |\mathscr{F}|^{\mathrm{O}(1)} = n^{\mathrm{O}(1)} |\mathscr{F}|^{\mathrm{O}(K)} = n^{\mathrm{O}(1)} (2^{\log n/\log\log n})^{\mathrm{O}(\log\log n)} = n^{\mathrm{O}(1)}.$$

Hence the number of tests in Ψ is polynomial, and the number of variables is $\leqslant |\mathscr{F}|^d|\Psi| = n^{O(1)}$.

4. Correctness of the construction

4.1. Completeness

Lemma 16 (completeness). *If there is an assignment $\mathscr{A}: V_\Phi \to \{\texttt{true}, \texttt{false}\}$ satisfying all of the tests in Φ, then there is a natural assignment $\mathscr{A}_\Psi : V_\Psi \to \mathscr{F}$ satisfying all of the tests in Ψ.*

We extend $\mathscr{A}$ in the obvious manner, i.e. by taking its low-degree-extension (see Definition 8) to the variables $\hat{V}_\Phi$, and then repeatedly taking the embedding extension of the previous-level variables, until we've assigned all of the variables in the system. More formally,

Proof. We construct an assignment $\mathscr{A}_\Psi : V_\Psi \to \mathscr{F}$ by inductively obtaining $[r_v, d]$-LDFs $P_v : \mathbf{dom}_v \to \mathscr{F}$ for every level-i node v of every tree in the CR-forest, as follows. We first set (for every $\varphi \in \Phi$) P_{root_φ} to be the low degree extension (see Definition 8) of $\mathscr{A}$ (we think of $\mathscr{A}$ as assigning each variable a value in $\{0,1\} \subset \mathscr{F}$ rather than $\{\texttt{true}, \texttt{false}\}$, see discussion in the beginning of Section 3). Assume we've defined an $[r_u, d]$-LDF P_u consistently for all level-i nodes, and let v be an offspring of u, labeled by $\mathscr{C}_v$. The restriction $f = P_u|_{\mathscr{C}_v}$ of P_u to the cube $\mathscr{C}_u$ is an $[r, D+2]$-LDF where $r = r_u$ or $r = r_u(D+2)$ depending on whether $\mathscr{C}_v$ is parallel to the axises or not. Proposition 12 says that f can be extended to a $[\sqrt[a]{r+1} - 1, a(D+2)]$-LDF f_{ext} over the larger domain $\mathscr{F}^d$ (recall that $d = a(D+2)$). We define $P_v = f_{\text{ext}}$ to be that $[r_v, d]$-LDF (recall that $b_v = \sqrt[a]{r+1}$ and $r_v = b_v - 1$).

Finally, for a variable $\mathtt{x} \in \mathbf{var}_v$, $\mathtt{x} = \mathbf{var}_v(x)$, we set $\mathscr{A}_\Psi(\mathtt{x}) \overset{\text{def}}{=} P_v(x)$. The construction implies that the assignment is well defined there are no collisions, i.e. whenever $\mathtt{x}' = \mathbf{var}_{v'}(x') = \mathbf{var}_v(x) = \mathtt{x}$ implies $P_v(x) = P_{v'}(x')$. □

4.2. Soundness

We need to show that a 'no' instance of SAT is mapped to a 'no' instance of $\mathscr{SSAT}_\infty$. We assume that the constructed $\mathscr{SSAT}_\infty$ instance has a consistent non-trivial super-assignment of norm $\leqslant g$, and show that Φ—the SAT instance we began with—is satisfiable.

Lemma 17 (soundness). *Let $g \overset{\text{def}}{=} |\mathscr{F}|^{1/102}$. If there exists a consistent super-assign ment of norm $\leqslant g$ for Ψ, then Φ is satisfiable.*

Let $\mathscr{A}$ be a consistent non-trivial super-assignment for Ψ, of size $\|\mathscr{A}\|_\infty \leqslant g$. It induces (by projection) a super-assignment to the variables

$$m : V_\Psi \to \mathbb{Z}^{|\mathscr{F}|}$$

i.e. for every variable $\mathtt{x} \in V_\Psi$, m assigns a vector $\pi_{\mathtt{x}}(\mathscr{A}(\psi))$ of integer coefficients, one per value in $\mathscr{F}$ where ψ is some test depending on $\mathtt{x}$. Since $\mathscr{A}$ is consistent, m is well defined (independent of the choice of test ψ). Alternatively, we view m as a labeling of the points $\bigcup_{v \in \mathbf{F}_n(\Phi)} \mathbf{dom}_v$ by a 'super-value'—a formal integer linear combination of values from $\mathscr{F}$. The label of the point $x \in \mathbf{dom}_v$ for some $v \in \mathbf{F}_n(\Phi)$, is simply $m(\mathbf{var}_v(x))$, and with a slight abuse of notation, is sometimes denoted $m(x)$. m is used as the "underlying point super-assignment" for the rest of the proof, and will serve as an anchor by which we test consistency.

The central task of our proof is to show that if a tree has a non-trivial leaf, then there is a non-trivial super-LDF for the domain in the root node that is consistent with m. We will later want to construct from these super-LDFs an assignment that satisfies all of the tests in Φ. For this purpose, we need the super-LDFs along the way to be legal.

Definition 18 (*legal*). An LDF P is called *legal for a node* $v \in \mathbf{T}_\varphi$ (for some $\varphi \in \Phi$), if it satisfies φ in the sense that if φ's variables have pre-images (under the mapping $\mathbf{var}_v : \mathbf{dom}_v \to V_\Psi$) $x_1, \ldots, x_D \in \mathbf{dom}_v$, then $P(x_1), \ldots, P(x_D)$ satisfy φ. A super-LDF $\mathscr{G}$ is called *legal for* $v \in \mathbf{T}_\varphi$ if for every LDF P appearing in $\mathscr{G}$, P is legal for $v \in \mathbf{T}_\varphi$.

The following lemma encapsulates the key inductive step in our soundness proof.

Lemma 19. *Let u be a level-i node for some $0 \leqslant i < K$. There is a legal super-$[r_u, d]$-LDF $\mathscr{G}_u$ with $\|\mathscr{G}_u\|_1 \leqslant \|m\|_\infty \overset{\text{def}}{=} \max_x \|m(x)\|_1$ such that for every $x \in \mathbf{dom}_u$, $\pi_x(\mathscr{G}_u) = m(x)$. Furthermore, if there is a node v in u's sub-tree for which $\mathscr{G}_v \neq \vec{0}$ then $\mathscr{G}_u \neq \vec{0}$.*

Proof. We prove the lemma for $i < K$ by induction on $K - i$. For nodes in level $i = K$ (or any other leaf) the lemma follows by setting $\mathscr{G}_u \overset{\text{def}}{=} \mathscr{A}(\psi_u)$.

Let $0 < i < K$, let u be a level-i node, and assume (by induction) that for every offspring v of u there is a legal super-$[r_v, d]$-LDF $\mathscr{G}_v$ over $\mathbf{dom}_v$, with $\|\mathscr{G}_v\|_1 \leqslant \|m\|_\infty$ such that

$$\forall x \in \mathbf{dom}_v \quad \pi_x(\mathscr{G}_v) = m(x).$$

For each such v, let $\tilde{\mathscr{G}}_v$ be the super-$[(b_v)^a - 1, D+2]$-LDF that is the restriction of $\mathscr{G}_v$ to the manifold $E_{b_v}(\mathscr{C}_v)$ as defined in Section 2.2. Note that the degree of $\tilde{\mathscr{G}}_v$ is $(b_v)^a - 1 = r_u$ when $\mathscr{C}_v$ is parallel to the axises, and $(b_v)^a - 1 \leqslant r_u(D+2)$ in any other case. Thus the super-LDFs $\tilde{\mathscr{G}}_v$ have *total* degree $\leqslant (D+2)^2 r_u$. We know (see Proposition 13 from Section 2.2) that if $\mathscr{G}_v \neq \vec{0}$ then $\tilde{\mathscr{G}}_v \neq \vec{0}$. □

The following consistency lemma will imply the existence of a super-LDF $\mathscr{G}_u$ for u with the desired consistency property.

Lemma 20 (Dinur et al. [9]). *Let $\alpha < \frac{1}{100}$ be an arbitrary positive constant. Let u be a level-i node for some $0 \leqslant i < K$. If for every offspring v of u there is a super-LDF $\tilde{\mathscr{G}}_v$ over $\mathscr{C}_v$, of total degree $\leqslant r$ and norm $\|\tilde{\mathscr{G}}_v\|_1 \leqslant s$, such that*

$$\Pr_{x \in \mathscr{C}_v} (\pi_x(\tilde{\mathscr{G}}_v) = m(x)) \geqslant 1 - \alpha,$$

then there is a super-LDF $\mathscr{G}_u$ over $\mathbf{dom}_u$ of total degree r and norm $\|\mathscr{G}_u\|_1 \leqslant 2s$ that obeys

$$\Pr_{\mathscr{C}_v}(\pi_{\mathscr{C}_v}(\mathscr{G}_u) = \tilde{\mathscr{G}}_v) \geqslant 1 - \alpha.$$

(This lemma is the Consistency Lemma (Lemma 7) from [9]. It is stated there only for ‘good’ nodes u and v, however all nodes are ‘good’ in our context because we are dealing with l_∞ norm rather than l_1.).

We apply this lemma taking $s = \|m\|_\infty$ and $r \overset{\text{def}}{=} (D+2)^2 r_u$. Note that in our case the inductive assumption gives

$$\forall \mathscr{C}_v \in \mathbf{labels}(u), \quad \Pr_{x \in \mathscr{C}_v} (\pi_x(\tilde{\mathscr{G}}_v) = m(x)) = 1.$$

Thus we obtain a super-LDF $\mathscr{G}_u$ over $\mathbf{dom}_u$ of total-degree r. Although we only obtain consistency of $\mathscr{G}_u$ on most points $x \in \mathscr{F}^d$, we next claim that consistency follows in fact for *all* points. Let $\mathrm{N} = \{x \in \mathscr{F}^d \mid \pi_x(\mathscr{G}_u) \neq m(x)\}$ be the set of inconsistent points. For the sake of contradiction assume $\mathrm{N} \neq \phi$, and let $x_0 \in \mathrm{N}$. Consider any cube $\mathscr{C}_v \in \mathbf{labels}(u)$ that contains x_0. We have $\pi_{x_0}(\tilde{\mathscr{G}}_v) = m(x_0) \neq \pi_{x_0}(\mathscr{G}_u)$, so $\pi_{\mathscr{C}_v}(\mathscr{G}_u) \neq \tilde{\mathscr{G}}_v$, therefore the super-LDF $\pi_{\mathscr{C}_v}(\mathscr{G}_u) - \tilde{\mathscr{G}}_v$ (subtraction is defined as subtraction of two vectors in $\mathbb{Z}^{|\mathrm{LDF}_{r,D+2}|}$) is non-trivial. Proposition 10 (low-ambiguity), when applied to $\pi_{\mathscr{C}_v}(\mathscr{G}_u) - \tilde{\mathscr{G}}_v$ implies that for *almost all* points $x \in \mathscr{C}_v$, $\pi_x(\mathscr{G}_u) \neq \pi_x(\tilde{\mathscr{G}}_v) = m(x)$, so these points are also in N. A simple geometric argument shows that the distribution of choosing a $(D+2)$-cube $\mathscr{C}$ containing x_0, and then choosing a random point $x \in_R \mathscr{C}$ is very close to uniformly choosing a point $x \in_R \mathscr{F}^d$. We saw that a point chosen in this manner has high probability of being in N, thus N consists of (much more than) half of the points in $\mathscr{F}^d$. The fraction of $(D+2)$-cubes that don’t hit a point in N is (by another simple geometric argument, relying on the fact that N is large enough) very small, and in particular, less than $1-\alpha$. Thus by Lemma 20 there is a cube $\mathscr{C}_v$ for which $\pi_{\mathscr{C}_v}(\tilde{\mathscr{G}}_v) = \tilde{\mathscr{G}}_v$ with $\exists x_1 \in \mathrm{N} \cap \mathscr{C}_v$ and so $\pi_{x_1}(\mathscr{G}_u) = \pi_{x_1}(\tilde{\mathscr{G}}_v) = m(x_1)$, a contradiction to $x_1 \in \mathrm{N}$. Thus $\mathrm{N} = \phi$, or

$$\forall x \in \mathscr{F}^d \quad \pi_x(\mathscr{G}_u) = m(x).$$

We now turn to establish the legality of $\mathscr{G}_u$. For this we need to show that every LDF f appearing in $\mathscr{G}_u$ is legal, i.e. if φ is the test for which $u \in \mathbf{T}_\varphi$ and if φ’s variables appear in $\mathbf{var}_u(\mathbf{dom}_u)$, then f satisfies φ (see also Definition 18). First note that the equality $\pi_{\mathscr{C}_v}(\mathscr{G}_u) = \tilde{\mathscr{G}}_v$ actually holds for *every* offspring v of u (this follows instantly from the low ambiguity property, and because we already know that for any $x \in \mathscr{C}_v \subset \mathbf{dom}_u$, $\pi_x(\tilde{\mathscr{G}}_v) = \pi_x(\mathscr{G}_u)$). Now suppose indeed φ’s variables appear in $\mathbf{var}_u(\mathbf{dom}_u)$, and consider the $(D+2)$-cubes $\mathscr{C}_v \subset \mathbf{dom}_u$ (respectively, the offsprings v of u) that contain the D points $x_1, \ldots, x_D$ corresponding to these variables. Let $x_0 \in \mathscr{F}^d$ be a non-ambiguous point for $\mathscr{G}_u$ (most points qualify). It follows that x_0 is non-ambiguous for $\pi_{\mathscr{C}_v}(\mathscr{G}_u) = \tilde{\mathscr{G}}_v$ where $\mathscr{C}_v$ is a $(D+2)$-cube containing x_0 and $x_1, \ldots, x_D$. For every LDF f appearing in $\mathscr{G}_u$, its restriction $f|_{\mathscr{C}_v}$ appears in $\tilde{\mathscr{G}}_v$ which is a legal super-LDF (because $\mathscr{G}_v$ is legal by the inductive assumption). Hence f is legal, making $\mathscr{G}_u$ legal.

In addition, we claim that any LDF f appearing in $\mathscr{G}_u$, is of degree r_u rather than $r = (D+2)^2 r_u$. This follows by considering the set of cubes parallel to the axises in which f appears. The super-LDFs $\tilde{\mathscr{G}}_v$ over these cubes are of degree r_u. Proposition 9 (along with previously noted fact that $\pi_{\mathscr{C}_v}(\mathscr{G}_u) = \tilde{\mathscr{G}}_v$ for every v) thus implies that f is an $[r_u, d]$-LDF as claimed, and it makes sense to say that $\mathscr{G}_u$ is a super-$[r_u, d]$-LDF.

Finally, if $\mathscr{G}_v \neq \vec{0}$ for some offspring v of u, then $\tilde{\mathscr{G}}_v \neq \vec{0}$ because of Proposition 13 and since $\pi_{\mathscr{C}_v}(\mathscr{G}_u) = \tilde{\mathscr{G}}_v$, we deduce $\mathscr{G}_u \neq \vec{0}$. By Proposition 10 (low-ambiguity) for most points $\|m(x)\|_1 = \|\mathscr{G}_u\|_1$, so obviously $\|\mathscr{G}_u\|_1 \leqslant \max_x \|m(x)\|_1 = \|m\|_\infty$.

This completes the proof of Lemma 19. □

In order to complete the soundness proof, we need to find a satisfying assignment for Φ. We obtained, in Lemma 19, a super-$[r_0, d]$-LDF $\mathscr{G}_\varphi$ for each root node root_φ, such that $\forall x \in \mathbf{dom}_{\mathrm{root}_\varphi} = \mathscr{F}^{d_0}$, $m(x) = \pi_x(\mathscr{G}_\varphi)$. Note that indeed, for every pair of tests $\varphi \neq \varphi'$, the corresponding super-LDFs must be equal $\mathscr{G}_\varphi = \mathscr{G}_{\varphi'}$ (denote them by $\mathscr{G}$). This follows because they are point-wise equal $\pi_x(\mathscr{G}_\varphi) = m(x) = \pi_x(\mathscr{G}_{\varphi'})$, and so the difference super-LDF $\mathscr{G}_\varphi - \mathscr{G}_{\varphi'}$ is trivial on every point, and must therefore (again, by Proposition 10-low-ambiguity) be trivial.

If $\mathscr{A}$ is not trivial, then there is at least one test $\psi_v \in \Psi$ for which $\mathscr{A}(\psi_v) \neq \vec{0}$. Thus, denoting by φ the test for which v is a leaf in $\mathbf{T}_\varphi$, Lemma 19 implies $\mathscr{G} = \mathscr{G}_\varphi \neq \vec{0}$. Take an LDF f that appears in $\mathscr{G}$, and define for every $v \in V_\Phi$, $\mathscr{A}(v) \overset{\mathrm{def}}{=} f(x)$ where $x \in \mathscr{H}^{d_0}$ is the point mapped to v. Since $\mathscr{G}$ is legal, Φ is totally satisfied by $\mathscr{A}$.

5. From $\mathscr{SSAT}_\infty$ to SVP$_\infty$

In this section, we show the reduction from g-$\mathscr{SSAT}_\infty$ to the problem of approximating SVP$_\infty$. This reduction follows the same lines of the reduction in [3] from Pseudo-Label-Cover to SVP$_\infty$. We begin by formally defining the gap-version of SVP$_\infty$ (presented in Section 1) which is the standard method to turn an approximation problem into a decision problem.

Definition 21 (g-SVP_∞). Given a lattice $\mathscr{L}$ and a number $d > 0$, distinguish between the following two cases:
Yes. There is a non-zero lattice vector $v \in \mathscr{L}$ with $\|v\|_\infty \leqslant d$.
No. Every non-zero lattice vector $v \in \mathscr{L}$ has $\|v\|_\infty > gd$.

We will show a reduction from g-$\mathcal{SSAT}_\infty$ to $\sqrt{g}$-SVP$_\infty$, thereby implying SVP$_\infty$ to be NP-hard to approximate to within a factor of $\sqrt{g} = n^{\Omega(1)/\log\log n}$.

Let $I = \langle \Psi, V, \{\mathcal{R}_\psi\}\rangle$ be an instance of g-$\mathcal{SSAT}_\infty$, where $\Psi = \{\psi_1, \ldots, \psi_n\}$ is a set of tests over variables $V = \{v_1, \ldots, v_m\}$, and $\mathcal{R}_{\psi_i}$ is the set of satisfying assignments for $\psi_i \in \Psi$. We construct a $\sqrt{g}$-SVP$_\infty$ instance $(\mathcal{L}(B), d)$ where $d \overset{\text{def}}{=} 1$ and B is an integer matrix whose columns form the basis for the lattice $\mathcal{L}(B)$.

The matrix B will have a column $\vec{v}_{[\psi,r]}$ for every pair of test $\psi \in \Psi$ and an assignment $r \in \mathcal{R}_\psi$ for it. There will be one additional special column $\vec{t}$. The matrix B will have two kinds of rows, consistency rows and norm-measuring rows, defined as follows.

Consistency rows. B will have $|\mathcal{F}|+1$ rows for each threesome (ψ_i, ψ_j, x) where ψ_i and ψ_j are tests that depend on a mutual variable x. Only the columns of ψ_i and ψ_j will have non-zero values in these rows.

The special column $\vec{t}$ will have $\sqrt{g}$ in each consistency row, and zero in the other rows.

For a pair of tests ψ_i and ψ_j that depend on a mutual variable x, let's concentrate on the sub-matrix consisting of the columns of these tests, and the $|\mathcal{F}|+1$ rows of the threesome $\langle \psi_i, \psi_j, x\rangle$ viewed as a pair of matrices G_1 of dimension $(|\mathcal{F}|+1) \times |\mathcal{R}_{\psi_i}|$ and G_2 of dimension $(|\mathcal{F}|+1) \times |\mathcal{R}_{\psi_j}|$. Let $r \in \mathcal{R}_{\psi_i}$ be a satisfying assignment for ψ_i and $r' \in \mathcal{R}_{\psi_j}$ be a satisfying assignment for ψ_j. The rth column in G_1 equals $\sqrt{g}$ times the unit vector e_i where $i = r|_x$ (i.e. a vector with zeros everywhere and a $\sqrt{g}$ in the $r|_x$th coordinate). The r'th column in G_2 equals $\sqrt{g}(\vec{1} - e_i)$ where $i = r'|_x$ and $\vec{1}$ is the all-one vector (i.e. $\sqrt{g}$ everywhere except a zero in the $r'|_x$th coordinate).

Notice that any zero-sum linear combination of the vectors $\{e_i, \vec{1} - e_i, \vec{1}\}_{i=1\ldots|\mathcal{F}|}$ must give e_i the same coefficient as $\vec{1} - e_i$, because the vectors $\{\vec{1}, e_i\}_i$ are linearly independent (notice we are looking at vectors with $|\mathcal{F}|+1$ coordinates).

Norm-measuring rows. There will be a set of $\mathcal{R}_\psi$ rows designated to each test $\psi \in \Psi$ in which only ψ's columns have non-zero values. The matrix B, when restricted to these rows and to the columns of ψ, will be the $(|\mathcal{R}_\psi| \times |\mathcal{R}_\psi|)$ Hadamard matrix $\mathbf{H}$ (we assume for simplicity that $|\mathcal{R}_\psi|$ is a power of 2, thus such a matrix exists, see [5], p. 74). Recall that the Hadamard matrix $\mathbf{H}_n$ of order $2^n \times 2^n$ is defined by $\mathbf{H}_0 = (1)$ and $\mathbf{H}_n = \left(\begin{smallmatrix} \mathbf{H}_{n-1} & \mathbf{H}_{n-1} \\ \mathbf{H}_{n-1} & -\mathbf{H}_{n-1}\end{smallmatrix}\right)$.

The vector $\vec{t}$, as mentioned earlier, will be zero on these rows. (Fig. 1.)

Proposition 22 (completeness). *If there is a natural assignment to Ψ, then there is a non-zero lattice vector $\vec{v} \in \mathcal{L}(B)$ with $\|\vec{v}\|_\infty = 1$.*

Proof. Let $\mathcal{A}$ be a consistent natural assignment for Ψ. We claim that

$$\vec{v} = \vec{t} - \sum_{\psi \in \Psi} \vec{v}_{[\psi, \mathcal{A}(\psi)]}$$

is a lattice vector with $\|\vec{v}\|_\infty = 1$. Restricting $\sum_{\psi \in \Psi} \vec{v}_{[\psi, \mathcal{A}(\psi)]}$ to an arbitrary row in the consistency rows (corresponding to a pair of tests ψ_i, ψ_j with mutual variable x), gives

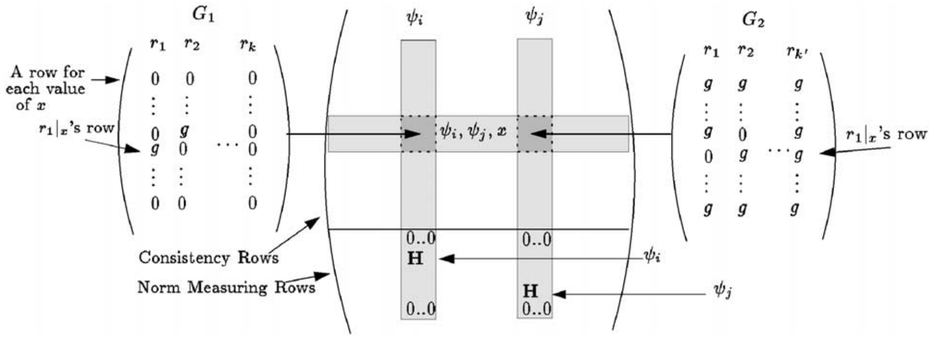

Fig. 1. The matrix B.

$\sqrt{g}$, because $\mathscr{A}(\psi_i)|_x = \mathscr{A}(\psi_j)|_x$. Subtracting this from $\vec{t}$ gives zero in each consistency-row.

In the norm-measuring rows, since every test $\psi \in \Psi$ is assigned one value by $\mathscr{A}$, $\vec{v}$ restricted to ψ's rows equals some column of the Hadamard matrix which is a ± 1 matrix. Altogether, $\|\vec{v}\|_\infty = 1$ as claimed. □

Proposition 23 (soundness). *If there is a non-zero lattice vector $\vec{v} \in \mathscr{L}(B)$ with $\|\vec{v}\|_\infty < \sqrt{g}$, then there is a consistent non-trivial super-assignment $\mathscr{A}$ for Ψ, for which $\|\mathscr{A}\|_\infty < g$.*

Proof. Let

$$\vec{v} = c_t \cdot \vec{t} + \sum_{\psi, r} c_{[\psi, r]} \cdot \vec{v}_{[\psi, r]}$$

be a lattice vector with $\|\vec{v}\|_\infty < \sqrt{g}$. The entries in the consistency rows of every lattice vector, are integer multiples of $\sqrt{g}$. The assumption $\|\vec{v}\|_\infty < \sqrt{g}$ implies that v is zero on these rows.

Define a super-assignment $\mathscr{A}$ to Ψ by setting for each $\psi \in \Psi$ and $r \in \mathscr{R}_\psi$, $\mathscr{A}(\psi)[r] \stackrel{\text{def}}{=} c_{[\psi, r]}$.

To see that $\mathscr{A}$ is consistent, let $\psi_i, \psi_j \in \Psi$ both depend on the variable x. Notice that (as mentioned above) any zero-sum linear combination of the vectors $\{\vec{1}, e_k, \vec{1} - e_k\}_k$ must give e_k and $\vec{1} - e_k$ the same coefficient because the vectors $\{\vec{1}, e_k\}_k$ are linearly independent (notice we are looking at $k+1$-coordinate vectors). This implies that for any value $k \in \mathscr{F}$ for x,

$$\sum_{r|_x = k} c_{[\psi_i, r]} = \sum_{r'|_x = k} c_{[\psi_j, r']}.$$

This, in turn, means that $\pi_x(\mathscr{A}(\psi_i)) = \pi_x(\mathscr{A}(\psi_j))$ thus $\mathscr{A}$ is consistent.

$\mathscr{A}$ is also not-all-zero because $\vec{v} \neq \vec{0}$ (if only c_t was non-zero, then $\|\vec{v}\|_\infty = \sqrt{g}$). The norm of $\mathscr{A}$ is defined as

$$\|\mathscr{A}\|_\infty = \max_{\psi \in \Psi}(\|\mathscr{A}(\psi)\|_1).$$

The vector $\vec{v}$ restricted to the norm-measuring rows of ψ is exactly $\mathbf{H}\mathscr{A}(\psi)$. Now since $1/\sqrt{|\mathscr{R}_\psi|}\mathbf{H}$ is a $(|\mathscr{R}_\psi| \times |\mathscr{R}_\psi|)$ orthonormal matrix, we have

$$\left\| \frac{1}{\sqrt{|\mathscr{R}_\psi|}} \mathbf{H}\mathscr{A}(\psi) \right\|_2 = \|\mathscr{A}(\psi)\|_2.$$

Since for every $z \in \mathbb{R}^n$, $\|z\|_\infty \geqslant \|z\|_2/\sqrt{n}$, we obtain $\|\mathbf{H}\mathscr{A}(\psi)\|_\infty \geqslant \|\mathscr{A}(\psi)\|_2$. Now for every integer vector z, $\sqrt{\|z\|_1} \leqslant \|z\|_2$, and altogether,

$$\sqrt{\|\mathscr{A}(\psi)\|_1} \leqslant \|\mathscr{A}(\psi)\|_2 \leqslant \|\mathbf{H}\mathscr{A}(\psi)\|_\infty \leqslant \|v\|_\infty < \sqrt{g}$$

showing $\|\mathscr{A}\|_\infty \overset{\text{def}}{=} \max_{\psi \in \Psi} \|\mathscr{A}(\psi)\|_1) < g$ as claimed. □

Finally, if Ψ is a $\mathscr{SSAT}_\infty$ no instance, then the norm of any consistent super-assignment $\mathscr{A}$ must be at least g, and so the norm of the shortest lattice vector in $\mathscr{L}(B)$, must be at least g. This completes the proof of the reduction.

The reduction to CVP_∞ is quite similar, taking $\vec{t}$ to be the target vector, and is omitted.

References

[1] M. Ajtai, Generating hard instances of lattice problems, Proc. of the 28th ACM Symp. on Theory of Computing, 1996, pp. 99–108.

[2] M. Ajtai, The shortest vector problem in *L2* is NP-hard for randomized reductions, Proc. of the 30th Annual ACM Symp. on Theory of Computing (STOC-98), May 23–26 1998, ACM Press, New York, pp. 10–19.

[3] S. Arora, L. Babai, J. Stern, Z. Sweedyk, The hardness of approximate optima in lattices, codes and linear equations, Proc. of the 34th IEEE Symp. on Foundations of Computer Science, 1993, pp. 724–733.

[4] L. Babai, On Lovász's lattice reduction and the nearest lattice point problem, Combinatorica 6 (1986) 1–14.

[5] B. Bollobás, Combinatorics, Cambridge University Press, Cambridge, 1986.

[6] J.Y. Cai, A. Nerurkar, Approximating the SVP to within a factor $(1 + 1/\dim^\varepsilon)$ is NP-hard under randomized reductions, Proc. of the 13th Annual IEEE Conference on Computational Complexity, 1998, pp. 46–55.

[7] S. Cook, The complexity of theorem-proving procedures, Proc. of the Third ACM Symp. on Theory of Computing, 1971, pp. 151–158.

[8] I. Dinur, E. Fischer, G. Kindler, R. Raz, S. Safra, PCP characterizations of NP: Towards a polynomially-small error-probability, STOC: ACM Symp. on Theory of Computing (STOC), 1999.

[9] I. Dinur, G. Kindler, R. Raz, S. Safra, Approximating-CVP to within almost-polynomial factors is NP-hard, manuscript, 1999.

[10] I. Dinur, G. Kindler, S. Safra, Approximating-CVP to within almost-polynomial factors is NP-hard. FOCS: IEEE Symp. on Foundations of Computer Science (FOCS), 1998.

[11] A. Frank, É. Tardos, An application of simultaneous approximation in combinatorial optimization, 26th Annual Symp. on Foundations of Computer Science, Portland, Oregon, 21–23 October 1985, IEEE Press, New York, pp. 459–463.

[12] O. Goldreich, S. Goldwasser, On the limits of non-approximability of lattice problems. Proc. of the 30th ACM Symp. on Theory of Computing, 1998, pp. 1–9.

[13] J.C. Lagarias, A.M. Odlyzko, Solving low-density subset sum problems, J. ACM 32 (1) (1985) 229–246.

[14] A.K. Lenstra, H.W. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, Math. Ann. 261 (1982) 513–534.

[15] L. Levin, Universal'nyĭe perebornyĭe zadachi universal search problems, Problemy Peredachi Informatsii 9 (3) (1973) 265–266 (in Russian).

[16] D. Micciancio, The shortest vector in a lattice is hard to approximate to within some constant, Proc. of the 39th IEEE Symp. on Foundations of Computer Science, 1998.

[17] C.P. Schnorr, A hierarchy of polynomial-time basis reduction algorithms, Proceedings of Conference on Algorithms, Pécs (Hungary), North-Holland, Amsterdam, 1985, pp. 375–386.