IOP Publishing Quantum Sci. Technol. 11 (2026) 025018 https://doi.org/10.1088/2058-9565/ae4cc6

Quantum Science and Technology

OPEN ACCESS

RECEIVED
4 June 2025

REVISED
4 November 2025

ACCEPTED FOR PUBLICATION
3 March 2026

PUBLISHED
13 March 2026

PAPER

A practically scalable approach to the closest vector problem for sieving via QAOA with fixed angles

Ben Priestley[1,2] and **Petros Wallden**[2,*]

[1] Department of Computer Science, University of Oxford, Oxford, United Kingdom
[2] Quantum Software Lab, School of Informatics, University of Edinburgh, Edinburgh, United Kingdom
[*] Author to whom any correspondence should be addressed.

E-mail: petros.wallden@ed.ac.uk and bwp25@cam.ac.uk

Keywords: lattice cryptography, VQA, QAOA, quantum advantage, fixed angles, CVP

Abstract

The NP-hardness of the closest vector problem (CVP) is an important basis for quantum-secure cryptography, in much the same way that integer factorisation's conjectured hardness is at the foundation of cryptosystems like RSA. Recent work with heuristic quantum algorithms (Yan *et al* 2022 arXiv:2212.12372 [quant-ph]) indicates the possibility to find close approximations to (constrained) CVP instances that could be incorporated within fast sieving approaches for factorisation. This work explores both the *practicality* and *scalability* of the proposed heuristic approach to explore the potential for a quantum advantage for approximate CVP, without regard for the subsequent factoring claims. We also extend the proposal to include an antecedent 'pre-training' scheme to find and fix a set of parameters that generalise well to increasingly large lattices, which both optimises the scalability of the algorithm, and permits direct numerical analyses. Our results further indicate a noteworthy quantum speed-up for lattice problems obeying a certain 'prime' structure, approaching fifth order advantage for quantum approximate optimisation algorithm of fixed depth $p = 10$ compared to classical brute-force, motivating renewed discussions about the necessary lattice dimensions for quantum-secure cryptosystems in the near-term.

1. Introduction

The conjectured hardness of integer (prime) factorisation makes it a central problem to modern information security, becoming the foundation for many widespread public-key cryptosystems, such as RSA [1]. No classical algorithm has yet been found—nor do we ever expect to find one—for factoring in polynomial time (see e.g. Zhang *et al* [2]).

However, it is well known that these cryptosystems are vulnerable to *quantum* adversaries, following Peter Shor's seminal paper uncovering that fault-tolerant quantum computation allows one to factor composite integers in polynomial-time [3]. Since its proposal, we have seen many successful experimental demonstrations of Shor's algorithm on trivial problem instances [4–7].

There have been many new proposals for cryptosystems based on different problems that are believed to be secure against any adversary with a large, fault-tolerant quantum computer (see Bernstein and Lange [8] for a recent review). Most attention is given to *lattice-based cryptography* [9–15]. Whilst there has been no shortage of counter-arguments against some of these works [16–21], proposals based on lattices offer arguably the most promising route to quantum-secure cryptosystems. This is reinforced by the fact that three of the four standardised postquantum secure cryptosystems by NIST are based on lattices (see section 2.3 of Alagic *et al* [22], or the NIST website [23]).

1.1. Quantum-accelerated sieving on a lattice

Classical factoring algorithms often use a method called *sieving* to find smooth relation pairs (sr-pairs; a.k.a. fac-relations). Most famously, the quadratic sieve [24, 25] and the general number sieve [26, 27] (and their variants) are the fastest known classical algorithms at time of writing (see Boudot *et al* [28]).

In these algorithms, sieving is a major bottleneck; e.g. the factoring record set for RSA-250 in 2020 required a computational cost of some 2700 core years, roughly 90.7% of which is accounted for by the sieving procedure [28]. It is thus very motivating to find a fast approach to the search for sr-pairs—*the faster we find them, the faster we factor*.

Claus Peter Schnorr has presented a number of theoretical works for lattice-based sieving methods for integer factorisation [29–31]. Recently, Yan *et al* [32] proposed to accelerate Schnorr's lattice sieving by way of nearest neighbour search via the quantum approximate optimisation algorithm (QAOA) [33].

The principle claim advertised by these works is that this method reduces the spatial requirements for factoring an integer N to $O(\log N/\log\log N)$ qubits. Many have presented strong evidence that this is a considerable underestimate [34–37], and relies on outdated assumptions in the underlying theoretical framework of Schnorr [31]. Moreover, we note the omission of any time-complexity analysis, and thus a disregard for a consideration of the practical utility of their algorithm.

Besides the contention for the suitability of a lattice-based sieving method like Schnorr's [37, 38], the claim in Yan *et al* [32] implies a potential quantum advantage for the (approximate) solving of a particular form of closest vector problem (CVP), which, if realised, has further implications for the security parameters required in lattice-based cryptosystems. We take a particular interest in the QAOA subroutine used to solve this CVP, and specifically with its practical scalability towards larger problem instances.

Furthermore, we extend the subroutine with an antecedent pre-training step for the parameters of the QAOA's ansatz to obtain fixed angles, which allows us to greatly simplify our analyses (as in Boulebnane and Montanaro [39], and Brandao *et al* [40]) and dramatically reduce the computational workload. In parallel with Prokop and Wallden [41], which performs similar analysis for the shortest vector problem (SVP), this is one of the first applications of fixed angles to a cryptographically significant problem, facilitating novel—and much neglected—analysis of time complexity.

1.2. Our contributions

For clarity, we now outline our contributions, primarily extending from the works of Yan *et al* [32] and Schnorr's collection of works for sieving on a lattice [29–31], and from Boulebnane and Montanaro [39] and Brandao *et al* [40] for QAOA with fixed angles:

- A simple yet robust pre-training scheme that dramatically reduces the computational requirements for refining solutions to the CVP.
- Novel time-complexity analysis of a QAOA-based sieving method, using our own implementation derived from Khattar and Yosri [36]. The code for our experiments can be found in [42].
- An optimal scaling for the time complexity and 'generalisability' of the method with respect to lattice dimension by our proposed extension. Interestingly, for fixed depth QAOA (for p up to 10), we achieve polynomial advantage compared to brute-force, that exceeds significantly the 'usual' Grover-type quadratic speed-up [43].

In undertaking the above, we provide an empirical analysis of the prospective security of cryptosystems based on lattice problems with respect to newer variational methods, giving indication for the required security parameters (e.g. lattice dimension) within such schemes. This is important insight due to the rapidly growing interest in, and usage of, variational algorithms driven by their suitability for near-term hardware [44].

1.3. Limitations to application and scope

There are two major limitations that should be taken into account considering the expected impact of our results in practice, and for the scope of their utility.

Firstly, since we adopt the underlying framework from Yan *et al* [32] for the sake of analysis, we inherent their search space: constant size in each basis-vector (dimension), leading to space-complexity $O(n)$ and search space of the form $2^{O(n)}$. However, for similar works with the SVP, it is well-known that a size $O(\log n)$ is required for each basis-vector, and thus a search space of $2^{O(n\log n)}$ is needed to ensure that the true shortest vector is within the search space [45, 46]. Consequently, we do not have guarantees that the quality of the solution is close to the true Closest Vector, asymptotically. This is also related with limitations of Schnorr's method and with the ultimate claim of factorisation [31, 32]. While we are restricted in this same way, we can still say *something* about solution quality; namely, we can give empirical observation on the degree of improvement over the classical guarantees of Babai's nearest plane algorithm [47].

Secondly, by motivating our CVP via a factorisation problem, we are working with a restricted structure in our lattice. As such, our work can be framed as an analysis of 'best-case scenario' lattice problems, wherein assumptions about the discrete gaps between basis vectors can be utilised to design an optimistic algorithm. We will show that exponential effort is required even for this constrained lattice, which can be taken as further evidence against the claims of sublinear factoring, though nonetheless imply a quantum advantage for CVP on certain lattices.

The results of this work are most appropriately interpreted as an understanding of the *practical scalability* of this kind of neighbourhood search on a particularly constrained lattice. These constraints allow us to leverage inherent symmetries of the problem to design a highly effective pre-training scheme. Whether this can be generalised, e.g. to arbitrary CVP structures or for tighter approximations, remains an important open question.

2. Background and preliminaries

2.1. Prime factorisation and sieving

This section gives an overview of the current state of *classical* integer factorisation. We give essential definitions in factoring and number theory, and reduce the problem of factoring to the problem of finding sr-pairs.

Definition 2.1. (integer factorisation problem) Given an odd composite integer $N > 2$, find the prime factors p, q (with $p < q$) such that $N = p \cdot q$.

Let $P_n = \{p_i\}_{i=0,\dots,n}$ denote the nth *prime basis*, where p_i is the ith prime number, and $p_0 := -1$ allows us the capacity to represent negative integers.

Definition 2.2 (smooth number). An integer is called p_n-smooth if all its factors are in P_n. We call p_n the *smooth bound.*

Definition 2.3 (smooth relation pair). Moreover, a pair of p_n-smooth numbers (u_j, v_j) are called a p_n-*smooth relation pair* (sr-pair; a.k.a. *fac-relation* in Schnorr [31]) if for $e_{i,j}, e'_{i,j} \in \mathbb{N}$, we have that

$$u_j = \prod_{i=1}^{n} p_i^{e_{i,j}} \quad \text{and} \quad u_j - v_j N = \prod_{i=0}^{n} p_i^{e'_{i,j}} . \tag{1}$$

The method for factoring that forms the foundations for many of the most efficient classical algorithms goes back to works like Kraitchik [48] and Morrison and Brillhart [49], and was later developed by Dixon [50]. Much of this discussion comes from Schnorr [31].

Given $n+1$ sr-pairs (u_j, v_j), and taking the quotient of the terms in equation (1), we have for $e_{i,j}, e'_{i,j} \in \mathbb{N}$ that

$$\frac{u_j - v_j N}{u_j} \equiv \prod_{i=0}^{n} p_i^{e'_{i,j} - e_{i,j}} \equiv 1 \mod N , \tag{2}$$

since $u_j - v_j N \equiv u_j \mod N$ for any (u_j, v_j). Now, any solution $t_1, \dots, t_{n+1} \in \{0, 1\}$ of the equations

$$\sum_{j=1}^{n+1} t_j \left(e'_{i,j} - e_{i,j} \right) \equiv 0 \mod 2 , \tag{3}$$

for $i = 0, \dots, n$ solves a difference of squares $X^2 - 1 = (X-1)(X+1) = 0 \mod N$ by

$$X = \prod_{i=0}^{n} p_i^{\frac{1}{2} \sum_{j=1}^{n+1} t_j \left(e'_{i,j} - e_{i,j} \right)} \mod N . \tag{4}$$

If $X \neq \pm 1 \mod N$, then this yields two nontrivial factors $\gcd(X \pm 1, N)$ of N, where $\gcd(\cdot)$ denotes the greatest common divisor algorithm, which may be efficiently computed using Euclid's algorithm. This idea comes from Fermat's method for factoring.

Solutions to (3) can be obtained within $O(n^3)$ bit operations since the dimension of the linear system is $O(n)$, and so we are free to neglect this minor part of the workload in factoring N. Hence, the bottleneck in factoring is in *finding these $n+1$ sr-pairs.*

2.2. Lattices and lattice problems

Definition 2.4 (Euclidean lattice). A (Euclidean) lattice $\mathcal{L}$ is a discrete additive subgroup of $\mathbb{R}^n$; that is, a subset $\mathcal{L} \subseteq \mathbb{R}^n$ that is closed under addition and subtraction, and wherein there exists some $\varepsilon > 0$ such that any two distinct lattice points $\mathbf{x} \neq \mathbf{y} \in \mathcal{L}$ are separated by a distance of a least $\|\mathbf{x} - \mathbf{y}\| \geqslant \varepsilon$.

For a basis matrix $B = [\mathbf{b}_1, \ldots, \mathbf{b}_m] \in \mathbb{R}^{n \times m}$ consisting of m linearly independent column vectors in $\mathbb{R}^n$, the lattice $\mathcal{L}(B) = \{B\mathbf{x} : \mathbf{x} \in \mathbb{Z}^m\}$ is generated by all integer linear combinations of $\mathbf{b}_1, \ldots \mathbf{b}_m$. Intuitively, a lattice is simply a regular ordering of points.

Following these semantics, the *dimension* of $\mathcal{L}$ is n, and its *rank* is m. When $n = m$, the lattice $\mathcal{L}$ (and the matrix B) are called *full rank*.

2.2.1. Properties of lattices

There are a couple of interesting properties to notice: lattices are (1) dense; and (2) hard to approximate. For a given region, there may be many lattice points, which can make finding specific points difficult. And, given an arbitrary point, finding nearby points is again difficult.

Definition 2.5 (successive minima). The successive minima for a lattice $\mathcal{L} \subseteq \mathbb{R}^n$ are the positive values $\lambda_1(\mathcal{L}) \leqslant \cdots \leqslant \lambda_n(\mathcal{L})$, where $\lambda_k(\mathcal{L})$ is the smallest radius of a zero-centred ball containing k linearly independent vectors of $\mathcal{L}$.

Thus, $\lambda_1 = \lambda_1(\mathcal{L})$ is the shortest nonzero vector in $\mathcal{L}$.

Definition 2.6 (Hermite constant). The minimal γ satisfying $\lambda_1^2 \leqslant \gamma(\det \mathcal{L})^{2/n}$ for all lattices of dimension n is called the Hermite constant γ_n, where $\det \mathcal{L} = (\det B^{\mathrm{T}} B)^{1/2}$ is the determinant of $\mathcal{L}$.

2.2.2. Problems on lattices

Definition 2.7 (SVP). Given a basis B for a lattice $\mathcal{L}(B)$, find a vector $\mathbf{v} \neq \mathbf{0} \in \mathcal{L}$ such that $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$, where $\|\mathbf{v}\| = (\mathbf{v}^{\mathrm{T}}\mathbf{v})^{1/2}$.

Often it suffices to be only 'close' to the true shortest vector λ_1. In these cases, we refer instead to an *approximate SVP (α-SVP)* for which the condition in definition 2.7 is amended as $\|\mathbf{v}\| \leqslant \alpha \cdot \lambda_1(\mathcal{L})$ for approximation factor $\alpha \geqslant 1$.

The exact value of λ_1 can, in itself, be hard to obtain due to the inherent hardness of the SVP (and of α-SVP). It may then be preferable to instead define the problem according to a (relatively) easily computable value relating to λ_1. For example, we can again amend the condition in definition 2.7 to $\|\mathbf{v}\| \leqslant r \cdot (\det \mathcal{L})^{1/n}$ to obtain the *r-Hermite SVP*. This allows us to check solutions far more easily and thus efficiently, although we lose accuracy in the comparison with the true shortest vector.

Definition 2.8 (CVP). Given a basis B for a lattice $\mathcal{L}(B)$, and a target vector $\mathbf{t} \in \mathrm{Span}(B)$, find a vector $\mathbf{v} \in \mathcal{L}$ such that the distance $\|\mathbf{v} - \mathbf{t}\|$ is minimised; i.e. that $\|\mathbf{v} - \mathbf{t}\| = \mathrm{dist}(\mathcal{L}, \mathbf{t})$.

Again, there exists an *approximate CVP* to weaken the condition of closeness in definition 2.8, and the *r-approximate CVP* weakens the condition further to bring this distance to within r, which could, for example, be computed according to $\det \mathcal{L}$ in place of $\mathrm{dist}(\mathcal{L}, \mathbf{t})$.

All of these problems are understood to be difficult—especially in higher dimensions—to such a degree that even a quantum advantage may be insufficient to yield a polynomial-time solution. Specifically, the decision variant of SVP (GapSVP) is conjectured to be NP-hard [51], and solving either of SVP or GapSVP within a polynomial factor requires superpolynomial time with quantum computation [52]. In general, CVP is thought to be even harder [53–55], and is known to be NP-hard [55]. Figure 1 illustrates simple SVP and CVP problems on a two-dimensional lattice.

For the interested reader, Micciancio and Goldwasser [55] remains a prominent text in the complexity of lattice problems for applications in cryptography.

2.3. QAOA

A good introduction to variational quantum algorithms in general is offered by Cerezo *et al* [44]. In this work, we will focus exclusively on perhaps the most widely studied variational algorithm: *QAOA* due to Farhi *et al* [33]. We are especially interested in the possibility of pre-training a fixed set of angles (see section 2.4), for which QAOA is an ideal candidate [40].

QAOA was originally inspired by the quantum adiabatic algorithm [56, 57]. Adiabatic evolution is replaced with several rounds of propagation between a *problem Hamiltonian* $\hat{H}_C$, which encodes the solution to an optimisation problem $C(z)$ over binary variables $z = z_1, z_2, \ldots$ within its ground state, defining

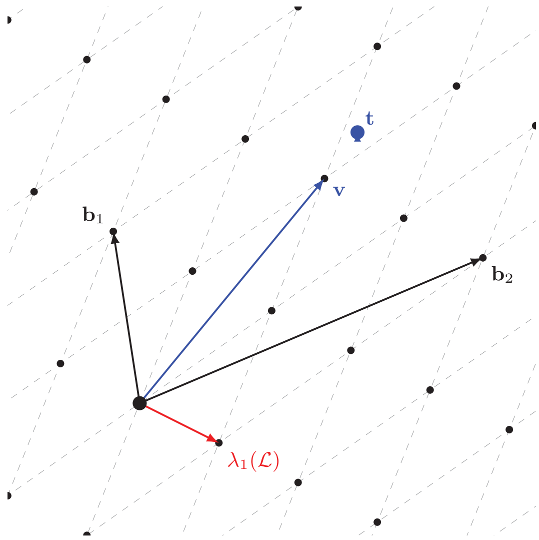

Figure 1. Visualising a two-dimensional, full rank lattice $\mathcal{L}$ generated by the basis $B = [\mathbf{b}_1, \mathbf{b}_2]$. Shown in red is the shortest vector $\lambda_1(\mathcal{L})$ solving the SVP for this lattice. Shown in blue is a CVP with target $\mathbf{t}$, solved by the lattice vector $\mathbf{v}$.

the unitary operator

$$U\left(\gamma, \hat{H}_C\right) = \mathrm{e}^{-\mathrm{i}\gamma \hat{H}_C} \ , \tag{5}$$

and *mixer Hamiltonian B*, whose ground-state is known, defining the unitary operator

$$U(\beta, B) = \mathrm{e}^{-\mathrm{i}\pi\beta B} = \prod_j \mathrm{e}^{\mathrm{i}\pi\beta\sigma_j^x/2} \ , \tag{6}$$

parametrised by angles $0 \leqslant \gamma \leqslant 2\pi$ and $0 \leqslant \beta \leqslant \pi$ respectively, where σ_j^x denotes a Pauli-X operator applied on the jth qubit.

We open a uniform superposition over computational basis states to yield an initial state $|s\rangle = \frac{1}{\sqrt{2}} \sum_z |z\rangle$. Then, for some integer number of ansatz layers $p > 0$, we define the angle-dependent state

$$\begin{aligned} |\gamma, \beta\rangle = & U\left(\beta_p, B\right) U\left(\gamma_p, \hat{H}_C\right) \cdots \\ & \cdots U\left(\beta_1, B\right) U\left(\gamma_1, \hat{H}_C\right) |s\rangle \ , \end{aligned} \tag{7}$$

parametrised by angles $\gamma \equiv \gamma_1 \ldots \gamma_p$ and $\beta \equiv \beta_1 \ldots \beta_p$.

It is shown in Farhi *et al* [33] that

$$\lim_{p \to \infty} \min_{\gamma, \beta} \langle \gamma, \beta | \hat{H}_C | \gamma, \beta \rangle = \min_z C(z) \ , \tag{8}$$

which implies that the optimisation problem can be solved with enough repetitions, if only a good set of angles can be selected. Pessimistically, we can use an outer optimisation loop on a classical computer to search for a good set within the optimisation landscape of $C(z)$ [44]. Optimistically, there may be precedence to neglect much or all of this optimisation by *fixing these angles*.

2.4. Fixed angles for QAOA

Interesting remarks have been made about the apparent independence between the optimisation landscape for the objective function (of some e.g. combinatorial search problem) and the specific problem instance [40, 58, 59].

In particular, Brandao *et al* [40] fix the γ and β parameters to show that, when instances are generated by some reasonable distribution, the objective function is nearly independent of the chosen instance. It is suggested that this could be leveraged to find a good set of parameters for some given instance (possibly at great computational expense), which can then be utilised at no further expense for any subsequent (typical) instance. Indeed, removing the need to search for parameters in each instance allows us to neglect the outer optimisation loop once a set has been found, and hence the amortised cost tends to zero inversely with the number of instances being solved [40].

More recently, Boulebnane and Montanaro [39] have given numerical results that speak to the validity of a 'fixed angle' scheme for eventually obtaining a quantum advantage with QAOA for random k-SAT. They find a significant improvement over Grover's algorithm [60] with greatly reduced quantum circuit depth.

In all of these works, the method for obtaining a fixed set of angles is either to take a random (usually small) problem instance to train (e.g. in Brandao *et al* [40]), or the angles may be randomised within some sensible interval. Preliminary investigation with such a scheme led to underwhelming results for this work, so we seek a more robust pre-training scheme that will find a good set of angles more deliberately.

2.5. Alternative approaches

In a slightly different flavour, the costly optimisation loop within variational algorithms can also be made redundant via the recent linear ramp (LR) QAOA [61–63]. Instead of finding and fixing a set of angles, LR-QAOA uses a (linear) schedule to transfer the optimal parameters from one combinatorial optimisation problem to another (a process referred to as 'transfer learning' in [63]). The principle, however, carries over: expensive instance-optimised VQAs can be traded for highly scalable parameter regimes in QAOA, possessing the capacity to generalise well to differently constructed problems.

The fixed-angles approach we present in this work can be viewed as a genetic algorithm-inspired alternative to the LR-QAOA of Montanez-Barrera *et al* [63]; we are not looking to transfer parameters from some other problem to (α-)CVP, but instead finding a scalable set for combinatoric problems on the prime lattice. Although, promising analyses [61] suggest we might expect that the parameters we find for the CVP could perform admirably on other cryptographically interesting problems on a (prime) lattice, namely (α)-SVP.

Following our discussions in subsection 2.4, our choice to adopt QAOA over e.g. VQE [64] is in the very natural route towards a fixed-angles scheme. Importantly, the parametrisation of the QAOA circuit itself (its structure, the number of parameters, etc) does not depend on the number of qubits, and thus has no dependence on the problem size. Whether our CVP lives in 10 dimensions or 100, the circuits we construct to solve each can be appropriately parametrised in the same way. This very useful circuit construction is not generally available among other popular VQAs; common ansatz used in e.g. VQE in general depend on the circuit width, and thus the number of parameters grows with problem size.

3. Reducing sieving to a CVP on the prime lattice

The problem of finding small integers whose *product* is close to N is translated into the equivalent problem of finding logarithms of small numbers whose *sum* is close to $\log N$. In doing this, we have given ourselves a combinatorial optimisation problem which may be directly expressed as a linear system of lattice vectors.

Of course, the lattice vectors in question must exhibit the properties of the logarithms of primes so that a linear combination represents an sr-pair. Schnorr [31] suggests to construct the so-called *prime lattice* whose basis is defined according to the corresponding factor basis. Some additional randomness is baked in to bring about unique lattice problems that (hopefully) have unique solutions.

Concretely, define the prime lattice $\mathcal{L}(B_{n,c})$ by

$$B_{n,c} = \begin{pmatrix} f(1) & 0 & \cdots & 0 \\ 0 & f(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f(n) \\ N^c \ln p_1 & N^c \ln p_2 & \cdots & N^c \ln p_n \end{pmatrix}, \tag{9}$$

with c the *precision parameter*, and where the $f(i)$ elements correspond to elements from a random permutation of $\{\lceil 1/2\rceil, \lceil 2/2\rceil, \ldots, \lceil n/2\rceil\}$. Hence, the number of lattices arising from a factor basis P_n scales as $O(n!)$.

The target for our CVP on this lattice is defined directly from the composite integer N to be factored:

$$\mathbf{t} = \begin{pmatrix} 0 & \overset{n}{\cdots} & 0 & N^c \ln N \end{pmatrix}^\top. \tag{10}$$

3.1. Suitability of approximation

A critical consideration of the method due to Schnorr [31] is that an approximate solution to this CVP is sufficient. Theoretical results [29–31] leave the door open to the existence of an approximate solution that improves on polynomial-time classical approximations, but which does not succumb to the NP-hardness of exactly solving the full CVP [53–55]. We draw from excellent discussion in Aboumrad *et al* [35], extended from Schnorr [31], to detail the rationale behind this possibility.

Suppose we obtain a set of coefficients e_j for the linear combination of basis vectors of the prime lattice $\mathcal{L}(B_{n,c})$ that approximately solve the CVP. That is, suppose we have found e_j such that

$$\varepsilon := \left| \sum_{j=1}^{n} e_j \ln p_n - \ln N \right| \approx 0 \ . \tag{11}$$

If we set that $u = \prod_{e_j \geqslant 0} p_j^{e_j}$ and $v = \prod_{e_j > 0} p_j^{-e_j}$, then we obtain $\ln |(\frac{u}{vN})| = \varepsilon$. By Taylor's theorem, $u - vN = vN(e^{\varepsilon} - 1) \approx \varepsilon vN$. Schnorr [31]'s argument is that since ε is small, $u - vN$ is also small and thus *likely* to be p_n-smooth. By definition 2.3, (u, v) is an sr-pair.

If we are content to run multiple problem instances, then we need only find e_j coefficients for equation (11) that reduce ε to be small enough such that the probability that they correspond to a useful sr-pair is 'good enough'.

3.2. On the sublinear assumptions of Schnorr [31]

By Minkowski's first theorem (see [65]), the shortest vector λ_1 for any full rank n-dimensional lattice $\mathcal{L} \subseteq \mathbb{R}^n$ is bounded from above as

$$\lambda_1 (\mathcal{L})^2 \leqslant n \cdot (\det \mathcal{L})^{2/n} \ . \tag{12}$$

The discrepancy between the *real* shortest vector λ_1 and this bound can be measured by the *relative density* $\mathrm{rd}(\mathcal{L})$ of the lattice, which gives a ratio between λ_1 and the upper bound estimated by the Hermite constant;

$$\mathrm{rd}(\mathcal{L}) := \frac{\lambda_1 (\mathcal{L})}{\sqrt{\gamma_n} (\det \mathcal{L})^{1/n}} \ , \tag{13}$$

where γ_n is the Hermite constant of definition 2.6.

Schnorr [29] made the critical assumption that a random lattice $\mathcal{L}$ with size-ordered basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_m]$ has a relative density satisfying

$$\mathrm{rd}(\mathcal{L}) \leqslant \left(\sqrt{\frac{e\pi}{2n}} \cdot \frac{\lambda_1 (\mathcal{L})}{\|\mathbf{b}_1\|} \right)^{1/2} \ , \tag{14}$$

and since $\lambda_1(\mathcal{L})\|/\mathbf{b_1}\| \leqslant 1$, this then leads to

$$\mathrm{rd}(\mathcal{L}) = \frac{\lambda_1 (\mathcal{L})}{\sqrt{\gamma_n} (\det \mathcal{L})^{1/n}} \leqslant \left(\frac{e\pi}{2n} \right)^{1/4} \ . \tag{15}$$

We can then propose that, for the lattice $\mathcal{L}(B_{n,c})$ whose dimension satisfies $n = 2c \log N / \log \log N$, and whose relative density satisfies equation (15), there exists some vector $\mathbf{v} \in \mathcal{L}(B_{n,c})$ such that $\|\mathbf{v} - \mathbf{t}\|^2 = O(\log N)$ for any target vector $\mathbf{t} \in \mathrm{Span}(B_{n,c})$. Proofs for this proposition are given in Schnorr [31] and in the appendix of Yan *et al* [32]. Hence, assuming that equation (15) is satisfied, we have only a sublinearly growing lattice dimension in the bit-length $\log N$.

There are strong doubts concerning the validity of this method at scale [34–38], and in particular that the sampling probability decays exponentially such that sr-pairs cannot be found by refinement alone (or, at least, not quickly). We break down this sublinearity assumption with three main arguments from the literature.

Firstly, while the approach to sieving for sr-pairs as a nearest-neighbour search on a lattice has merit experimentally, such a sublinear scheme is only of practical use if the reduction from factoring to the search for sr-pairs works in polynomial time, given the heuristic assumptions of density culminating in equation (15). Practical implementations, such as Ducas [37], dispute that this assumption successfully scales well enough to be cryptographically significant. Theoretical considerations tend to agree with this disputatious attitude [34].

Secondly, since the foundations of Schnorr [29]'s guarantees were formulated in the early 1990s, the theory of lattice problems has developed tremendously, including the reliance of most popular CVP heuristics on ℓ_2-measurements (e.g. [66]). This leaves the proofs in Schnorr [29] (in particular, lemma 2) with limited practical bearing [35].

Thirdly, Aboumrad *et al* [35] use theoretic results [67, 68] to estimate the density of p_N-smooth numbers as N grows (note that p_N-smooth here refers to integers whose prime factors are not greater than N). Combining Dickman's function with the prime number theorem allows them to estimate that the proportion of p_N-smooth numbers below N that may be collected under the sublinear lattice dimension scheme is exponentially small (see section 3.1 of Aboumrad *et al* [35]).

For small N, the assumption is acceptable, but the exponential decay of available smooth numbers renders the assumption unacceptable for large N, and hence the scalability of Schnorr's method [29–31] should be doubted under a sublinear scheme. This decay is the primary cause for the necessity for exponentially many CVP instances needing to be solved for a factorisation to be possible—we need exponentially many solutions to be sure of finding enough unique sr-pairs for post-processing. *For this reason, it is widely criticised that this method, with a sublinear scheme, leads to practical factoring at scale.*

It is not the subject of this work to bring another argument for or against factoring via lattice sieving. Instead, we are keenly interested in the number-theoretic structure that Schnorr [31] imposes on the lattice and the CVP defined on it. While the computational effort required to solve this CVP to an acceptable approximation factor proves to be exponential, there is ample opportunity to exploit the rigid structure of the prime lattice to obtain a quantum advantage heuristically. In section 4, we make the approach to solving the CVP in Yan *et al* [32] *practical* and *scalable* through the use of a fixed-angles scheme. We reiterate that the advantage we can obtain with our approach for the approximate solving of the above CVP *does not lead to factoring*, but could have practical relevance to the bit security of lattice-based cryptographic protocols.

3.3. Hyperparameters

So far, we can see that the hyperparameter n has the role of dictating all of: (1) the dimension of the lattice; (2) the size of the factor basis, and hence the number of integers that may be considered p_n-smooth; and (3) the number of unique prime lattices that can be constructed in our sieving procedure.

Increasing n comes with a dimension-complexity trade-off: with a larger smooth bound p_n, we can more easily find smooth numbers, but we will require more of them (recall we require $n+1$ sr-pairs in section 2.1). The exact correlation between n and time complexity is not well understood in this context, and much of our work here is dedicated to uncovering this relationship empirically.

The N^c term in the formulation of $B_{n,c}$ gives parametrisable precision to the lattice. This is conjectured to be positively correlated with the probability of finding solutions to problems within this lattice, though some have voiced concerns that this is unsubstantiated [36, 37]. As for our later analyses, we will fix c and, similar to Yan *et al* [32] and Khattar and Yosri [36], we exchange N^c for 10^c to give a consistent basis between instances. This will improve the effectiveness of pre-training.

4. A method for CVP refinement

4.1. Polynomial-time classical approximation

We begin by considering a polynomial-time procedure that serves up an approximate solution to the CVP. This procedure consists of a lattice reduction step using the LLL-reduction algorithm [69] (described in appendix A) followed by Babai's nearest plane algorithm [47], which brings us to within a factor $2(2/\sqrt{3})^n$ of $\mathrm{dist}(\mathcal{L},\mathbf{t})$. The following is a high-level summary of Babai's algorithm.

Having produced an LLL-reduced basis $D=[\mathbf{d}_1,\ldots,\mathbf{d}_n]$ for $B_{n,c}$, we perform Gram–Schmidt orthogonalisation (without column normalisation) to yield $\tilde{D}=[\tilde{\mathbf{d}}_1,\ldots,\tilde{\mathbf{d}}_n]$. The nearest plane algorithm then comprises a series of projections of the target vector $\mathbf{t}$ onto $\mathrm{Span}(\tilde{D})$ followed by a rounding of the coefficients to snap onto a nearby lattice point. Concretely, begin by setting $\mathbf{b}_{\mathrm{op}}\leftarrow\mathbf{t}$, then for each j in $n,\ldots,1$ compute the Gram–Schmidt coefficient $\mu_j=\langle\mathbf{t},\tilde{\mathbf{d}}_j\rangle/\langle\tilde{\mathbf{d}}_j,\tilde{\mathbf{d}}_j\rangle$, round to the nearest integer $c_j=\lceil\mu_j\rfloor$, and update $\mathbf{b}_{\mathrm{op}}\leftarrow\mathbf{b}_{\mathrm{op}}-c_j\mathbf{d}_j$.

Intuitively, when considering index j, we are taking the j-dimensional subspace $\mathrm{Span}\{\tilde{\mathbf{d}}_1,\ldots,\tilde{\mathbf{d}}_j\}$ and finding the integer c_j that minimises the distance from $c_j\mathbf{b}_j+\mathrm{Span}\{\tilde{\mathbf{d}}_1,\ldots,\tilde{\mathbf{d}}_{j-1}\}$ to $\mathbf{t}$ [35]; in each step, we find the *nearest (hyper)plane*, giving the algorithm its name.

4.2. Refinement as a minimum-eigenstate optimisation problem

Suppose we have found an approximate solution $\mathbf{b}_{\mathrm{op}}=\sum_{j=1}^{n}c_j\mathbf{d}_j$, where again the $c_j=\lceil\mu_j\rfloor$ are the rounded Gram–Schmidt coefficients. Our goal is to 'refine' this solution efficiently.

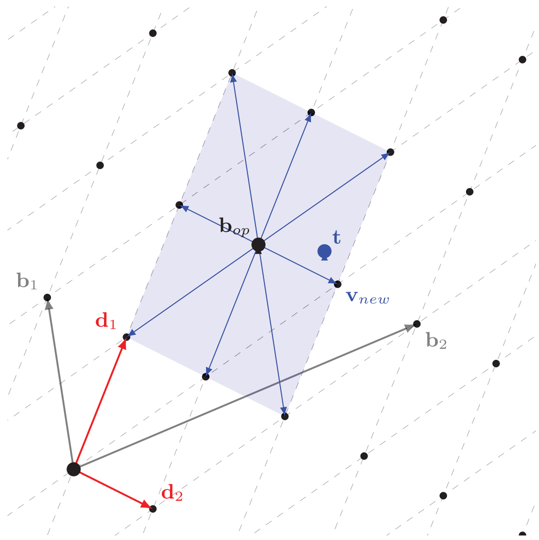

Figure 2. Visualisation for the nearest neighbour search centred on the approximate solution $\mathbf{b}_{\text{op}}$. The LLL-reduced basis $D = [\mathbf{d}_1, \mathbf{d}_2]$ has been computed and used to define a local neighbourhood (shaded blue region) that aims to encompass lattice points that may be closer to the target $\mathbf{t}$. In this case, $\mathbf{v}_{\text{new}}$ is shown with floated values $x_1 = -1$ and $x_2 = 1$, and represents a refinement to the approximate solution.

The potential for a quantum advantage falls out of the rounding operation $\lceil\cdot\rfloor$; classically, this operation considers rounding in each direction one after another (i.e. $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$), but quantumly they can be considered simultaneously [32]. Doing this by classical means increases the number of operations exponentially, making it infeasible. Yan *et al* [32] propose to leverage the effect of superposition to simultaneously encode the two values obtained by each rounding operation within qubits.

To this effect, we will search in the unit neighbourhood centred on $\mathbf{b}_{\text{op}}$ to capture all possible rounding arrangements for the coefficients, looking for that which is closest to the target vector $\mathbf{t}$. This search has been illustrated on a simple two-dimensional lattice in figure 2 to provide a flavour for the combinatoric problem we are setting up. Concretely, any new vector $\mathbf{v}_{\text{new}}$ is obtained by randomly floating $x_j \in \{0, \pm 1\}$ on the c_j coefficients, where this $\pm$ comes from the rounding already applied in Babai's nearest plane algorithm;

$$\mathbf{v}_{\text{new}} = \sum_{j=1}^{n} \left(c_j + x_j\right) \mathbf{d}_j = \mathbf{b}_{\text{op}} + \sum_{j=1}^{n} x_j \mathbf{d}_j \ . \tag{16}$$

To clean up the complication of hard-coded $\pm$s, we can set $x_j = \text{Sign}(\mu_j - c_j) \cdot z_j = \kappa_j z_j$, where we now have binary variables z_j. Correspondingly,

$$\mathbf{v}_{\text{new}} = \mathbf{b}_{\text{op}} + \sum_{j=1}^{n} \kappa_j z_j \mathbf{d}_j \ . \tag{17}$$

The cost function in the quadratic unconstrained binary optimisation (QUBO) problem is constructed according to the Euclidean distance between the new vector $\mathbf{v}_{\text{new}}$, defined by the bit-string $z_1, \ldots, z_n$, and $\mathbf{t}$:

$$C(z_1, \ldots, z_n) := \left\| \mathbf{t} - \mathbf{b}_{\text{op}} - \sum_{j=1}^{n} \kappa_j z_j \mathbf{d}_j \right\|^2 \ . \tag{18}$$

Any QUBO problem of the form $\sum_i h_i z_i + \sum_{i<j} J_i z_i z_j$ can be expressed by an Ising Hamiltonian—an energy observable over a system of spin-$1/2$ particles [34]—of the form $-\sum_i h_i \sigma_i^z - \sum_{i<j} J_i \sigma_i^z \sigma_j^z$, where σ_i^z is the Pauli-Z operator applied on the ith qubit [70]. For our cost in equation (18), the Hamiltonian $\hat{H}_C$ can be obtained by directly mapping the binary variables z_j to the Pauli-Z terms $\hat{z}_j = (I - \sigma_j^z)/2$, giving

$$\begin{aligned} \hat{H}_C &= \left\| \mathbf{t} - \mathbf{b}_{\mathrm{op}} - \sum_{j=1}^{n} \kappa_j \hat{z}_j \mathbf{d}_j \right\|^2 \\ &= \sum_{i=1}^{n+1} \left| t_i - b_{\mathrm{op}}^i - \sum_{j=1}^{n} \kappa_j \hat{z}_j d_{j,i} \right|^2 , \end{aligned} \tag{19}$$

with any negation of terms absorbed into the κ_j terms to keep the formulation clean and closer in notation to that of Yan *et al* [32].

The energy states (eigenstates) of $\hat{H}_C$ correspond then to the vectors in the unit neighbourhood centred on $\mathbf{b}_{\mathrm{op}}$ (including $\mathbf{b}_{\mathrm{op}}$ itself), with the corresponding energies (eigenvalues) given by their distance to $\mathbf{t}$. As such, we have formulated a minimum eigenstate problem that may be solved by QAOA (see section 2.3).

The number of qubits required to refine the approximation is linear in the dimension of the lattice. Whether this is sufficient to yield a significant enough improvement to lead to efficient sieving is another question, and one that we will briefly explore in section 5. However, it is far more common to search with $O(n\log n)$ resources, and criticism of Yan *et al* [32] indicate that the linear search space is not enough [34–36].

4.3. QAOA pre-training to obtain fixed angles

In this work, we focus on whether the search of Yan *et al* [32] by QAOA can be made more efficient by a simple yet robust pre-training scheme, accompanying this with extensive and novel numerical analysis of time/query complexity in section 5—noting that in the original reference no analysis of the time-complexity was performed. Of course, since we derive the same search, we also derive the same flaws with respect to factoring (see section 3.2) and with the above search space and lattice density concerns. However, the generality and practical relevance of our work in providing fixed-angles to this search is not lost: the success of fixed angles in this context will be largely independent from the density of the lattice and size of the search space, up to precise time-complexity measures.

We take an approach to finding a good set of angles in the QAOA inspired by the search for parameters that has become synonymous with machine learning. Our simple yet robust pre-training algorithm is presented in algorithm 1.

The high-level intuition is as follows: train a collection of sets of angles, each on their own CVP instance drawn randomly from a training distribution, then evaluate how effective each set is at limiting the decay of the probability to sample the best solution on several random CVP instances drawn from a validation distribution. This is designed to find the set of angles best able to scale to larger problem instances, rather than is best at exploiting the nuances of a small training set.

4.3.1. Notable design choices

The issue of 'overfitting' has been given careful consideration. A great source of attraction to this method is the ability to find a set of angles on small instances that may be solved relatively efficiently to save the expense in larger instances. However, finding a set of angles that is *too* good for the smaller problems may not generalise well to larger problems, and so we must give great care to noticing when this is becoming the case in our search. This is where our method will greatly diverge from that of existing methods in Brandao *et al* [40] or Boulebnane and Montanaro [39].

In each training instance, we initialise a set of angles from the best known angles at that time. We find that this works to make efficient the training loop where a head start can be offered. Our direct addressing of overfitting via a validation loop acts as mitigation for the potential that this 'head start' may introduce.

We also make the explicit choice to work on the probability to sample *the best* solution, rather than any solution improving on $\mathbf{b}_{\mathrm{op}}$. This is done to: (1) avoid the unnecessary lattice reduction and computation of $\mathbf{b}_{\mathrm{op}}$ in each validation instance; (2) ensures our angle sets tune the QAOA to find good solutions regardless of their numerosity (e.g. not biasing a set of parameters which are tuned on instances for which an atypical proportion of solutions are better than $\mathbf{b}_{\mathrm{op}}$ by happenstance); and (3) avoid the complications that arise when $\mathbf{b}_{\mathrm{op}}$ is already the best solution in the neighbourhood.

4.4. Possible extensions and improvements

This work presents a simple proof-of-concept for the use of pre-training in QAOA-based lattice cryptography. Future endeavours therefore have a plethora of extensions to be made to algorithm 1, including but not limited to:

Algorithm 1. Pre-training for QAOA angles.

Input: Training distribution $\mathcal{T}$ and set size s_t, validation distribution $\mathcal{V}$ and set size s_v, precision parameter c
Output: Optimal array of angles $\mathbf{a}_{op}$

Initialise optimal array of angles $\mathbf{a}_{op} \leftarrow \mathbf{0}$

for some number of epochs **do**
  Initialise population of arrays of angles $A \leftarrow \{\}$

  for $\_ \leftarrow 1, \ldots, s_t$ **do**
    Draw instance size $m \sim \mathcal{T}$
    Set $n \leftarrow m/\log m$

    Construct the prime lattice $\mathcal{L}(B_{n,c})$
    Sample m-bit N and define $\mathbf{t}$ accordingly

    Initialise array of angles $\mathbf{a} \leftarrow \mathbf{a}_{op}$
    Optimise $\mathbf{a}$ for CVP$(\mathcal{L}, \mathbf{t})$ by QAOA
    Update $A \leftarrow A \cup \{\mathbf{a}\}$
  end for

  Initialise best scaling $\alpha^* \leftarrow \infty$

  for array $\mathbf{a} \in A$ **do**
    Initialise set of data points $D \leftarrow \{\}$

    for $\_ \leftarrow 1, \ldots, s_v$ **do**
      Draw instance size $m \sim \mathcal{V}$
      Set $n \leftarrow m/\log m$

      Construct the prime lattice $\mathcal{L}(B_{n,c})$
      Sample m-bit N and define $\mathbf{t}$ accordingly

      Obtain set of vectors and probs $(\mathbf{v}_i, p_i)_{i=0,\ldots,2^n}$ in the unit neighbourhood of $\mathbf{t}$ in $\mathcal{L}$
      Initialise best distance $d^* \leftarrow \infty$ and prob $p^* \leftarrow 0$

      for $\mathbf{v}, p \in (\mathbf{v}_i, p_i)_{i=0,\ldots,2^n}$ **do**
        Compute distance $d \leftarrow \|\mathbf{t} - \mathbf{v}\|^2$
        if $d < d^*$, update $d^* \leftarrow d$ and $p^* \leftarrow p$
      end for

      Update $D \leftarrow D \cup \{(n, p^*)\}$
    end for

    Find α such that $p(n) = 1/2^{\alpha n}$ is best-fit over D
    if $\alpha < \alpha^*$, update $\alpha^* \leftarrow \alpha$ and $\mathbf{a}_{op} \leftarrow \mathbf{a}$
  end for
end for

- *Cross-validation* to make for more efficient usage of data during an overfitting-aware scheme;
- *Evaluating by cost* rather than by the probability to sample the minimum eigenstate, which may be more effective in larger neighbourhoods with greater variance of solution quality;
- *Evolutionary algorithms* considering the population of arrays of angles A as 'evolving' over time, with the possibility for interaction, competition, and mutation between set of angles.

5. Experiments and results

5.1. Pre-training and angle convergence

Ahead of our experimentation, we pre-train p-deep QAOA circuits according to algorithm 1 for $p = 1, \ldots, 10$. The validation performances at each epoch are shown in figure 3, showing general improvement for increasing p.

Separately, we train a large number of $p = 1$ layer QAOA circuits independently for random CVP instances and make note of the obtained values for β and γ. We may plot these by the instance size, as shown in figure 4, to observe the convergence of the angles for growing problem complexity. An indicative optimisation (for a small instance) landscape is exemplified by figure 5, though the landscape

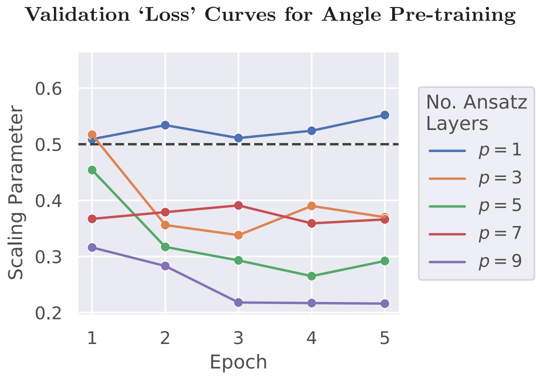

Figure 3. Validation performance during the pre-training of p-depth QAOA circuits. The 'loss' being minimised is the scaling parameter α in the curve $q(n) = 1/2^{\alpha n}$ over a dataset of validation points (n_i, q_i). The dashed line denotes the speed-up offered by Grover's algorithm [60].

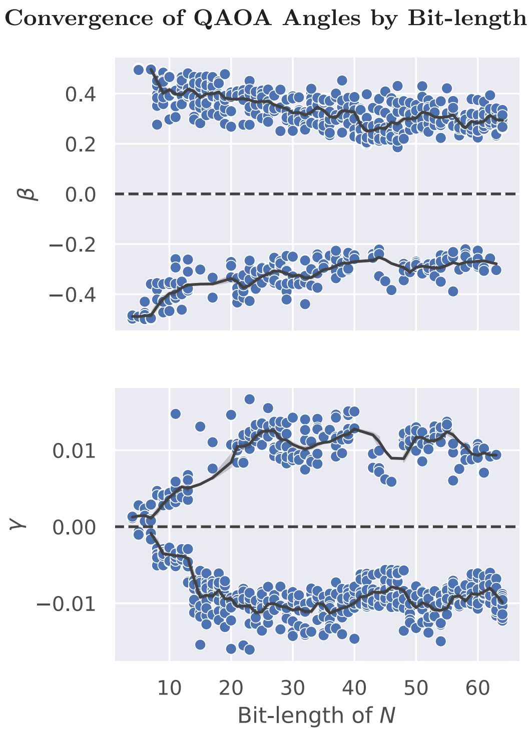

Figure 4. Convergence of the angles in a $p = 1$ layer QAOA over different instances of varying bit-length.

becomes increasingly flat as the problem complexity (here, lattice dimension) grows—the dreaded 'barren plateau' phenomenon [71–75].

From figure 4, we gain confidence that a stable set of angles can be easily learnt, and that they will have the capacity to scale to larger instances without the necessity to re-train nor fine-tune.

Figure 4 also highlights the issue of overfitting extremely well. Suppose we had followed a method similar to that of Brandao *et al* [40] and selected a single small instance on which to find our angles. The angles found on these smaller problems are not yet converged, unlike with later cases, and so are less likely to generalise well. Then any small instance we select will provide angles that are suboptimal in general. Hence, our scheme is advantageous in being aware of convergence and overfitting.

5.2. Obtaining statistics

For our experiments, we conduct a broad numerical analysis to determine the trending relationship between refinement probability and lattice dimension, which then implies time-complexity via the expected number of queries to the QAOA circuit. Cases wherein the approximately found solution already represents the best solution in the unit neighbourhood are omitted.

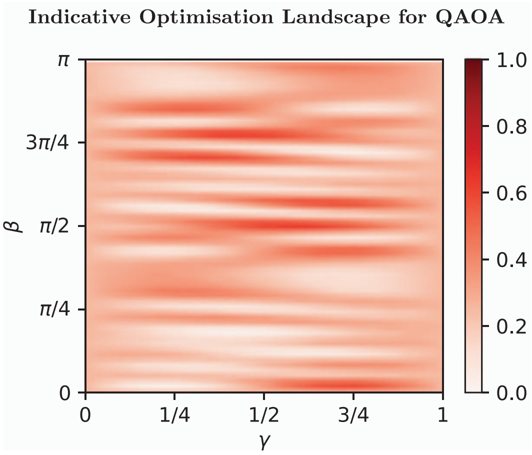

Figure 5. Plotting the the probability to successfully refine the solution (indicative of the underlying optimisation landscape) over a pair of parameters (γ, β) in a $p = 1$ layer QAOA for a random three-qubit problem instance.

First, generate a CVP for the n-dimensional lattice $\mathcal{L}(B_{n,c})$ for an m-bit composite integer N, where $n = m/\log m$ as specified by Schnorr [31]. The method in Yan *et al* [32] is only exemplified for three cases, however we simulate on the order of tens of thousands of cases so that an accurate scaling curve may be plotted to indicate asymptotic runtime.

We implement the QAOA solving a minimum-eigenstate problem for the Hamiltonian $\hat{H}_C$, as detailed in section 4.2, to obtain an outcome measurement $|\psi\rangle$ representing a binary string ψ deciding whether to 'step' in each of the reduced basis directions $D = [\mathbf{d}_1, \ldots, \mathbf{d}_n]$ from $\mathbf{b}_{\text{op}}$. Hence, we may translate any $|\psi\rangle$ into the corresponding lattice vector $\mathbf{v}_{\text{new}}$ by performing $\mathbf{b}_{\text{op}} + \kappa \circ \psi D$, where $\circ$ denotes element-wise multiplication.

The probability to refine the solution (i.e. sample an improvement over $\mathbf{b}_{\text{op}}$) is ascertained by aggregating the probabilities for each $|\psi\rangle \mapsto \mathbf{v}_{\text{new}}$ for which $\|\mathbf{t} - \mathbf{v}_{\text{new}}\|^2 < \|\mathbf{t} - \mathbf{b}_{\text{op}}\|^2$. This is the statistic whose decay we work to reduce with increasing lattice dimension n.

5.3. Complexity analysis for the refinement

Obtaining statistics for greatly many instances produces a dataset of points (n_i, q_i), where $n_i = \log N_i / \log\log N_i$ is the 'exact' lattice dimension computed directly from the composite integer N_i, and q_i is the estimated probability to refine the approximation (obtained classically by the method in section 4.1). From q_i, we can expect to make $1/q_i$ queries to the circuit to yield the desired solution.

These statistics are obtained by QAOA circuits of depths $p = 1, \ldots, 10$ and plotted in figure 6. In each case, we consider bit-lengths $4 \leqslant m \leqslant 128$, and thus lattice dimensions $3 \leqslant n \leqslant 18$. This is substantially larger than is considered in Yan *et al* [32], and should be large enough to reveal any scalability concerns [34–37].

Our optimal scaling is obtained, unsurprisingly, by $p = 10$ layers, relating the refinement probability to lattice dimension as $q(n) \approx 1/2^{0.225n}$, and thus indicating a time-complexity scaling as $O(2^{0.225n})$. This is more than a quadratic speed-up over the famous Grover's algorithm [60], with far shallower depth and without requirement for fault tolerance. These findings mimic that of Boulebnane and Montanaro [39] for the improvement over Grover by fixed angles for QAOA. In fact, we find improvement with any depth $p > 2$.

Our results indicate a promising relationship between the scaling parameter α, which characterises the degree of the exponential decay by $1/2^{\alpha n}$, and the lattice dimension n. Figure 7 fits an exponential curve to this relationship and extrapolates to greater depths. Optimistically, we estimate that the time complexity will reduce to $O(2^{0.1n})$ for $p > 20$.

To observe longer-range statistical performance of our fixed-angles approach, we extend the experiment for $p = 10$ ansatz layers to push up to lattice dimension of $n \leqslant 28$, shown in figure 8. Plotting a best-fit curve to this longer-range data obtains the same $\alpha = 0.225$ exponent, though a moderate increase in scedasticity may be present with $n \gg 20$. The stability of this fit to greater n—much greater than those seen during training or validation of the angles—is good precedent for a highly scalable approach to CVP refinement.

Experimentation greatly beyond these dimensions towards cryptographically-significant sizes, say in the hundreds, is infeasible to classically simulate, particularly to conduct numerical analyses of the

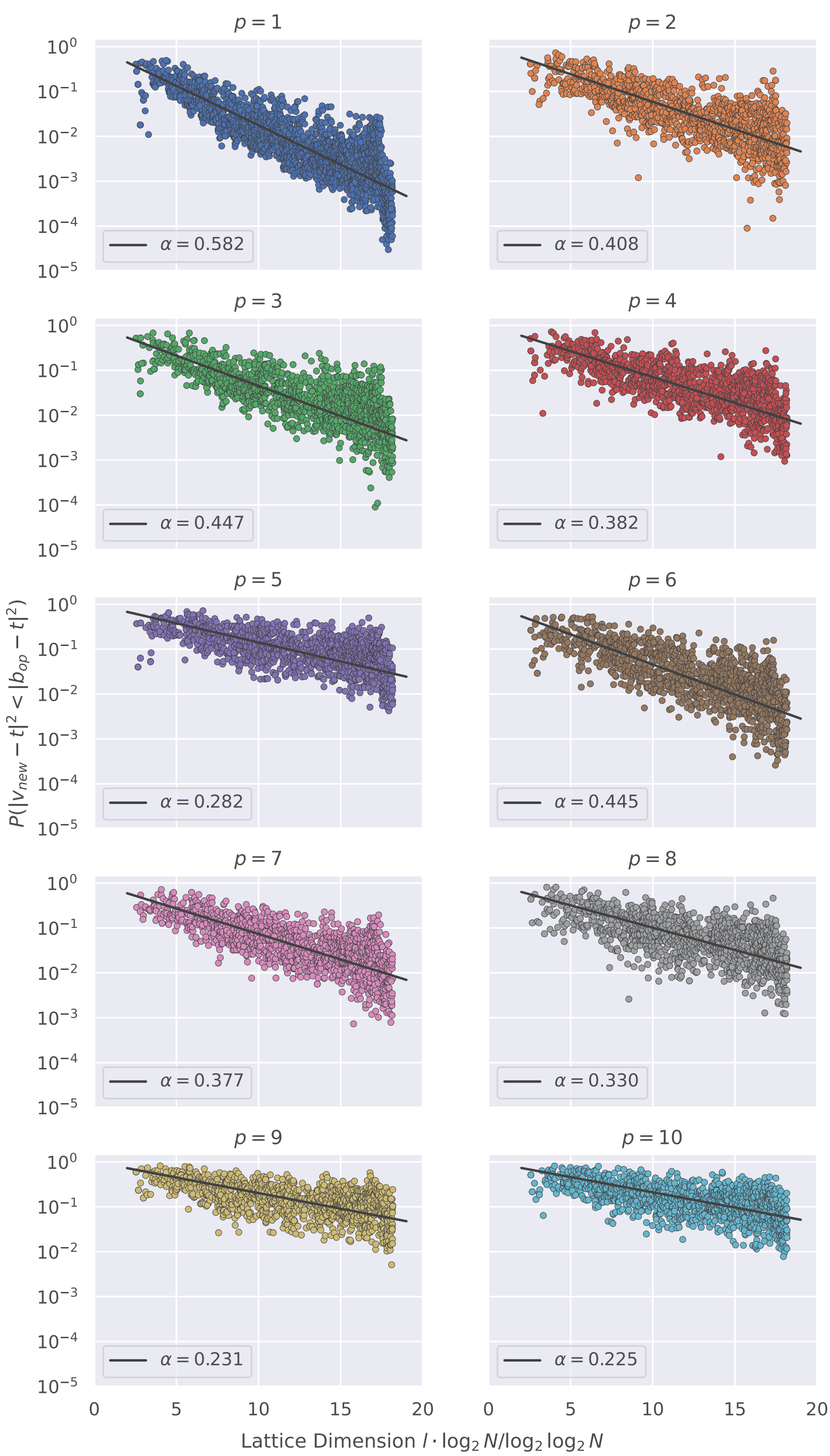

Figure 6. The probability to refine the classical solution (for cases in which a refinement exists) by exact lattice dimension for different depths p of QAOA circuits. Each plot is equipped with a best-fit curve of the form $1/2^{\alpha n}$, and α is shown.

style this work presents. However, we should reiterate that the scalability concerns we face here due to classical simulability *do not* hold for quantum computers; with access to a quantum computer with more qubits (in particular, we would need $O(d)$ logical qubits to address d-dimensional lattice), we can provide the QAOA circuit and obtained angles for some p and query without such concerns of computational feasibility. In fact, the angles we provide can be exactly those used in the experiments presented in this work, obtained through classical simulation, with no need to perform pre-training with

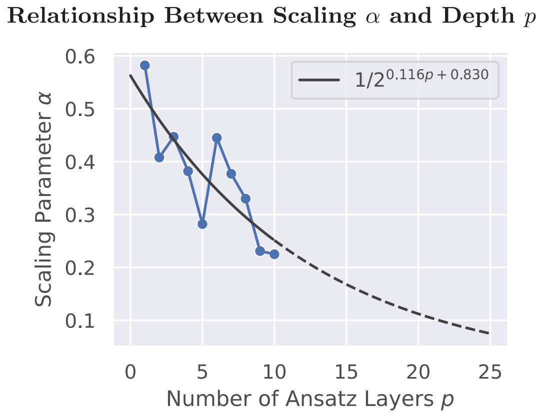

Figure 7. Extrapolating the relationship between the optimal scaling obtainable at each QAOA depth p, taken from figure 6.

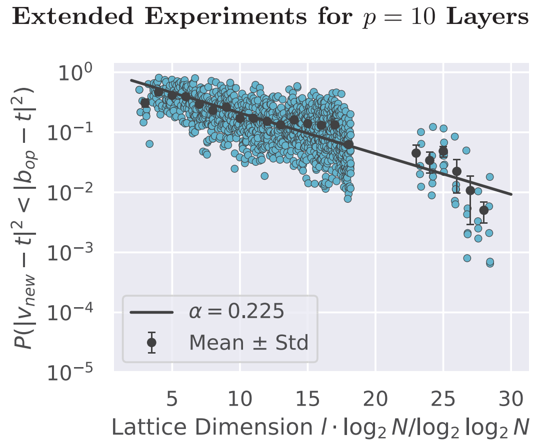

Figure 8. Extended simulation statistics for $p = 10$ ansatz layers (bottom right plot of figure 6), up to lattices of dimension $n \leqslant 28$. The scaling parameter α is obtained to be equal, up to 3 s.f., as in figure 6.

real quantum hardware. The query complexity scales with the reciprocal of the probabilities given in figures 6 and 8, and our ability to refine a poly-time classical solution demonstrates a capacity to reduce bit security.

5.4. Refinement quality

We will take brief notice of the scaling of the *quality* of the refinement in figure 9. While the improvement may appear small, we must note that *any* improvement is significant due to the discrete nature of $\mathcal{L}$—an improvement indicates that an entirely different sr-pair has been identified, which comes with a different likelihood for usefulness in cryptography.

The quality of available solutions with this method is a primary concern in Grebnev *et al* [34], and is noted by other literature [35, 36]. We consider data for which a refinement exists (i.e. situations in which the unit neighbourhood around $\mathbf{b}_{\mathrm{op}}$ contains a better solution), and note a subexponential decay of quality in figure 9. However, it is still unlikely that this quality is sufficient for sieving and thus factoring.

We are optimistic that, given our findings for the potential quantum advantage in reducing the decay of refinement probability, the search space may be increased by dropping Schnorr [31]'s contentious sublinear lattice dimension scheme.

5.5. Exceeding the grover-type speedup

Our results mirror that of Boulebnane and Montanaro [39] in the significant improvement over naive use of Grover's algorithm to conduct the search. This is non-trivial to understand, since the quadratic speedup of Grover's is well-known to be optimal, for query models, with NP-hard problems like CVP. So let us now see the intuition to understand why QAOA can outperform Grover for the special case structures in this work.

Figure 9. The improvement that the best solution in the neighbourhood obtains over $\mathbf{b}_{\mathrm{op}}$ with increasing lattice dimension. A rolling average is given and a best-fit curve is shown.

Principally, *we do not use an oracle, we use structure*. By encoding a neighbourhood of refined solutions into a problem Hamiltonian, we establish a *structured* search problem for approximately solving our CVP. More than this, the lattice itself comprises a relatively consistent set of basis vectors, up to permutations along the diagonal, which encode another explicit structural bias into our problem (and parameters). In this way, the setting for our QAOA refinement method is not a purely unstructured search problem in which Grover's algorithm would be optimal.

There is also a conceptual distinction to be made in the means by which 'solutions' are obtained. Instead of the typical Grover-like setup of marked solutions states, ours is a (relatively) smooth energy landscape defined by distance to a known target. The dynamics of the QAOA is then to bias towards sampling from low-energy subspaces of the landscape, allowing us to find solutions without such a strong necessity to amplify any one particular state. So, under the assumption that good angles can preclude classical optimisation loops, there is good intuition supporting the potential to improve on Grover, since the marked set defining our problem presents with dense pockets of market solution states in non-arbitrary arrangements that may be exploited variationally.

5.6. Contrasting with no pre-training

This section presents some prospective results for our scaling analysis with alternative methods for angle selection. This serves to contextualise the success of our proposed pre-training scheme.

5.6.1. Random instance

The most straightforward way to obtain angles—short of drawing them randomly—is to train on a single (random) instance and take the consequent angles. This is the impression left by Brandao *et al* [40].

Ideally, the single instance we choose is small enough that the angles may be obtained relatively efficiently, but large enough that the resultant angles are general (see figure 4 to get a sense for the convergence of the angles by instance size).

Figure 10 gives the time complexity (mimicking the subplots in figure 6) one might expect to obtain in general having pre-trained by a single random instance at the indicated size. This is roughly inline

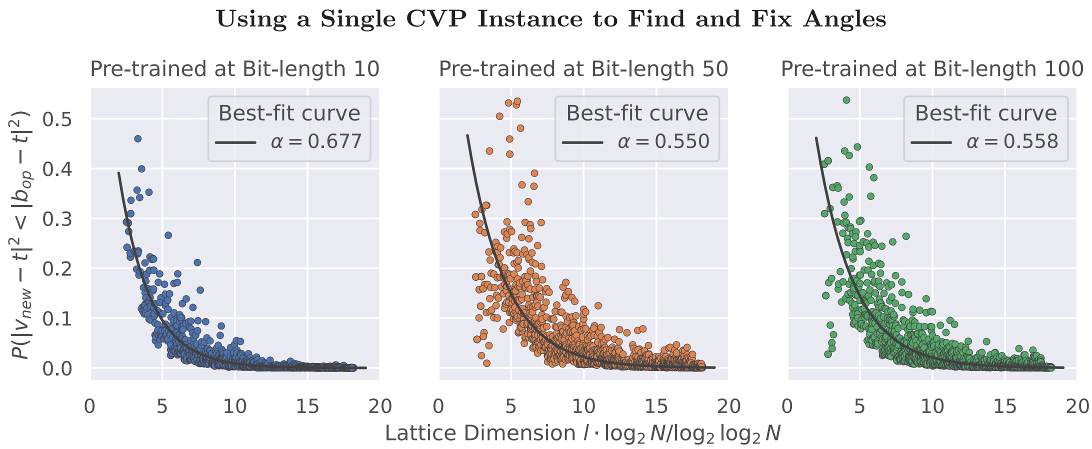

Figure 10. Probability to refine the approximate solution by lattice dimension for three distinct QAOA circuits with angles specified by a single (random) instance. The size of these instance (given in bit-length of N) is indicated above each plot.

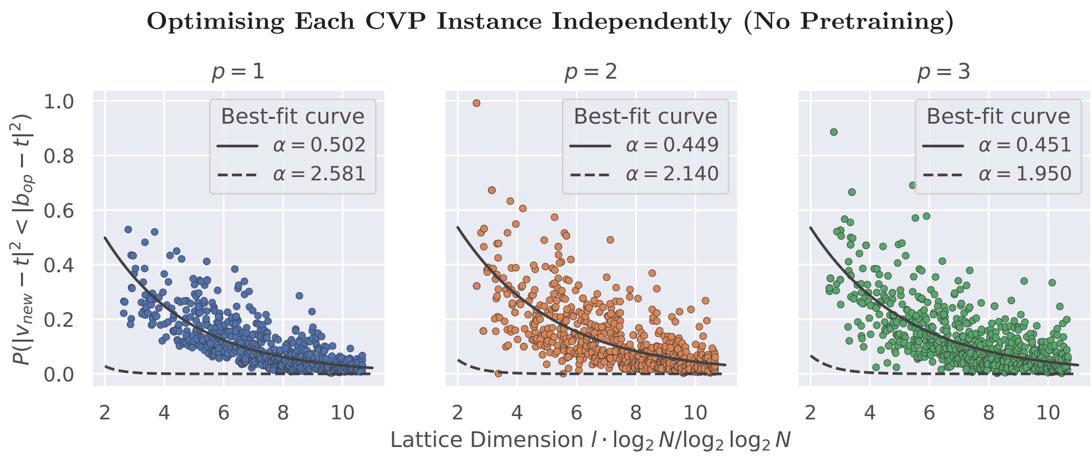

Figure 11. Similarly to figure 10, though without any pre-training scheme. That is, each instance of the QAOA obtains its angles independently. We show here also a dashed line representing the scaling when also considering cases in which no refinement exists within the neighbourhood.

with our expectation that bit-lengths <20 are ineffective for obtaining generalisable angles. We further notice that increasing bit-length does not appear to continue reducing the probability decay.

Our preliminary conclusion is that pre-training by a single random instance has the limit of Grover's speed up ($\alpha \approx \frac{1}{2}$). More instances are required to gather the information necessary to find general angles, as in our proposed scheme in section 4.

5.6.2. No pre-training (instance-optimised)

We would also like to show experimental results from a lack of pre-training altogether. Here, we obtain the angles in each instance independently, as is currently standard in QAOA.

This, of course, results in immense computational effort when considering greatly many instances as we do, hence we are unable to explore to the largest lattice dimensions. Already, this is a decisive difference between a fixed angle scheme and the absence thereof—when you are happy to omit the optimisation of angles, we can set out immediately querying the circuit.

Moreover, we note that extrapolating the performance for larger instances, may be less reliable, if one uses optimisation to obtain the suitable angles. The reason is that as dimension increases, the energy landscape changes—the energy gaps vary, eventually leading to a barren-plateau type of landscape. This means that there may be a scale, beyond what we can classically simulate, that the barren plateau effect kicks in, and in that scale we would see an dramatic drop of the probability, not following the trend of the decay observed in the much smaller instances we consider. This problem, importantly, is not present in the fixed angles approach, since there will be no optimisation performed to find 'optimal' angles for those large instances.

The prospective results, shown in figure 11, are better than for random instance pre-training, though have a similar apprehension for continued improvement. The improvement over random instance pre-training is expected (after all, each instance finds its own 'perfect' angles), but an overall lack of improvement over our own pre-training scheme (see figure 6) is surprising. Indeed, exploring to greater depths may be the deciding factor in demonstrating an ability to withstand the exponential decay of probability.

Again, we conclude that pre-training is an indispensable tool in the usage of QAOA, and leads to the potential for a quantum advantage that we highlight in this work.

6. Conclusion

In this work, we proposed a simple yet robust pre-training algorithm to use for fixed-angles QAOA. The method has been used on lattice-based problems, and enabled a heuristic analysis of the time-complexity of QAOA for sieving on the prime lattice [29–31].

Extending from the method in Yan *et al* [32] for refining approximations to the CVP as a reduction for the sieving problem, we indicate the possibility for a quantum advantage in searching for refinements via a fixed angle scheme for QAOA. In doing this, we hope to reveal the threat posed by newer variational approaches to lattice problems in cryptography.

Our results demonstrate that, with such a fixed-angles scheme, we have a quantum advantage in obtaining an approximate solution to the CVP faster than any classical enumeration targeting the same degree of approximation (see e.g. [76–78]), with a linear space requirement; our $\alpha = 0.225$ significantly reduces typical single-exponent enumeration exponents in the range of 0.3–0.4 [76].

Further work should explore whether this potential advantage persists as the search space for refinement grows—say, for a neighbourhood encoding in $O(n \log n)$ qubits rather than $O(n)$. If the advantage is not lost in larger spaces, then lattice problems may be sufficiently challenged by these heuristic search methods, provided a set of angles can generalise well across instances. For a large enough neighbourhood (which is, in itself, not trivial to determine), there may be potential to find *exact* solutions with a reasonable exponent in the time-complexity, though we expected that a highly constrained lattice should be assumed for such optimism. The prime lattice may be sufficiently constrained for explorations in this direction.

6.1. Limitations and opportunities for future work

Beyond the limitation due to insufficient search space, which has been discussed previously in connection with Yan *et al* [32] and Schnorr [31] regarding their 'sublinear factoring' claim, there are two limitations in this work that it are worth reiterating as a motivation for future research.

6.1.1. Prime lattice and transfer learning

First, we only consider CVPs constructed on *the prime* lattice, as by their very nature (see section 4), they are focused on sieving for sr-pairs. This gives our work a strong flavour of cryptanalysis, which serves as the main interest for finding a quantum advantage for lattice problems, though it may be argued to lack generality. Indeed, our success in finding a good set of angles may not be replicated with ease in more general CVPs, in which the structure has greater variance.

With recent analyses from Montañez-Barrera and Michielsen [61] in mind, we optimistically suggest that the parameter set used to obtain our strong performance in figures 6 and 8 could be 'transferred' [63] to general CVP instances. We leave this experimentation, and application to other interesting lattice problems, as an open problem worth exploring as the literature on fixed-angles VQAs continues to evolve.

Whether pre-trained angles on the prime lattice can be transferred to general lattice problems is an extremely interesting and important question. The most exciting answer to consider is that we *can* transfer effectively, in which case this work opens the door to a new paradigm for training VQAs; by constructing a highly structured special-case instance of the problem we have interest in solving (e.g. CVPs on the prime lattice), we can easily find and fix a set of angles by our algorithm 1 on small instances to then be transferred efficiently to the harder problem (e.g. CVPs on general lattices). This may be one of the most promising new avenues for avoiding the barren plateau problem [74, 75, 79, 80]; for example, we could imagine utilising the transferability of fixed angles in conjunction with results on classical simulability in VQAs [81] to formulate a classical means for avoiding barren plateaus pre-emptively (i.e. classically find angles on a small, structured problem, then apply to non-simulable larger problems without trainability issues). We highly encourage future work answer this question, and in particular understand the utility for classical simulation to obtain quantum advantage at computationally large scales.

6.1.2. Noiseless simulation

Second, all results in this work are based on *ideal simulations* and the impact that noise may have on the effectiveness of pre-training have not yet been observed. Our algorithm, even compared to fault-tolerant algorithms such as Grover's, requires fewer logical qubits, has smaller quantum depth, and the overall success probability is polynomially better than Grover's search. This means that our results can be viewed as an improvement even for error-corrected algorithms, putting our approach well in the 'early-fault-tolerant' era. However, addressing the effect of noise and the performance of our approach as a genuinely NISQ era approach (without error-correction) is also important.

On the one hand, we may find that NISQ-era constraints confound the process of pre-training, and that using static parameters has the potential to exacerbate random noise in large instances. On the other hand, the approach outlined in this work has many qualities that should make it relatively robust to noise among other near-term quantum algorithms:

- *Circuit depths are shallow*. Choosing the number p of ansatz layers to taste, and noting that the depth of a single ansatz layer is at most polynomial in the width n [33] (in this case, lattice dimension), we have a quantum circuit scaling as $O(\mathrm{poly}(n,p))$. One motivating advantage for VQAs in general is that they can be effectively used at far shallower depths than traditional fault-tolerant algorithms (such as Grover's), having a well-understood mitigating effect on noisy hardware [44].
- *Lattice structures are controlled and randomised at train-time*. Our CVP instances populating our training set are randomly permuted along the diagonals of their basis matrices, all of which inherit from the prime lattice family. This gives a certain predictability and a notion of 'average case' to our training set that undermines any relatively small random noise in the long run. With enough epochs, we would still expect good convergence of the found parameters, given the impact of the noise does not dilute the underlying structural details of the CVP.
- *Parameter pre-training optimises for generalisability*. The loss we employ in our parameter pre-training regime is defined by the performance of the parameters on an unseen validation set of larger instances. Therefore, the parameter sets obtained by our method are those naturally inclined to generalise well. In the presence of NISQ-era constraints, with noisy hardware amplifying the random errors in larger instances, we would expect our found parameters to tolerate unseen noise relatively well on average. In some sense, an incorporation of small-scale noise in the small-instance training set in collaboration with the loss we describe above is a strong philosophy to adopt with near-term algorithms to bake a sense a error-bias directly within the parameters of our ansatz.

An exploration, empirical or otherwise, into whether the properties of our algorithm outlined here have the expected effect with respect to hardware noise would be interesting. In particular, we leave it to future work to understand whether such a loss defined by generalisability and scalability is more robust than: (i) random parameters; (ii) fixed angles taken from a single instance; and (iii) instanced-optimised. An improvement in robustness is certainly expected over (i); it may be improved in (ii) when we evaluate on instances of a larger size to the given single instance; but should not be more robust than (iii) for relatively small random noise, though it may be true that robust fixed angles are at least as robust to noise than any parameters obtained on a particular instance *when noise is present between classical optimisation loops*. Such a robustness result for fixed angles would sign-post a key algorithmic design choice for variational algorithms of practical concern.

Acknowledgments

PW acknowledges support by EPSRC Grants EP/T001062/1 and EP/Z53318X/1.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI:. https://github.com/BenPrie/qaoa-for-cvp [93].

Appendix A. Lattice reduction

The difficulty of any lattice problem is dictated in large part by the 'quality' of the given basis B. A 'good' basis is one consisting of short, relatively mutually orthogonal vectors, making navigation precise and intuitive. On the other hand, a 'bad' basis consists of long, relatively mutually parallel vectors that confound the method of walking toward particular points by combinations of basis vectors. This intuition leads to public-key cryptosystems on lattices, for which a pair of good and bad bases imply private and public keys.

Ideally then, we should like to start with a good basis, even if we are given a bad one to work from. The process of making a given basis 'better' is referred to as *lattice reduction*. Useful literature for developing an intuitive understanding for lattice reduction algorithms in cryptanalysis include Joux and Stern [82], Nguyen and Stern [83], and Bremner [84] in particular for an introduction.

In this work, we consider the famous LLL-reduction algorithm due to Lenstra *et al* [69]. For convenience, we give a brief description here. Useful texts include those aforementioned, or Wübben *et al* [85]. Discussion here draws also from Schnorr [31].

Definition A.1 (QR-decomposition). Any basis matrix B has the unique decomposition $B = QR$, where $Q \in \mathbb{R}^{n\times m}$ is isometric (with pairwise orthogonal column vectors of unit length) and $R = [r_{i,j}]_{1\leqslant i,j\leqslant m} \in \mathbb{R}^{m\times m}$ is upper triangular with positive diagonal entries $r_{i,i}$.

Furthermore, $R = \mathrm{GNF}(B)$ is the *generic normal form* of B, whose Gram–Schmidt coefficients $\mu_{j,i} = r_{i,j}/r_{i,i}$ are rational for integer matrices.

Definition A.2 (LLL reduction). A basis B is δ-LLL reduced, if $|\mu_{i,j}|$ for all $i<j$, and $\delta r_{i,i}^2 \leqslant r_{i,i+1}^2 + r_{i+1,i+1}^2$ for $i = 1,\ldots,n-1$.

We enforce that $\frac{1}{4} < \delta \leqslant 1$. Lenstra *et al* [69] show that any basis can be δ-LLL reduced for $\delta < 1$ in polynomial time, and that they approximate the successive minima well.

Appendix B. Implementation details

Our work was produced in Python (version 3.8.18), and is available at [42].

LLL-reduction [69] and Babai's nearest plane algorithm [47] are implemented via FPyLLL (version 0.6.1; [86]), a Python interface for the lattice algorithm library FPLLL (version 5.4.5; [87]). This further relied on NumPy (version 1.22.3; [88]).

For the quantum portion of our implementation, we use Google Cirq (version 1.1.0; [89]) and the accompanying qsimcirq (version 0.18.0; [90]) for simulation. Unfortunately, Cirq does not provide an implementation of quantum approximate optimisation algorithm (yet), so we have derived our own implementation from Khattar and Yosri [36].

For the results presented in this work, we use the Nelder–Mead simplex algorithm [91] for minimisation of the expectation values of the observables in our circuit via SciPy (version 1.10.1; [92])

ORCID iDs

Ben Priestley 0009-0001-5894-574X
Petros Wallden 0000-0002-0255-6542

References

[1] Rivest R L, Shamir A and Adleman L 1978 A method for obtaining digital signatures and public-key cryptosystems *Commun. ACM* **21** 120
[2] Zhang D, Wang H, Li S and Wang B 2024 Progress in the prime factorization of large numbers *J. Supercomput.* **80** 11382–400
[3] Shor P W 1995 Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer *SIAM Rev.* **41** 303
[4] Lucero E *et al* 2012 Computing prime factors with a Josephson phase qubit quantum processor *Nat. Phys.* **8** 719–23
[5] Lanyon B P, Weinhold T J, Langford N K, Barbieri M, James D F V, Gilchrist A and White A G 2007 Experimental demonstration of a compiled version of Shor's algorithm with quantum entanglement *Phys. Rev. Lett.* **99** 250505
[6] Lu C-Y, Browne D E, Yang T and Pan J-W 2007 Demonstration of a compiled version of Shor's quantum factoring algorithm using photonic qubits *Phys. Rev. Lett.* **99** 250504
[7] Martín-López E, Laing A, Lawson T, Alvarez R, Zhou X-Q and O'Brien J L 2012 Experimental realization of Shor's quantum factoring algorithm using qubit recycling *Nat. Photon.* **6** 773–6
[8] Bernstein D J and Lange T 2017 Post-quantum cryptography *Nature* **549** 188
[9] Goldreich O, Goldwasser S and Halevi S 1997 Public-key cryptosystems from lattice reduction problems *Advances in Cryptology—CRYPTO'97: 17th Annual Int. Cryptology Conf. (Santa Barbara, California, USA August 17–21, 1997) (Proc. 17)* (Springer) pp 112–31

[10] Hoffstein J, Pipher J and Silverman J H 1998 NTRU: a ring-based public key cryptosystem *Int. Algorithmic Number Theory Symp.* (Springer) pp 267–88
[11] Hoffstein J, Pipher J and Silverman J H 2001 NSS: an NTRU lattice-based signature scheme *Advances in Cryptology—EUROCRYPT 2001: Int. Conf. on the Theory and Application of Cryptographic Techniques (Innsbruck, Austria, May 6–10, 2001) (Proc. 20)* (Springer) pp 211–28
[12] Hoffstein J, Howgrave-Graham N, Pipher J, Silverman J H and Whyte W 2003 NTRUSIGN: digital signatures using the NTRU lattice *Cryptographers' Track at the RSA Conf.* (Springer) pp 122–40
[13] Lyubashevsky V 2012 Lattice signatures without trapdoors *Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques* (Springer) pp 738–55
[14] Ducas L, Durmus A, Lepoint T and Lyubashevsky V 2013 Lattice signatures and bimodal Gaussians *Annual Cryptology Conf.* (Springer) pp 40–56
[15] Bernstein D J, Chuengsatiansup C, Lange T and van Vredendaal C 2018 NTRU prime: reducing attack surface at low cost *Selected Areas in Cryptography–SAC 2017: 24th Int. Conf., (Ottawa, ON, Canada, 16–18 August 2017) Revised Selected Papers 24* (Springer) pp 235–60
[16] Coppersmith D and Shamir A 1997 Lattice attacks on ntru *Int. Conf. on the Theory and Applications of Cryptographic Techniques* (Springer) pp 52–61
[17] Nguyen P Q and Regev O 2006 Learning a parallelepiped: cryptanalysis of GGH and NTRU signatures *Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques* (Springer) pp 271–88
[18] Ducas L and Nguyen P Q 2012 Learning a zonotope and more: cryptanalysis of NTRUSIGN countermeasures *Int. Conf. on the Theory and Application of Cryptology and Information Security* (Springer) pp 433–50
[19] Laarhoven T 2015 Sieving for shortest vectors in lattices using angular locality-sensitive hashing *Advances in Cryptology–CRYPTO 2015: 35th Annual Cryptology Conf., (Santa Barbara, CA, USA, 16–20 August 2015) (Proc., Part I 35)* (Springer) pp 3–22
[20] Laarhoven T and de Weger B 2015 Faster sieving for shortest lattice vectors using spherical locality-sensitive hashing *Progress in Cryptology–LATINCRYPT 2015: 4th Int. Conf. on Cryptology and Information Security in Latin America, (Guadalajara, Mexico, 23–26 August 2015) (Proc. 4)* (Springer) pp 101–18
[21] Becker A, Ducas L, Gama N and Laarhoven T 2016 New directions in nearest neighbor searching with applications to lattice sieving *Proc. 37th Annual ACM-SIAM Symp. on Discrete Algorithms* (SIAM) pp 10–24
[22] Alagic G *et al* 2022 Status report on the third round of the NIST post-quantum cryptography standardization process *CSRC* NIST IR 8413
[23] Computer Security Division, Information Technology Laboratory 2017 Post-quantum cryptography standardization—post-quantum cryptography: Csrc (available at: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization)
[24] Pomerance C 1984 The quadratic sieve factoring algorithm *Workshop on the Theory and Application of of Cryptographic Techniques* (Springer) pp 169–82
[25] Davis J A and Holdridge D B 1984 Factorization using the quadratic sieve algorithm *Advances in Cryptology: Proc. of Crypto 83* ed D Chaum (Springer) pp 103–13
[26] Lenstra A K and Lenstra H W 1993 *The Development of the Number Field Sieve* vol 1554 (Springer)
[27] Briggs M E 1998 An introduction to the general number field sieve *PhD Thesis* Virginia Tech
[28] Boudot F, Gaudry P, Guillevic A, Heninger N, Thomé E and Zimmermann P 2022 The state of the art in integer factoring and breaking public-key cryptography *IEEE Secur. Privacy* **20** 80
[29] Schnorr C P 1991 Factoring integers and computing discrete logarithms via diophantine approximation *Advances in Cryptology—EUROCRYPT '91* ed D W Davies (Springer) pp 281–93
[30] Schnorr C P 2013 Factoring integers by CVP algorithms *Number Theory and Cryptography: Papers in Honor of Johannes Buchmann on the Occasion of His 60th Birthday* vol 73
[31] Schnorr C P 2021 Fast factoring integers by SVP algorithms, corrected *Cryptology EPrint Archive, Paper 2021/933* (available at: https://eprint.iacr.org/2021/933)
[32] Yan B *et al* 2022 Factoring integers with sublinear resources on a superconducting quantum processor (arXiv:2212.12372 [quant-ph])
[33] Farhi E, Goldstone J and Gutmann S 2014 A quantum approximate optimization algorithm (arXiv:1411.4028)
[34] Grebnev S V, Gavreev M A, Kiktenko E O, Guglya A P, Efimov A R and Fedorov A K 2023 Pitfalls of the sublinear QAOA-based factorization algorithm *IEEE Access* **11** 134760–8
[35] Aboumrad W, Widdows D and Kaushik A 2023 Quantum and classical combinatorial optimizations applied to lattice-based factorization (arXiv:2308.07804)
[36] Khattar T and Yosri N 2023 A comment on "factoring integers with sublinear resources on a superconducting quantum processor" (arXiv:2307.09651)
[37] Ducas L 2021 Lducas/schnorrgate: testing Schnorr's factorization claim in sage (available at: https://github.com/lducas/SchnorrGate)
[38] Vera A I 2010 A note on integer factorization using lattices (arXiv:1003.5461)
[39] Boulebnane S and Montanaro A 2022 Solving boolean satisfiability problems with the quantum approximate optimization algorithm (arXiv:2208.06909 [quant-ph])
[40] Brandao F G S L, Broughton M, Farhi E, Gutmann S and Neven H 2018 For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances (arXiv:1812.04170 [quant-ph])
[41] Prokop M and Wallden P 2025 Heuristic time complexity of NISQ shortest-vector-problem solvers (arXiv:2502.05284 [quant-ph])
[42] Priestley B 2025 Code for the paper: a practical scalable approach to the CVP for sieving via QAOA with fixed angles (available at: https://github.com/BenPrie/qaoa-for-cvp)
[43] We note that this does not conflict any known results on the asymptotic optimality of Grover, since QAOA is not a "black-box" oracle algorithm and uses the structure of the problem (via the problem Hamiltonian) in the way the ansatz is constructed
[44] Cerezo M *et al* 2021 Variational quantum algorithms *Nat. Rev. Phys.* **3** 625
[45] Albrecht M R, Prokop M, Shen Y and Wallden P 2023 Variational quantum solutions to the shortest vector problem *Quantum* **7** 933
[46] Joseph D, Callison A, Ling C and Mintert F 2021 Two quantum Ising algorithms for the shortest-vector problem *Phys. Rev. A* **103** 032433
[47] Babai L 1986 On Lovász' lattice reduction and the nearest lattice point problem *Combinatorica* **6** 1

[48] Kraitchik M 1922 *Théorie des Nombres* vol 1 (Gauthier-Villars)
[49] Morrison M A and Brillhart J 1975 A method of factoring and the factorization of f7 *Math. Comput.* **29** 183–205
[50] Dixon J D 1981 Asymptotically fast factorization of integers *Math. Comput.* **36** 255
[51] Bennett H 2023 The complexity of the shortest vector problem *SIGACT News* **54** 37–61
[52] Regev O 2005 On lattices, learning with errors, random linear codes and cryptography *Proc. 37th Annual ACM Symp. on Theory of Computing (STOC '05)* (Association for Computing Machinery) pp 84–93
[53] Bennett H, Golovnev A and Stephens-Davidowitz N 2017 On the quantitative hardness of CVP *2017 IEEE 58th Annual Symp. on Foundations of Computer Science (FOCS)* (IEEE) pp 13–24
[54] Micciancio D 2001 The hardness of the closest vector problem with preprocessing *IEEE Trans. Inf. Theory* **47** 1212
[55] Micciancio D and Goldwasser S 2002 *Complexity of Lattice Problems: A Cryptographic Perspective* vol 671 (Springer)
[56] Farhi E, Goldstone J, Gutmann S and Sipser M 2000 Quantum computation by adiabatic evolution (arXiv:quant-ph/0001106)
[57] Farhi E, Goldstone J, Gutmann S, Lapan J, Lundgren A and Preda D 2001 A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem *Science* **292** 472
[58] Zhou L, Wang S-T, Choi S, Pichler H and Lukin M D 2020 Quantum approximate optimization algorithm: performance, mechanism and implementation on near-term devices *Phys. Rev. X* **10** 021067
[59] Bravyi S, Browne D, Calpin P, Campbell E, Gosset D and Howard M 2019 Simulation of quantum circuits by low-rank stabilizer decompositions *Quantum* **3** 181
[60] Grover L K 1996 A fast quantum mechanical algorithm for database search (arXiv:quant-ph/9605043 [quant-ph])
[61] Montanez-Barrera J and Michielsen K 2025 Toward a linear-ramp QAOA protocol: evidence of a scaling advantage in solving some combinatorial optimization problems *npj Quantum Inf.* **11** 131
[62] Kremenetski V, Apte A, Hogg T, Hadfield S and Tubman N M 2023 Quantum alternating operator ansatz (QAOA) beyond low depth with gradually changing unitaries (arXiv:2305.04455)
[63] Montanez-Barrera J, Willsch D and Michielsen K 2025 Transfer learning of optimal QAOA parameters in combinatorial optimization *Quantum Inf. Process.* **24** 129
[64] Peruzzo A, McClean J, Shadbolt P, Yung M-H, Zhou X-Q, Love P J, Aspuru-Guzik A and O'brien J L 2014 A variational eigenvalue solver on a photonic quantum processor *Nat. Commun.* **5** 4213
[65] Thompson A C 1996 *Minkowski Geometry* (Cambridge)
[66] Ajtai M 1998 The shortest vector problem in l2 is NP-hard for randomized reductions *Proc. 13th Annual ACM Symp. on Theory of Computing* pp 10–19
[67] Ramaswami V 1949 On the number of positive integers less than x and free of prime divisors greater than x^c *Project Euclid* **55** 1122–7
[68] de Bruijn N G 1951 On the number of positive integers $\leqslant x$ and free of prime factors $> y$ *Proc. K. Ned. Akad. Wet. A* **54** 50–60
[69] Lenstra A K, Lenstra H W and Lovász L 1982 Factoring polynomials with rational coefficients *Math. Ann.* **261** 515
[70] Lucas A 2014 Ising formulations of many NP problems *Front. Phys.* **2** 5
[71] Wang S, Fontana E, Cerezo M, Sharma K, Sone A, Cincio L and Coles P J 2021 Noise-induced barren plateaus in variational quantum algorithms *Nat. Commun.* **12** 6961
[72] Uvarov A and Biamonte J D 2021 On barren plateaus and cost function locality in variational quantum algorithms *J. Phys. A: Math. Theor.* **54** 245301
[73] Anschuetz E R and Kiani B T 2022 Quantum variational algorithms are swamped with traps *Nat. Commun.* **13** 7760
[74] Cerezo M, Sone A, Volkoff T, Cincio L and Coles P J 2021 Cost function dependent barren plateaus in shallow parametrized quantum circuits *Nat. Commun.* **12** 1791
[75] Larocca M, Thanasilp S, Wang S, Sharma K, Biamonte J, Coles P J, Cincio L, McClean J R, Holmes Z and Cerezo M 2024 A review of barren plateaus in variational quantum computing (arXiv:2405.00781)
[76] Gama N, Nguyen P Q and Regev O 2010 Lattice enumeration using extreme pruning *Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques* (Springer) pp 257–78
[77] Dadush D, Peikert C and Vempala S 2011 Enumerative lattice algorithms in any norm via M-ellipsoid coverings *2011 IEEE 52nd Annual Symp. on Foundations of Computer Science* (IEEE) pp 580–9
[78] Dadush D and Kun G 2012 Lattice sparsification and the approximate closest vector problem (arXiv:1212.6781 [cs.DS])
[79] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 Barren plateaus in quantum neural network training landscapes *Nat. Commun.* **9** 4812
[80] Cerezo M, Verdon G, Huang H-Y, Cincio L and Coles P J 2022 Challenges and opportunities in quantum machine learning *Nat. Comput. Sci.* **2** 567
[81] Cerezo M *et al* 2025 Does provable absence of barren plateaus imply classical simulability? *Nat. Commun.* **16** 7907
[82] Joux A and Stern J 1998 Lattice reduction: a toolbox for the cryptanalyst *J. Cryptol.* **11** 161
[83] Nguyen P Q and Stern J 2000 Lattice reduction in cryptology: an update *Int. Algorithmic Number Theory Symp.* (Springer) pp 85–112
[84] Bremner M 2011 *Lattice Basis Reduction* (CRC Press)
[85] Wübben D, Seethaler D, Jalden J and Matz G 2011 Lattice reduction *IEEE Signal Process. Mag.* **28** 70
[86] The FPLLL Development Team 2023 fpylll, a Python wrapper for the fplll lattice reduction library, Version: 0.6.1 (available at: https://github.com/fplll/fpylll)
[87] The FPLLL Development Team yr2023 fplll, a lattice reduction library, Version: 5.4.5 (available at: https://github.com/fplll/fplll)
[88] Harris C R *et al* 2020 Array programming with NumPy *Nature* **585** 357
[89] Cirq Developers 2024 Cirq (https://doi.org/10.5281/zenodo.11398048)
[90] Quantum AI Team and Collaborators 2020 qsim (https://doi.org/10.5281/zenodo.4023103)
[91] Gao F and Han L 2012 Implementing the Nelder-Mead simplex algorithm with adaptive parameters *Comput. Optim. Appl.* **51** 259
[92] Virtanen P *et al* SciPy 1.0 Contributors 2020 SciPy 1.0: fundamental algorithms for scientific computing in Python *Nat. Methods* **17** 261
[93] Priestley B 2025 Code for the paper: a practical scalable approach to the CVP for sieving via QAOA with fixed angles (available at: https://github.com/BenPrie/qaoa-for-cvp)