On the (Classical and Quantum) Fine-Grained Complexity of Approximate CVP and Max-Cut

Jeremy Ahrens Huang
Department of Computer Science and Engineering,
Pennsylvania State University, University Park, PA, USA

Young Kun Ko
Department of Computer Science and Engineering,
Pennsylvania State University, University Park, PA, USA

Chunhao Wang
Department of Computer Science and Engineering,
Pennsylvania State University, University Park, PA, USA

Abstract

We show a linear-size reduction from gap Max-2-Lin(2) (a generalization of the approximate Maximum Cut, or gap Max-Cut, problem) to γ-CVP$_p$ for $\gamma = \mathrm{O}(1)$ and finite $p \geq 1$, as well as a no-go theorem against poly-sized non-adaptive quantum reductions from k-SAT to CVP$_2$. This implies three headline results:

(i) Faster algorithms for γ-CVP$_p$ are also faster algorithms for Max-2-Lin(2) and Max-Cut. Depending on the approximation regime, even a $2^{0.78n}$-time or $2^{0.3n}$-time algorithm would improve upon state-of-the-art algorithms such as Williams' 2004 algorithm [*TCS* 2005] or Arora, Barak, and Steurer's 2010 algorithm [*JACM* 2015]. This provides evidence that γ-CVP$_p$ for $\gamma = \mathrm{O}(1)$ requires exponential time, improving upon the previous exponential lower-bound for γ-CVP$_2$ with $\gamma < 3$ by Bennett, Golovnev, and Stephens-Davidowitz [*FOCS* 2017].

(ii) A new almost $2^{(1/2+\varepsilon/4\varsigma+o(1))n}$-time classical algorithm and a new almost $2^{(1/3+\varepsilon/6\varsigma+o(1))n}$-time quantum algorithm for $(1-\varepsilon, 1-\varsigma)$-gap Max-Cut. This algorithm is faster than the algorithm of Arora, Barak and Steurer [*JACM* 2015], as well as the algorithm of Williams [*TCS* 2005], and the algorithm of Manurangsi and Trevisan [*APPROX* 2018] when $c_0\varepsilon < \varsigma < c_1\varepsilon$ for constants c_0, c_1.

(iii) If the Quantum Strong Exponential Time Hypothesis (QSETH) can be used to show a $2^{\delta n}$-time lower-bound for Max-Cut, Max-2-Lin(2), or CVP$_2$ for any constant $\delta > 0$, it must be via an adaptive quantum reduction unless NP $\subseteq$ pr-QSZK. This illuminates some difficulties in characterizing the hardness of approximate constraint satisfaction problems and shows that the post-quantum security of lattice-based cryptography likely cannot be supported by QSETH. This result complements the no-go results of Aggarwal and Kumar [*FOCS* 2023], who showed that the classical security of lattice-based cryptography likely cannot be supported by the classical Strong Exponential Time Hypothesis (SETH).

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Approximation algorithms analysis; Theory of computation → Computational complexity and cryptography; Theory of computation → Quantum complexity theory

Keywords and phrases fine-grained complexity, instance compression, quantum algorithms, approximation algorithms, CVP, Max-Cut, Min-UnCut, Max-2-Lin, approximation-preserving reductions

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.111

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2411.04124` [21]

Funding *Jeremy Ahrens Huang*: National Science Foundation grant CCF-2238766 (CAREER).
Chunhao Wang: National Science Foundation grant CCF-2238766 (CAREER).

Acknowledgements We thank Sean Hallgren for valuable discussions and feedback.

53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).
Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;
Article No. 111; pp. 111:1–111:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The Approximate Closest Vector Problem (γ-CVP) is a hard approximate problem whose asymptotic time complexity $T_{\text{CVP},\gamma}(n)$ is not well understood for approximation parameters $3 \leqslant \gamma < 2^n$. The Approximate Minimum Un-Cut Problem ((ε, ς)-gap Min-UnCut), also known as the Approximate Maximum Cut Problem, is another hard approximate problem whose asymptotic time complexity $T_{\text{MC},\varepsilon,\varsigma}(n)$ is not well understood for approximation parameters $\varepsilon < \varsigma \leqslant \varepsilon^2$. We show a reduction from (ε, ς)-gap Min-UnCut to γ-CVP with a remarkable combination of properties: the reduction precisely preserves both the size and the approximation factor of any instance given to it. By possessing both properties at once, our reduction establishes a tight relationship between the time complexity of γ-CVP and of (ε, ς)-gap Min-UnCut in the approximation regimes of interest. As we will explain in Section 1.6, finding a reduction possessing both properties at once was a longstanding technical hurdle in the way of establishing such a relationship.

Specifically, our main theorem (Theorem 1.1) states that there is a reduction from (ε, ς)-gap Min-UnCut of size n to $\sqrt[p]{\varsigma/\varepsilon}$-CVP$_p$ of size n (where $p \geqslant 1$ is a constant parameter from CVP). This implies that

$$T_{\text{MC},\varepsilon,\varsigma}(n) \leqslant T_{\text{CVP},\gamma}(n)$$

whenever the following (simplified) inequality between the approximation parameters is satisfied:

$$\gamma \leqslant \sqrt[p]{\varsigma/\varepsilon}.$$

We use Theorem 1.1 to make three major advancements in the time complexity of CVP and Min-UnCut:

First, Theorem 1.1 directly gives new exponential-time conditional lower bounds (CLBs) on γ-CVP for $\gamma = \mathrm{O}(1)$ in both the classical and quantum settings. These CLBs are a win-win-win situation: either there are very strong lower bounds on γ-CVP, or there are extremely exciting new algorithms for (ε, ς)-gap Min-UnCut, or there are both strong lower bounds on γ-CVP and exciting new algorithms for (ε, ς)-gap Min-UnCut. This win-win-win situation is the defining property of fine-grained reductions (see Section 1.7 and the full version for more on fine-grained complexity). Previous exponential-time CLBs were only for CVP with $\gamma < 3$ or $\gamma = 1 + \mathrm{O}(1/\mathrm{poly}(n))$ in the classical setting from the fine-grained reductions of [11, 3].

Second, in combination with a new quantum algorithm presented in this paper and a classical algorithm from [25] for special instances of γ-CVP, Theorem 1.1 implies two new algorithms for (ε, ς)-gap Min-UnCut, one classical and one quantum, which we present in the full version. Our classical algorithm is faster than both the fastest exact algorithm [33] and the previous fastest approximation algorithm [26][1] for all instances where $c_1\varepsilon \geqslant \varsigma > c_0\varepsilon$ for some constants c_1, c_0. Our quantum algorithm is the first to beat Grover search for (ε, ς)-gap Min-UnCut!

Third, in combination with a new no-go theorem against *quantum* reductions from k-SAT to CVP presented in this paper and the no-go theorems against classical reductions in [5], we show that time-complexity lower bounds conditioned on k-SAT for Min-UnCut, including

[1] Note that a Min-UnCut algorithm is listed as an open question in the published version of this article and is present in the arXiv version which was submitted later.

exact weighted Min-UnCut, are unlikely in *both* the quantum and classical settings, assuming no complexity-theoretic disasters occur, for reasons we will explain in Section 1.4. This result could help explain why some desired results in the complexity of approximate problems, such as a proof of the Unique Games Conjecture, have been difficult to obtain.

Next we will define γ-CVP and (ε, ς)-gap Min-UnCut, state our main theorem, and then we will explain each advancement and its related work in more detail.

1.1 γ-CVP, Min-UnCut, and our main theorem

First we will define γ-CVP. Given a lattice $\mathcal{L}$, a target vector $\boldsymbol{t}$, and a distance r, γ-CVP is the problem of deciding if $\mathcal{L}$ is r close to $\boldsymbol{t}$. If $\mathcal{L}$ is not r close to $\boldsymbol{t}$, then it is promised to be more than γr far from $\boldsymbol{t}$. A lattice $\mathcal{L} \subset \mathbb{R}^d$ is a set of all the linear combinations of linearly independent basis vectors $\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(n)} \in \mathbb{R}^d$ with integer coefficients,

$$\mathcal{L} = \mathcal{L}(\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(n)}) := \left\{ \sum_{i=1}^{n} c_i \boldsymbol{b}^{(i)} | c_i \in \mathbb{Z} \right\}.$$

Here n is known as the rank of the lattice $\mathcal{L}$ and d is known as the ambient dimension of the lattice $\mathcal{L}$; they are main parameters governing the input size of γ-CVP. The usual way we specify a lattice $\mathcal{L}$ in the input to γ-CVP is by a choice of basis vectors for $\mathcal{L}$, which we represent as a matrix $B := \begin{pmatrix} | & & | \\ \boldsymbol{b}^{(1)} & \cdots & \boldsymbol{b}^{(n)} \\ | & & | \end{pmatrix}$ that relates every lattice point $\boldsymbol{v} \in \mathcal{L}(B)$ to its coordinates $\boldsymbol{y} \in \mathbb{Z}^n$ by $\boldsymbol{v} = B\boldsymbol{y}$. The distance $\mathrm{dist}(\mathcal{L}, \boldsymbol{t})$ from a lattice $\mathcal{L}$ to a target $\boldsymbol{t}$ is given by the distance of the closest point in $\mathcal{L}$ to $\boldsymbol{t}$ in the ℓ_p-norm. From now on we will call the problem γ-CVP$_p$ and the distance function dist_p to emphasize that different ℓ_p-norms give us different problems. γ-CVP$_p$ is known to be NP-complete for $\gamma \leqslant n^{c/\log\log n}$ (where $0 < c < \frac{1}{2}$ is a constant) [14], to NOT be NP-complete (unless P=NP) for all $\gamma > \sqrt{n}$ [7], and to be in P for $\gamma \geqslant 2^n$ [30].

Next we will define (ε, ς)-gap Min-UnCut. Given a weighted list of pairs between binary variables and two constants $0 \leqslant \varepsilon < \varsigma \leqslant 1$, (ε, ς)-gap Min-UnCut is the problem of deciding if there is an assignment to the variables such that at most a ε-fraction (by weight) of the variable pairs have equal values. If there is no such assignment, then it is promised that for every possible variable assignment at least a ς-fraction of pairs have equal values. This makes (ε, ς)-gap Min-UnCut a weighted 2-CSP (constraint satisfaction problem) where either at most an ε-fraction of the constraints are unsatisfiable or at least a ς-fraction of the constraints are unsatisfiable. Min-UnCut is a long-studied problem – the exact case was on Karp's original list of 21 NP-complete problems [22]. (ε, ς)-gap Min-UnCut is known to be NP-hard for constant $0 < \varepsilon < 1$ and $\varsigma \leqslant \sqrt{\varepsilon}$ under a widely accepted conjecture about the hardness of approximation [24], but in P when $\varepsilon \leqslant \frac{1}{\log n}$ or when $\varsigma > C\sqrt{\varepsilon}$ (for some constant C) [12, 13]. Note that CSPs are sometimes framed in terms of maximizing satisfied constraints instead of minimizing unsatisfied constraints, in which case the gap is given as the fraction of satisfied constraints. For example, $(1-\varepsilon, 1-\varsigma)$-gap Max-Cut is an alternate name which refers to the same problem as (ε, ς)-gap Min-UnCut.

Our main theorem actually works with a slight generalization of (ε, ς)-gap Min-UnCut known as $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2), which allows each constraint to specify whether the variables should be equal or unequal. Now we are ready to state our main theorem:

▶ **Theorem 1.1** (Main theorem). *There is a linear-time classical reduction and a linear-time quantum reduction from* $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2) *to* $\sqrt[p]{\varsigma/\varepsilon}$-CVP$_p$ *with* $p \in [1, \infty)$ *that uses at most one basis vector per variable.*

This reduction has the remarkable property that, for any fixed instance size, the approximation factor γ of the output instance increases *without limit* as the gap of the input instance increases. This means our reduction can produce γ-CVP_p instances of any size with any approximation factor! Previous reductions to γ-CVP_p had some inherent upper limit on the γ of the instances they can produce with n basis vectors for each n. See Section 1.6 for a summary of the difficulties involved in overcoming this longstanding limitation. Note also that our reduction maps $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2) instances with n variables to $\sqrt[p]{\varsigma/\varepsilon}$-$\mathrm{CVP}_p$ instances with *exactly* n basis vectors; this fact is critical for the tightness of our conditional time bounds on γ-CVP_p, which we will describe next. After that, we will describe our two other major advancements and their related works.

1.2 Our conditional lower bound for γ-CVP_p and related works

1.2.1 Previous work

The way that we show a conditional lower bound (CLB) for the time-complexity of γ-CVP_p is via a reduction that links the time-complexity of γ-CVP_p to the time complexity of a well-studied problem. This gives us a win-win situation: the lower bound on γ-CVP_p conditioned on an assumption about the well-studied problem holds unless there is a breakthrough on the well-studied problem. The assumption is necessary since without it we'd be proving that $\mathrm{P} \neq \mathrm{NP}$. Reductions from well-studied problems to γ-CVP_p have a long history: [32] showed a reduction to exact CVP_p and a sequence of works including [9, 15, 14, 20][2] gave reductions to γ-CVP_p for greater γ, with the greatest approximation factor ($\gamma = n^{c/\log\log n}$ for some constant $c > 0$) first achieved by [14]. These reductions showed that γ-CVP_p is NP-hard for $\gamma \leqslant n^{c/\log\log n}$.

However, these reductions only provide a very weak link from the time complexity of their well-studied problems to the time complexity of γ-CVP_p, because they produce instances of γ-CVP_p with too many basis vectors. A strong link would require a reduction that uses n basis vectors or less – even using $2n$ basis vectors would meaningfully weaken the link (see Section 1.7 for why). These works use $O(n^k)$ many basis vectors for some (often large and unspecified) constants $k > 1$, and the reductions in [9][3] for γ-CVP_p with super-constant γ even use super-polynomially many basis vectors! So at best, these results can establish a $2^{\sqrt[k]{n}}$-time conditional lower bound on γ-CVP_p. For practical values of n these bounds can be even weaker than they appear: at just $k = 33$, the inequality $2^{\sqrt[k]{n}} < n$ holds true for $1 < n \leqslant 10^{80}$ (that's the number of atoms in the visible universe).

The fine-grained complexity of γ-CVP_2 has recently gained the interest of cryptographers because breaking the lattice-based post-quantum cryptosystems [29, 28, 18] currently in widespread use [31] can be done by solving instances of γ-CVP_2. A small change in its quantum time complexity (e.g. from $2^{0.5n}$ to $2^{0.05n}$) can make cracking a secret meaningfully faster, and the previously discussed reductions can give very little evidence for or against such a change. This motivated [11] to show the first result in the fine-grained complexity of γ-CVP_p: a reduction from $(1-\varepsilon, 1-\varsigma)$-gap unweighted Max-2-SAT or unweighted Max-Cut on n variables to γ-CVP_p on n basis vectors for $p \in [1, \infty)$ where

$$\gamma = \sqrt[p]{\frac{1+\varsigma(3^p-1)}{1+\varepsilon(3^p-1)}} < 3. \tag{1}$$

[2] Note that while some of these reductions as written are to a slightly different problem γ-SVP, by [18] they are also reductions to γ-CVP_p and the same principles and difficulties apply.

[3] If reading [9], be aware that the problem they call "Label Cover" is very different from (and much easier than) the problem that Label Cover usually refers to today ([27]).

$(1-\varepsilon, 1-\varsigma)$-gap unweighted Max-2-SAT is a 2-CSP like $(1-\varepsilon, 1-\varsigma)$-gap unweighted Max-Cut, but with constraints that are satisfied when at *least* one variable is 1 instead of when *exactly* one of the two variables is 1. [11] also showed another reduction which was improved by [3] to go from k-SAT on n variables to almost exact CVP_p ($\gamma = 1 + 1/\mathrm{poly}(n)$) on n basis vectors, but only for non-even p. From now on we will refer to the former reduction as the [11] reduction and the latter reduction as the [3] reduction for the convenience of the reader. [11] also gave a reduction from exact Min-UnCut on n variables to exact CVP_p on n basis vectors.

While there have been no better reductions to γ-CVP_p with even p or larger γ since [11] and [3], there is a good reason why. [3] and [5] found that, unless the polynomial hierarchy collapses, any classical reduction to γ-CVP_p with even p or larger γ will likely need to both be from a problem dissimilar to k-SAT and use different techniques from the [11] and [3] reductions. We also find similar restrictions against such quantum reductions in Section 1.4. This is why our reduction is from Max-2-Lin(2) and uses different geometric gadgets to implement the Max-2-Lin(2) constraints.

We summarize the CLBs on γ-CVP_2 in Figure 1.

1.2.2 Our contribution

We show a reduction from $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2) on n variables to γ-CVP_p with n basis vectors for $p \in [1, \infty)$ with $\gamma = \sqrt[p]{\varsigma/\varepsilon}$ in both the quantum and classical settings. It allows us to give the general CLB on γ-CVP_p in Corollary 1.2 which follows directly from Theorem 1.1. It implies that if one believes a $2^{\delta n}$-time lower bound for some specific constant $\delta > 0$ for $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2) or (ε, ς)-gap Min-UnCut, then one must also believe a $2^{\delta n}$-time lower bound for $\sqrt[p]{\varsigma/\varepsilon}$-$\mathrm{CVP}_p$.

▶ **Corollary 1.2.** *Any $O(f(n))$-time classical algorithm for $\sqrt[p]{\varsigma/\varepsilon}$-$\mathrm{CVP}_p$, where $f(n)$ is lower-bounded by the dimension of the lattice and $p \in [1, \infty)$, is an $O(f(n))$-time algorithm for $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2). The same holds for quantum algorithms.*

Our reduction makes several improvements over the previous state-of-the-art fine-grained reduction to γ-CVP_p by [11]: it considers the quantum setting in addition to the classical setting, it works for both weighted and unweighted input instances, and there is no hard limit on how large γ is. Using our reduction, we can give interesting CLBs on γ-CVP_p for approximation factors up to $\gamma = O(1)$.

The increase in the range of approximation factors our reduction works for is very significant. Not only does this tell us something new about the time complexity of many γ-CVP_p problems that we did not previously have good CLBs for (since [11] only works for $\gamma < 3$), but it also applies a much broader scope of existing and future research on lattice problems to the gap CSPs Min-UnCut and Max-2-Lin(2). For example, while our reduction gives us new faster classical and quantum algorithms for (ε, ς)-gap Min-UnCut and Max-2-Lin(2) in Section 1.3 using algorithms for general $O(1)$-CVP_p, these improvements cannot be applied to $(1-\varepsilon, 1-\varsigma)$-gap unweighted Max-2-SAT because the [11] reduction does not work for general $O(1)$-CVP_p.

In Table 1 we compare our lower bounds to those given by [11] under a stronger version of the above conditions and the conditions used in [11]. The conditions are stronger because we use unweighted Max-Cut instead of Max-2-Lin(2) since [11] wouldn't work with $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2). In Table 2 we show the maximum possible γ with interesting time bounds for some other CLBs.

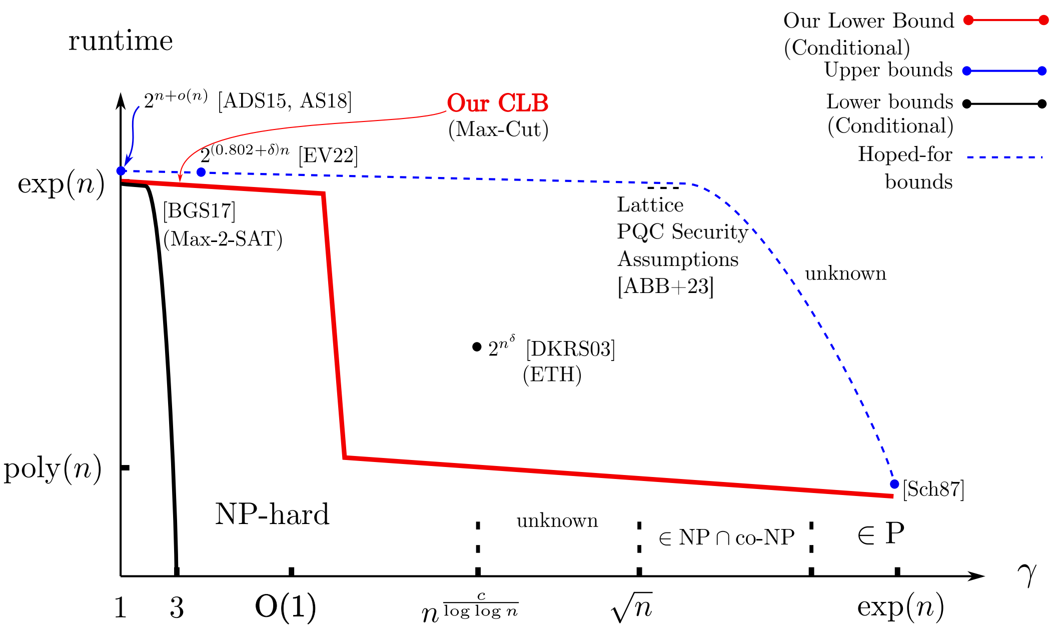

Figure 1 (Color online.) The classical time-complexity of γ-CVP$_2$ as a function of the approximation factor γ. Conditional lower bounds: (solid black line) [11] gives an exponential-time lower bound from $\gamma = 1$ to $\gamma \approx 1.5$, which quickly transitions to a $O(1)$-time lower bound and ends at $\gamma = 3$. This corresponds to the transition into the $O(1)$-regime of $(1-\varepsilon, 1-\varsigma)$-gap unweighted Max-2-SAT as ς tends towards 1. $\gamma = 3$ corresponds to $\varsigma = 1$, which is equivalent to deciding if there is a variable assignment that satisfies at least one constraint (there always is). (solid red line) We give an exponential-time lower bound from $\gamma = 1$ through $\gamma = O(1)$, which then transitions to a polynomial-time lower bound that continues past $\gamma = \exp(n)$. This corresponds to the transition away from the exponential-time regime of (ε, ς)-gap Min-UnCut which begins as the approximation ratio ς/ε exceeds $O(1)$, and the transition into the poly(n)-time regime of (ε, ς)-gap Min-UnCut which starts when ς increases past $\sqrt{\varepsilon}$ or ε decreases past $1/\log n$. (black dot) [14] gives a 2^{n^δ}-time bound when $\gamma = n^{c/\log\log n}$. (dashed black line) lattice based post-quantum cryptosystems assume an exponential-time lower bound for $\gamma = \mathrm{poly}(n)$. The line is intended to correspond to the regimes listed in [2]. Upper bounds, left to right: [4, 6] give a $2^{n+o(n)}$-time algorithm for γ near 1. [16] gives an exponential-time algorithm for γ less than some unspecified constant. [30] gives a polynomial-time algorithm for exponential γ.

The main reason our CLB applies to γ-CVP$_p$ with large approximation factors while also giving tight time bounds is that our reduction uses a novel geometric gadget to represent Max-2-Lin(2) constraints that only uses two basis vectors, while allowing the ratio between the "cost" when a constraint is unsatisfied and the cost when a constraint is satisfied to be fully parameterized. We label this parameter as ι, and our gadget forces the γ-CVP$_p$ instance to "pay" a cost of ι when a constraint is unsatisfied instead of a cost of 1 when

[4] Gap-ETH can be applied to $(1-\varepsilon, 1-\varsigma)$-gap unweighted Max-2-SAT and (ε, ς)-gap unweighted Max-Cut via the first two reductions in [17]. However, these reductions restrict the size of the gap of their output instances. Since the resulting γ is so small, we decided that, as in [11], the effort required to calculate it exactly is not worthwhile.

[5] a gap variant of SETH

[6] $0 < \delta, c < 1/2$

Table 1 Comparison of [11]'s and our CLBs for γ-approximate CVP$_2$, demonstrating that our reduction allows many lower bound conditions to apply to a much wider range of γ-CVP$_p$ problems than before. Note that while we list unweighted Max-Cut here to allow comparison with [11], our CLBs give the same time bounds under the much weaker condition that the equivalent gap regime for $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2) (a weighted problem) requires $2^{\Omega(n)}$ time.

Condition				Our CLB		[11]'s CLB	
Name	ε	ς	Time	γ	Time	γ	Time
unweighted Max-Cut	$O(1)$	$\sqrt{\varepsilon}$	$2^{\Omega(n)}$	$O(1)$	$2^{\Omega(n)}$	$\sqrt{2}$	$2^{\Omega(n)}$
[33] Optimal † ⋆	O(1)	$c_0\varepsilon$	$2^{\omega n/3}$	$\sqrt{c_0}$	$2^{\omega n/3}$	$\min(\sqrt{2}, \sqrt{c_0})$	$2^{\omega n/3}$
Gap-ETH [4] †	0	$O(1)$	$2^{\Omega(n)}$	< 1.115	$2^{\Omega(n)}$	< 1.115	$2^{\Omega(n)}$

†) These are the conditions used in [11] and assume a classical setting.

⋆) [33] is only optimal when $\varsigma \leqslant c_0\varepsilon$ for some constant c_0; thereafter the algorithm presented in Section 1.3 is faster. The time complexity of [33] depends on the matrix multiplication constant $2 \leqslant \omega$.

Table 2 Some conditional lower bound results for γ-approximate CVP$_p$, presented so as to maximize the value of γ while still maintaining interesting time lower bounds.

Result	Condition	Lower Bound	Maximum γ	p
Our Result	gap Max-2-Lin(2) exponential	$2^{\Omega(n)}$	$O(1)$	finite
[11]	gap Max-2-SAT exponential	$2^{\Omega(n)}$	< 3	finite
[3]	gap-SETH[5]	2^n	$< 1 + 1/\mathrm{poly}(n)$	non-even
[14][6]	ETH	2^{n^δ}	$O(n^{c/\log\log n})$	finite

it is satisfied. In the previous state-of-the-art reduction due to [11] this ratio is 3, and it cannot be increased as the family of geometric gadgets they use is inherently limited to a small constant dependent on the choice of norm and the number of variables in a constraint.

The value γ given by the unweighted version of our reduction, shown in Equation (2), helps illustrate the significance of the ratio ι.

$$\gamma = \sqrt[p]{\frac{1+\varsigma(\iota^p - 1)}{1+\varepsilon(\iota^p - 1)}} \tag{2}$$

Comparing it to the value of γ for [11] given in Equation (1) clearly shows that fixing the ratio at 3 constrains the contribution of the (ε, ς)-gap to the approximation ratio while allowing the ratio to be an arbitrarily large parameter which can depend on n and p allows us to maximize the contribution of the gap to γ. But it can go further: when the ratio is constant, that places a constant upper bound on the possible values of γ, even when we make the input problem trivial by setting $\varsigma = 1$. But when the ratio is a parameter we can get any arbitrary γ we want by varying ι and setting $\varepsilon \ll \varsigma$ (or even $\varepsilon = 0$) and keeping $\varsigma \ll 1$, which gives us a non-trivial, polynomial-time conditional lower bound for γ-CVP$_p$ for all γ. This showcases our reduction's unlimited approximation-ratio preserving property, which as far as we can tell is unprecedented in the literature.

1.2.3 Subsequent work

Recently [1] showed an algorithm (related to [33]) for exact CVP$_p$ which can solve the CVP$_p$ instances created by our and [11]'s reductions in the same amount of time as [33] solves their input Max-2-CSP instances. This means that the [33]-based CLBs for exact CVP$_p$ shown by us and by [11] are perfectly tight – any reduction to CVP$_p$ on fewer basis vectors would result in a faster algorithm for an exact Max-2-CSP than [33].

That concludes our overview of CLBs for γ-CVP$_p$. Next we will discuss our new algorithms for $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2).

1.3 Our Min-UnCut algorithms

Exact Min-UnCut was one of Karp's first 21 NP-complete problems [22]. The current best classical algorithm for it is due to [33], who gave an exponential-sized reduction from Min-UnCut and Max-2-SAT to a graph problem in P and a way to solve that problem using matrix multiplication. This means that the time complexity of [33]'s algorithm is given in terms of the matrix multiplication exponent $2 \leqslant \omega$. The complexity of [33]'s algorithm is $O(2^{\omega n/3})$ which is approximately $O(2^{0.79n})$ using the current best exponent $\omega \approx 2.371$ due to [8]. The current best algorithm for (ε, ς)-gap Min-UnCut with constant approximation factors $\alpha = \varsigma/\varepsilon$ in the conjectured NP-hard regime is $\exp(n/2^{\Omega(\alpha^2)})$-time [26][7]. [10] give a sub-exponential algorithm for gap Max-2-Lin(2) in the region where $\varsigma = C\sqrt{\varepsilon}$ for some constant C which gets slower as $C \to 1$, reaching $O(2^n)$ time when $C = 1$. The current best quantum algorithm for exact Min-UnCut and Max-2-Lin(2) is naive search over the variable assignments using Grover's algorithm [19], which takes $O(2^{n/2})$ time. For more about (ε, ς)-gap Min-UnCut, Max-2-Lin(2), or the Unique Games Conjecture, see Section 2.4 of the full version of this paper or the survey [23].

We give a new $O\Big(2^{\left(\frac{1}{3}+\frac{2}{6\gamma^2-3}+o(1)\right)n}\Big)$-time quantum algorithm for a special case of γ-CVP$_2$ and then use our reduction from Theorem 1.1 to give an almost $O(2^{n(\frac{1}{3}+\frac{\varepsilon}{6\varsigma}+o(1))})$-time algorithm for $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2). This is the first quantum algorithm to improve upon naive search for $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2). We also get an almost $O(2^{n(\frac{1}{2}+\frac{\varepsilon}{4\varsigma}+o(1))})$-time classical algorithm for $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2) by starting with a $O(2^{\left(\frac{1}{2}+\frac{1}{4\gamma^2-2}+o(1)\right)n})$-time classical algorithm from [25].

Our classical algorithm is faster than [10] when $\varsigma \leqslant \sqrt{\varepsilon}$ and it is faster than [33] when $c_0\varepsilon < \varsigma$ for some constant $c_0 \approx 3$ which depends on the sub-constant factors in the exponent and on ω. Due to large implicit constants in [26], our algorithm is faster when $\varsigma < c_1\varepsilon$ for some large constant c_1. That makes it the best known classical algorithm for $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2) or (ε, ς)-gap Min-UnCut when $c_0\varepsilon < \varsigma < c_1\varepsilon$. Our quantum algorithm is faster than our classical algorithm, and is faster than naive search when $\varsigma > c_q\varepsilon$ for some constant c_q dependent on the sub-constant factors in the exponent, so it is the best known algorithm for $c_q\varepsilon < \varsigma < c_1\varepsilon$.

When $\varepsilon = o(1)$, our quantum and classical algorithms have complexities $O\Big(2^{\left(\frac{1}{3}+o(1)\right)n}\Big)$ and $O\Big(2^{\left(\frac{1}{2}+o(1)\right)n}\Big)$ respectively. So our algorithms are also faster than all other known algorithms when $\varepsilon = 1/f(n)$ for $f(n) \leqslant \log n$ and $\varsigma \leqslant \sqrt{\varepsilon}$.

Next we will give an overview of our last major advancement: the no-go results against quantum reductions from k-SAT to CVP$_2$ or Min-UnCut.

1.4 Our no-go theorems

After [11], attempts were made to circumvent the barriers against extending those reductions so that they are able to produce γ-CVP$_p$ instances with greater approximation factors γ or with even p. Although those attempts failed, they also found good reasons why the barriers

[7] Note that a Min-UnCut algorithm is listed as an open question in the published version of this article and is present in the arXiv version which was submitted later.

are likely insurmountable and created a new area of fine-grained complexity in the process. The focus was first on extending the reduction from k-SAT to CVP_2; [3] found that the geometric gadget used in the reduction ((p,k)-isolating parallelepipeds) to represent k-SAT clauses does not exist for even ℓ_p-norms when $k > p$, then [5] found that (most) reductions from k-SAT on n variables to CVP_2 on $\mathrm{poly}(n)$ basis vectors would collapse the polynomial hierarchy! This also helped explain the difficulty in extending the reduction to work for larger γ: [5] found that since for every p there is a constant γ_p such that γ_p-CVP_p can be reduced to CVP_2 (a result due to [16]), any reduction which produces γ-CVP_p instances with $\gamma \geqslant \gamma_p$ could still be subject to no-go theorems for CVP_2.

Although unweighted Max-2-SAT isn't k-SAT, these no-go results can also help explain the limitations of the [11] reduction from $(1-\varepsilon, 1-\varsigma)$-gap unweighted Max-2-SAT to γ-CVP_p. If k-SAT can't be reduced to γ-CVP_p with larger γ, then the geometric gadgets which could be used to represent k-SAT must fail for larger γ just like how [3] found that they fail for even p. Since the [11] reduction from unweighted Max-2-SAT is reliant on the same geometric gadgets (isolating parallelepipeds) used in their reduction from k-SAT, it makes sense that it would face limitations similar to what it would face if it were a reduction from k-SAT.

[5] realized that since tight conditional lower bounds (CLBs) hinge on reductions with a specific numerical relationship between instance size of the input problem and the instance size of the output problem (also known as fine-grained reductions, which we explain in more detail in Section 1.2), their no-go theorems mean that conditions about the time complexity of k-SAT like the Strong Exponential Time Hypothesis (SETH) probably can't be applied to CVP_2.

We show that there are no polynomial-sized, non-adaptive, *quantum* polynomial-time reductions from k-SAT to CVP_2 with two-sided error unless there are quantum statistical zero-knowledge proofs for all languages in NP. This result shows a substantial barrier against proving the fine-grained complexity of CVP_2 using the Quantum Strong Exponential Time Hypothesis and fills a gap in [5], which only showed no-go results for *classical* reductions. Somewhat counter-intuitively, because we were able to avoid these limitations with our fine-grained reduction from $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2) to γ-CVP_p, we can show that these limitations must apply to quantum fine-grained reductions from k-SAT to Max-2-Lin(2) no matter how complex the weights are. Note that since there is a trivial reduction from $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2), exact Min-UnCut, and (ε, ς)-gap Min-UnCut to Max-2-Lin(2), the no-go theorem applies to them as well.

Our reduction also allows us to directly rule out classical reductions from k-SAT to Min-UnCut based on the results of [5]. In particular, we rule out all fine-grained reductions with one-sided error and all non-adaptive fine-grained reductions with two-sided error. Surprisingly, these barriers seem to close off the most tempting approaches toward understanding the fine-grained complexity of Max-Cut. However, they might help explain the lack of results in this area. These results could also be derived indirectly from a corollary in [11] and [5], although this seems to have been unnoticed despite its significance for such a well-studied problem.

These results mean that there is now a group of problems with fine-grained reductions to CVP_2 in addition to the existing group of problems with fine-grained reductions from k-SAT, and that there (probably) cannot be fine-grained reductions from the latter group to the former group. This suggests that the two groups can be considered as two distinct *fine-grained* complexity classes. Note that these fine-grained complexity classes are independent of regular complexity classes since, as demonstrated by [33]'s algorithm described in Section 1.3, problems in P and problems that are NP-complete can reside in the same fine-grained complexity class.

We also show that all "natural" reductions from n-variable 3-SAT to Min-UnCut must use $\frac{4}{3}(n-2)$ vertices for a definition of "natural" given by [3].

Now we have completed an overview of all of our advancements. Next we will give a brief overview of our reduction.

1.5 Our Reduction

We first show an abridged reduction from unweighted Max-Cut to CVP_p for $p \in [1, \infty)$ and how the approximation factor for the output CVP_p instances follows from the gap of the input Max-Cut instances, without needing to handle the complexity of the full proof for Max-2-Lin(2). For the full reduction, see the full version of this paper [21].

Given a Max-Cut instance as an ordered set of unordered pairs $E \subseteq [n]^2$ representing m inequality constraints on n variables $v_1, \ldots, v_n$, we produce a basis B with n vectors on $2m$ dimensions and a target $\boldsymbol{t} \in \mathbb{R}^{2m}$. We represent B as a matrix in $\mathbb{R}^{2m \times n}$ with the basis vectors as its columns. We construct B and $\boldsymbol{t}$ as a series of m vertically stacked blocks, where each block is 2 rows tall:

$$B := \begin{pmatrix} B_{1:2} \\ B_{3:4} \\ \vdots \\ B_{2m-1:2m} \end{pmatrix}, \boldsymbol{t} := \begin{pmatrix} \boldsymbol{t}_{1:2} \\ \boldsymbol{t}_{3:4} \\ \vdots \\ \boldsymbol{t}_{2m-1:2m} \end{pmatrix}.$$

Each pair of blocks (one in B and one in $\boldsymbol{t}$) makes up one gadget, and each gadget encodes one constraint from E; specifically, we construct the gadget $B_{2k-1:2k}, \boldsymbol{t}_{2k-1:2k}$ to implement the k^{th} constraint of E as follows:

$$B_{2k-1:2k} := \begin{pmatrix} \cdots & -1 & \cdots & 1 & \cdots \\ \cdots & \iota & \cdots & \iota & \cdots \end{pmatrix}, \boldsymbol{t}_{2k-1:2k} := \begin{pmatrix} 0 \\ \iota \end{pmatrix},$$

where only the i^{th} and j^{th} columns of $B_{2k-1:2k}$ are shown with $(i,j) := E_k$, $i < j$, and the remaining columns are filled with 0s. We show the columns in order with column i first. $\iota > 1$ is a parameter we use to ensure that the distance to the target is greater when the constraint is unsatisfied than when the constraint is satisfied. We use two rows per constraint to ensure that the basis vectors for our lattice are linearly independent − otherwise our reduction wouldn't produce valid γ-CVP_p instances. We visualize the block in Figure 2 by plotting the $(2k-1)^{\text{th}}$ and $2k^{\text{th}}$ rows of the vectors $B_{\cdot,i}, B_{\cdot,j}, \boldsymbol{t}$, and $B_{\cdot,i} + B_{\cdot,j}$. $B_{\cdot,i}$ represents the i^{th} column of B, which is also the i^{th} basis vector.

As you can see from Figure 2, $B_{\cdot,i}$ and $B_{\cdot,j}$ are equidistant from the target $\boldsymbol{t}$, $\mathbf{0}$ and $B_{\cdot,i} + B_{\cdot,j}$ are also equidistant from $\boldsymbol{t}$, and $B_{\cdot,i}$ and $B_{\cdot,j}$ are closer to $\boldsymbol{t}$. If we interpret the coefficient of $B_{\cdot,i}$ as the assigned value of v_i, we find that the gadget matches the preferences of the constraint $v_i \neq v_j$ exactly: it prefers $(v_i, v_j) \in \{(0,1),(1,0)\}$ over $(v_i, v_j) \in \{(0,0),(1,1)\}$ and it has no preference between $(0,0)$ and $(1,1)$ and no preference between $(0,1)$ and $(1,0)$.

▶ **Theorem 1.3** (Informal). *The above reduction reduces* (ε, ς)-gap $(1-\varepsilon, 1-\varepsilon^c)$-gap unweighted Max-Cut *to* $\sqrt[p]{\varsigma/\varepsilon}$-$\mathrm{CVP}_p$.

Proof sketch. We first show that the reduction produces a lattice and target with distance at most $(m(1+\varepsilon(\iota^p - 1)))^{1/p}$ when given a set of constraints that is $(1-\varepsilon)$-satisfied by some variable assignment $v_1, \ldots, v_n$ (this is the completeness condition for our reduction).

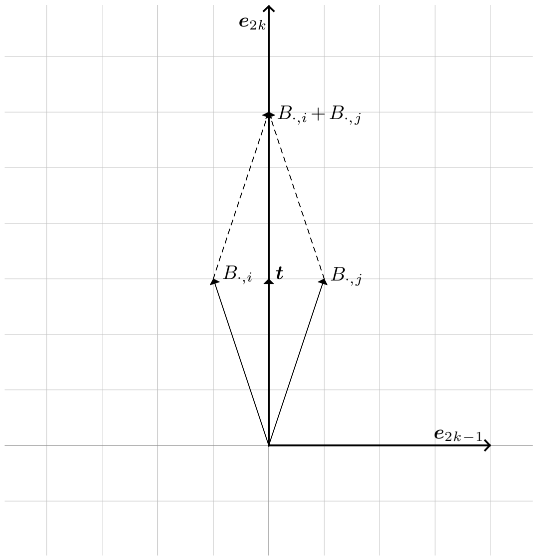

Figure 2 visualization using $\iota = 3$ of the k^{th} gadget, which encodes the constraint $v_i \neq v_j$.

Consider the lattice point $\boldsymbol{u} := \sum_{i=1}^{n} v_i B_{\cdot,i}$. Then

$$|\boldsymbol{u} - \boldsymbol{t}|_{2k-1:2k} = \begin{cases} \binom{1}{0} & \text{the } k^{\text{th}} \text{ constraint is satisfied} \\ \binom{0}{\iota} & \text{the } k^{\text{th}} \text{ constraint is not satisfied} \end{cases}$$

by the construction of the k^{th} gadget (where we use $||$ to denote element-wise absolute value), so by summing up the blocks corresponding to satisfied constraints and the blocks corresponding to unsatisfied constraints we get

$$\mathrm{dist}_p(\boldsymbol{u}, \boldsymbol{t})^p = m(1-\varepsilon) + m\varepsilon\iota^p = m(1 + \varepsilon(\iota^p - 1))$$

as claimed (note the power of p on the left hand side).

Next we show that the reduction produces a lattice and target with distance greater than $(m(1 + \varsigma(\iota^p - 1)))^{1/p}$ when given a set of constraints that is less than $(1 - \varsigma)$-satisfied by any variable assignment (this is the soundness condition for our reduction). Assume that there is always a lattice point with 0-1 coefficients that is at least as close to the target as every other lattice point (we omit proof of this fact in this sketch). We can construct a variable assignment $v_1, \ldots, v_n$ from any lattice point with 0-1 coefficients $\boldsymbol{u}$ by choosing v_i to be the coefficient of $B_{\cdot,i}$. Then by the argument given above for completeness the distance from $\boldsymbol{u}$ to $\boldsymbol{t}$ is given by the number of constraints satisfied by $v_1, \ldots, v_n$ as

$$\mathrm{dist}_p(\boldsymbol{u}, \boldsymbol{t})^p = (\#\text{ of satisfied constraints}) + \iota^p(\#\text{ of unsatisfied constraints}).$$

Since every variable assignment satisfies fewer than $m(1 - \varsigma)$ constraints,

$$\mathrm{dist}_p(\boldsymbol{u}, \boldsymbol{t})^p > m(1-\varsigma) + \iota^p m\varsigma = m(1 + \varsigma(\iota^p - 1))$$

for any lattice point $\boldsymbol{u}$ with 0-1 coefficients. Applying our assumption applies this bound to all lattice points as claimed.

By choosing $r := (m(1 + \varepsilon(\iota^p - 1)))^{1/p}$ and $\gamma r := (m(1 + \varsigma(\iota^p - 1)))^{1/p}$ we get that

$$\gamma = \frac{(m(\varsigma(\iota^p - 1) + 1))^{1/p}}{(m(\varepsilon(\iota^p - 1) + 1))^{1/p}} = \sqrt[p]{\frac{1 + \varsigma(\iota^p - 1)}{1 + \varepsilon(\iota^p - 1)}} \approx \sqrt[p]{\varsigma/\varepsilon}$$

as required, when ι is sufficiently large. ◀

Next we will give some context on the longstanding technical difficulty our reduction overcame.

1.6 Difficulty of fine-grained reductions to γ-CVP_p with large γ

Our reduction overcomes a longstanding technical difficulty for past reductions to lattice problems: producing an instance that has a large approximation factor while maintaining a small number of basis vectors. As you may have noticed in Section 1.2, all past reductions to γ-CVP_p have either produced many basis vectors to get large γ or produced few basis vectors with small γ. We will try to give some intuition as to why this double-bind occurred.

One way to think about how reductions to γ-CVP_p with large γ such as [9, 14, 20][8] work is that they increase the number of dimensions polynomially while also increasing the number of basis vectors polynomially. In general, increasing the number of basis vectors produced by a reduction from a CSP to γ-CVP_p decreases the distance to the target, while increasing the number of dimensions produced by a reduction increases the distance to the target. These reductions construct their additional basis vectors in a systematic manner which reduces the distance much more if the input instance is good than if it is bad. When combined with the fact that the bad instances have a long distance to reduce from this provides a large approximation factor (remember that the approximation factor is the long distance divided by the short distance). However, we know from Section 1.7 that this strategy doesn't produce fine-grained reductions because it uses too many basis vectors.

Ideally, we would like to produce a reduction from a problem on n variables to γ-CVP_p with exactly n basis vectors. Such reductions to exact CVP_p were known at least as far back as [9], but extending them to work for larger γ is difficult.

To describe these difficulties in constructing problems to γ-CVP_p, it is helpful to have a generalization for the problems we are reducing from. We will use Constraint Satisfaction Problems (CSPs), which neatly generalize all the well-studied problems mentioned in this paper. CSPs are about deciding if there is a combination of variable assignments that is good enough at satisfying a set of constraints (restrictions on variable assignments). An assignment is considered good enough if it satisfies a $1-\varepsilon$ fraction of the constraints; if no assignment is good enough, then it is promised that all assignments can only satisfy at most a $1-\varsigma$ fraction of constraints. The input size is given by the number of variables n and the number of constraints m.

One difficulty is as follows: these reductions (including our own) always represent the CSP variables as basis vectors so that they can correspond the integer coefficients of the lattice points to the values of variables in the CSP[9]. There are infinitely many integer coefficients, but only a finite number of possible values for each variable, so to prove that the reduction faithfully implements the CSP we must show that the valid integer coefficients that correspond to possible CSP variable values are always further away than the invalid coefficients. This is usually done by adding a "leashing gadget": one or more dimensions where the value of basis vectors and target forces lattice points to be far away when a basis vector is assigned an invalid integer coefficient. For example, to force a basis vector to only have valid coefficients 0 and 1 we might set the target to α in the leashing gadget's dimension and the basis vector to 2α for some very large α. Since the basis vectors must be allowed

[8] Again, while some of these reductions as written are to a slightly different problem γ-SVP, by [18] they are also reductions to γ-CVP_p and the principles described here are the same.

[9] This is in contrast to, for example, [14] where there were enough basis vectors to have them represent every possible combination of constraint and variable assignment.

at least two valid integer coefficients but the target can only have one value, the leashing gadgets must add some distance to every γ-CVP$_p$ instance, which we can again call α. To prove that the leashing gadget always works, α must exceed the gap between close and far instances of γ-CVP$_p$ for all the other dimensions combined which caps the approximation factor at 2.

This specific difficulty was first overcome by [11] which showed a CVP$_p$ gadget (2-dimensional isolating parallelepipeds) that they proved will faithfully implement a CSP constraint (2-ary binary conjunction) even when "unleashed", breaking the $\gamma = 2$ barrier. However, the double-bind remained, due to another difficulty: the factor of the distances of far and close lattice points on an isolating parallelepiped is inherently limited by a small explicit constant. Even with the most favorable parameters, unsatisfied constraints are limited to 3 times the distance of satisfied constraints, limiting the overall approximation factor to $\gamma < 3$.

1.7 Why fine-grained reductions must be small

Reductions from some Problem A to some Problem B are commonly used in both "coarse-grained" complexity, to establish that B is in the same complexity class as A, and "fine-grained" complexity, to establish a specific numerical relationship between the complexities of A and B. In general, a reduction from A to B is a procedure which solves an instance of A using the solutions to instances of B. A reduction R from Problem A to Problem B which solves any size-n instance of A using the solution(s) to $c_R(n)$ many size-$s_R(n)$ instance(s) of B in time $T_R(n)$ naturally gives the following upper-bound on time complexity of A T_A in terms of the time complexity of B T_B:

$$T_A(n) \quad \leqslant \quad c_R(n) T_B(s_R(n)) + T_R(n).$$

What makes reductions so useful in complexity theory is that this inequality is also a *lower-bound* on T_B in terms of T_A. We can make this clearer by rearranging the inequality and introducing a change of variables $n' = s_R(n)$:

$$\frac{T_A(s_R^{-1}(n')) - T_R(s_R^{-1}(n'))}{c_R(s_R^{-1}(n'))} \quad \leqslant \quad T_B(n').$$

Note that s_R, commonly known as the size of the reduction R, is present in every term.

A common practice is to give a polynomial-size, polynomial-time, many-to-one reduction and assume that problem A requires super-polynomial time; plugging these parameters into our inequality gives $T_B(n) \geqslant \text{superpoly}(\text{poly}^{-1}(n))$ which is a super-polynomial lower-bound on T_B (since $T_R(n) \in o(T_A(n))$ it does not contribute to the asymptotic complexity of T_B). Thus polynomial-size reductions are an effective way to place Problem B in the same complexity class as Problem A.

While such reductions can answer some coarse-grained complexity questions, they are too large to convert a good algorithm for B into a good algorithm for A or a good lower-bound for A into a good lower-bound for B. This is because the large reduction size converts large changes on one side of the inequality to small changes on the other side.

Let us demonstrate by a lower-bound example: assume that Problem A requires $2^{n/2}$-time and let C be a many-to-one reduction from A to B with size $2^k n^k$ for some $k > 1$ in polynomial time, and let R also be a reduction from A to B in polynomial time with size $2n$. To emphasize the point, let R also use n^{10} many instances of B instead of just one. Then by our inequality, reduction C establishes a $\Omega\left(2^{\sqrt[k]{n}/4}\right)$ lower-bound on B while reduction R establishes a $\Omega(2^{n/4}n^{-10})$ lower-bound. Since $k > 1$, it's clear that R gives a much, much better asymptotic lower-bound than C does. Now imagine that a celebrated result shows large improvement in the lower-bound for A, from $2^{n/2}$-time to 2^n. How does that change our lower-bound results for B? The bound given by C improves by only a factor of $2^{\sqrt[k]{n}/4}$ (to $\Omega\left(2^{\sqrt[k]{n}/2}\right)$) while the bound given by R improves by a factor of $2^{n/4}$ (to $\Omega(2^{n/2}n^{-10})$)!

Next we will demonstrate the importance of the exact constant factor of our linear-sized reduction. Let reduction R stay as before, and consider another reduction F which is exactly like R except that it has size n instead of $2n$ and takes time $2^{\sqrt[4]{n}}$ instead of poly(n). Then reduction F gives a lower bound of

$$\left(2^n - 2^{\sqrt[4]{n}}\right) n^{-10} \in \Omega(2^n n^{-10})$$

on B instead of the $\Omega(2^{n/2}n^{-10})$ bound given by R! The above comparisons demonstrate that a reduction which can give good fine-grained complexity results (a fine-grained reduction) for potentially exponential-time problems must be a linear-sized reduction, that the constant factor in the reduction size is more important than the complexity of other aspects of the reduction, and that the constant factor is ideally 1 or less.

1.8 Open questions

We leave the following questions to future research.

1. Can the fine-grained hardness of γ-CVP$_2$ be further extended to show that there are no $O(2^{n^{1-\delta}})$-time algorithms with $\delta > 0$ for γ-CVP$_2$ with approximation factors further into the poly$(\log n)$ regime and even beyond it towards $\sqrt{n}$ regime used by modern cryptography?
2. Now that we have identified that fine-grained approximation-ratio preserving reductions are achievable, can we create other fine-grained approximation-ratio preserving reductions that help answer questions in the hardness of approximation?
3. Can the fine-grained hardness results shown in this work also be applied to the approximate Shortest Vector Problem?
4. Do fine-grained reductions to γ-CVP$_p$ which utilize the "full power" of γ-CVP$_p$ by using lattice coordinates with many different coefficients instead of just two lead to stronger lower bounds? Or maybe γ-CVP$_p$ is fully characterized by existing reductions, since for $1 < p < \infty$ any line formed by a basis vector can only be closest to the target at two points in the ℓ_p-norm[10]? If the former is true, then hopefully we can prove much higher fine-grained lower bounds on γ-CVP$_p$. If the latter is true, then we can use existing algorithms to match our current lower bounds. See Section 2.3 of the full version of this paper [21] for how γ-CVP$_p$ problems with different lattice coefficients relate to each other.
5. The "no-go" results and reductions discussed in this paper suggest that Min-UnCut and γ-CVP$_2$ belong together in a certain fine-grained complexity class, and that this class is separate from the much better understood fine-grained complexity class to which k-SAT belongs. What is the true fine-grained complexity lower bound for this class? How should we formulate a fine-grained lower bound conjecture for this class, and which problem should the conjecture be formulated for? One candidate is that there is no $O(2^{n^{1-\delta}})$-time Max-Cut algorithm for $\delta > 0$; another candidate in use by some cryptographers is that there is no $O(2^{0.2075n})$-time algorithm for γ-CVP$_2$.

[10] [3] give a good technical argument for this possibility.

1.9 Full version of the paper

Our theorems and their proofs are presented in the full version of this paper [21]. What follows will be the structure of the full paper for the reader's convenience.

In Section 2 of [21] we introduce the notation used in this paper. We also present the necessary definitions and known results for fine-grained complexity, lattice problems, Max-Cut, and approximate nearest neighbor. In Section 3 of [21] we prove our main theorem Theorem 1.1. In Section 4 of [21] we use this theorem to apply classical and quantum fine-grained lower bounds on the hardness of γ-CVP_p. In Section 5 of [21] we use our reduction to show barriers against quantum and classical fine-grained hardness results for CVP_p, Max-2-Lin(2), and Min-UnCut based on (Q)SETH. In Section 6 of [21] we show faster classical and quantum algorithms for $(1-\varepsilon, 1-\varsigma)$-gap Max-2-Lin(2) including (ε, ς)-gap Min-UnCut.

References

1 Amir Abboud and Rajendra Kumar. On Beating \$2^n\$ for the Closest Vector Problem. doi:10.48550/arXiv.2501.03688.

2 Divesh Aggarwal, Huck Bennett, Zvika Brakerski, Alexander Golovnev, Rajendra Kumar, Zeyong Li, Spencer Peters, Noah Stephens-Davidowitz, and Vinod Vaikuntanathan. Lattice Problems beyond Polynomial Time. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1516–1526, Orlando FL USA, June 2023. ACM. doi:10.1145/3564246.3585227.

3 Divesh Aggarwal, Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. Fine-grained hardness of CVP(P): Everything that we can prove (and nothing else). In *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '21, pages 1816–1835, USA, 2021. Society for Industrial and Applied Mathematics.

4 Divesh Aggarwal, Daniel Dadush, and Noah Stephens-Davidowitz. Solving the Closest Vector Problem in 2^n Time – The Discrete Gaussian Strikes Again! In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 563–582, 2015. doi:10.1109/FOCS.2015.41.

5 Divesh Aggarwal and Rajendra Kumar. Why we couldn't prove SETH hardness of the Closest Vector Problem for even norms! In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2213–2230, Santa Cruz, CA, USA, November 2023. IEEE. doi:10.1109/FOCS57990.2023.00138.

6 Divesh Aggarwal and Noah Stephens-Davidowitz. Just Take the Average! An Embarrassingly Simple 2^n-Time Algorithm for SVP (and CVP). *OASIcs, Volume 61, SOSA 2018*, 61:12:1–12:19, 2018. doi:10.4230/OASIcs.SOSA.2018.12.

7 Dorit Aharonov and Oded Regev. Lattice problems in NP intersect coNP. *Journal of the ACM*, 52(5):749–765, 2005. doi:10.1145/1089023.1089025.

8 Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More Asymmetry Yields Faster Matrix Multiplication. doi:10.48550/arXiv.2404.16349.

9 Sanjeev Arora, László Babai, Jacques Stern, and Z Sweedyk. The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997. doi:10.1006/jcss.1997.1472.

10 Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential Algorithms for Unique Games and Related Problems. *Journal of the ACM*, 62(5):1–25, 2015. doi:10.1145/2775105.

11 Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. On the Quantitative Hardness of CVP. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 13–24, 2017. doi:10.1109/FOCS.2017.11.

12 Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for unique games. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 205–214, New York, NY, USA, 2006. Association for Computing Machinery. doi:10.1145/1132516.1132547.

13 Eden Chlamtac, Konstantin Makarychev, and Yury Makarychev. How to Play Unique Games Using Embeddings. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '06, pages 687–696, USA, 2006. IEEE Computer Society. doi:10.1109/FOCS.2006.36.

14 I. Dinur, G. Kindler, R. Raz, and S. Safra. Approximating CVP to Within Almost-Polynomial Factors is NP-Hard. *Combinatorica*, 23(2):205–243, April 2003. doi:10.1007/s00493-003-0019-y.

15 Irit Dinur. Approximating SVP-infinity to within almost-polynomial factors is NP-hard. *Theoretical Computer Science*, 285(1):55–71, 2002. doi:10.1016/S0304-3975(01)00290-0.

16 Friedrich Eisenbrand and Moritz Venzin. Approximate CVPp in time 20.802n. *Journal of Computer and System Sciences*, 124:129–139, 2022. doi:10.1016/j.jcss.2021.09.006.

17 M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified *NP*-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976. doi:10.1016/0304-3975(76)90059-1.

18 O. Goldreich, D. Micciancio, S. Safra, and J. P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999. doi:10.1016/S0020-0190(99)00083-6.

19 Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th ACM Symposium on Theory of Computing (STOC 1996)*. ACM Press, 1996. doi:10.1145/237814.237866.

20 Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, STOC '07, pages 469–477, New York, NY, USA, 2007. doi:10.1145/1250790.1250859.

21 Jeremy Ahrens Huang, Young Kun Ko, and Chunhao Wang. On the (Classical and Quantum) Fine-Grained Complexity of Log-Approximate CVP and Max-Cut. doi:10.48550/arXiv.2411.04124.

22 Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations: Proceedings of a Symposium on the Complexity of Computer Computations, Held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and Sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, pages 85–103. Springer US, Boston, MA, 1972. doi:10.1007/978-1-4684-2001-2_9.

23 Subhash Khot. On the Unique Games Conjecture (Invited Survey). In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 99–121, 2010. doi:10.1109/CCC.2010.19.

24 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, January 2007. doi:10.1137/S0097539705447372.

25 Young Kun Ko and Min Jae Song. Hardness of Approximate Nearest Neighbor Search under L-infinity. doi:10.48550/arXiv.2011.06135.

26 Pasin Manurangsi and Luca Trevisan. Mildly Exponential Time Approximation Algorithms for Vertex Cover, Uniform Sparsest Cut and Related Problems. doi:10.48550/arXiv.1807.09898.

27 Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *J. ACM*, 57(5):29:1–29:29, 2008. doi:10.1145/1754399.1754402.

28 Chris Peikert. *A Decade of Lattice Cryptography.* now, 2016. doi:10.1561/0400000074.

29 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34:1–34:40, 2009. doi:10.1145/1568318.1568324.

30 C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2):201–224, 1987. `doi:10.1016/0304-3975(87)90064-8`.

31 Douglas Stebila. Security analysis of the iMessage PQ3 protocol.

32 P. van Emde-Boas. *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*. Department, Univ., 1981.

33 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2):357–365, 2004. `doi:10.1016/j.tcs.2005.09.023`.