Variational quantum solutions to the Shortest Vector Problem

Martin R. Albrecht[1], Miloš Prokop[2], Yixin Shen[1], Petros Wallden[2]

[1] Royal Holloway, University of London. email: {martin.albrecht,yixin.shen}@rhul.ac.uk
[2] University of Edinburgh. email: m.prokop@sms.ed.ac.uk, petros.wallden@ed.ac.uk

Abstract. A fundamental computational problem is to find a shortest non-zero vector in Euclidean lattices, a problem known as the Shortest Vector Problem (SVP). This problem is believed to be hard even on quantum computers and thus plays a pivotal role in post-quantum cryptography. In this work we explore how (efficiently) Noisy Intermediate Scale Quantum (NISQ) devices may be used to solve SVP. Specifically, we map the problem to that of finding the ground state of a suitable Hamiltonian. In particular, (i) we establish new bounds for lattice enumeration, this allows us to obtain new bounds (resp. estimates) for the number of qubits required per dimension for any lattices (resp. random q-ary lattices) to solve SVP; (ii) we exclude the zero vector from the optimization space by proposing (a) a different classical optimisation loop or alternatively (b) a new mapping to the Hamiltonian. These improvements allow us to solve SVP in dimension up to 28 in a quantum emulation, significantly more than what was previously achieved, even for special cases. Finally, we extrapolate the size of NISQ devices that is required to be able to solve instances of lattices that are hard even for the best classical algorithms and find that with $\approx 10^3$ noisy qubits such instances can be tackled.

1 Introduction

Cryptography studies the limits of computing: what can and cannot efficiently be computed. In 1976 Diffie and Hellman significantly expanded the realm of the possible by inventing public key cryptography [DH76] which allows two parties to agree on a shared secret over a public channel in the presence of a wiretapping adversary.[3] Since its invention public-key cryptography has seen widespread adoption and is now a crucial building block for securing, say, the Internet. However, this advance was not unconditional but relies on the presumed hardness of some computational problem. Virtually all currently deployed public-key encryption schemes rely on the difficulty of one of two computational problems: the discrete logarithm problem and the problem of factoring large integers.

Everything changed in 1994 when Peter Shor's seminal work [Sho97] showed that quantum computers could effectively solve those two central problems. While the timeline for when sufficiently big quantum computers may be available is uncertain, the proposed such timelines and the threat of "collect-now-decrypt-later"-style attacks provoked global efforts to develop "post-quantum cryptography", cryptographic schemes that run on classical computers but resist attacks with quantum computers. The centre of these international efforts is the Post-Quantum Standardisation Process (NIST PQC) by the US National Institute of Standards and Technology (NIST) [NIS]. To date, there are several candidates for post-quantum cryptography, mainly lattice-based, code-based, hash-based, multivariate cryptography and supersingular elliptic curve isogeny cryptography. Lattice-based cryptography seems to be a prime contender for large scale adaption: among the Round 3 finalists of the NIST PQC, five out of seven finalists are based on lattices.

While, of course, all post-quantum candidates are conjectured to be hard also on a quantum computer, a pressing question for adoption is "how hard?". That is, in order to pick parameters that are both efficient in practice but quantifiably resist attacks, the research community studies the quantum resources required to solve the underlying hard problems.

On the other hand, it is expected that the transition to quantum computers will start with a phase referred to as *Noisy Intermediate Scale Quantum* (NISQ), featuring devices that consist of at most one thousand erroneous qubits. This low number of qubits makes use of any known error-correction technique infeasible, something that also puts stringent restrictions on the depth a quantum computation can have before the noise becomes dominant leading to essentially random outcomes. To overcome this limitation of these devices, hybrid classical-quantum algorithms are designed specifically for these devices while most of the famous quantum algorithms like Grover's search, Quantum Fourier Transform

[3] If the public channel is authenticated, the adversary may even actively interact with both parties.

or Shor's algorithms are impracticable. A natural question is thus "how hard are lattice problems on NISQ devices?". This question sheds light on the performance of these devices on a natural and central computational problem.

Contributions. After some preliminaries (Section 2), we determine a suitable mapping of the central hard problem on lattices, the *Shortest Vector Problem* (SVP) into the ground state of a Hamiltonian operator, the form required by the classical-quantum optimisation algorithms of our interest. In Section 3, we introduce Ising spin Hamiltonian operators and explain how to map SVP to an optimisation problem in this framework. We discuss the challenges that arise with this formulation and give a solution to the "zero vector problem" for VQE.[4] Our problem formulation requires to know a priori bounds for each coordinate of a shortest vector in a given basis $\boldsymbol{B}$ of the lattice. In Section 4 we thus analyse the cost of solving SVP in a non-adaptive exhaustive search which allows us to quantify the search space. In particular, we show that the size of the search space depends on the norms of the vectors forming the "dual basis" $\widehat{\boldsymbol{B}} := (\boldsymbol{B}\boldsymbol{B}^T)^{-1}\boldsymbol{B}$. This allows us to obtain a NISQ quantum algorithm to compute an HKZ reduced basis, one of the strongest notion of reduction, using $\frac{3}{2}n\log_2 n + O(n)$ qubits and thus solving SVP in passing. This cost of non-adaptive enumeration was previously known only for a special class of lattices. We also perform extensive classical numerical experiments to study the average case behaviour of lattice reduction in the context of our NISQ quantum enumeration algorithm. In Section 5 we then show that our bounds allow us to run quantum emulation experiments that using up to 28 qubits are able to solve SVP in dimension 28 which is considerably more than prior literature. Extrapolating our experimental data we find that between $1,000$ and $1,600$ qubits suffice to encode SVP for a dimension 180 lattice, the current record dimension for "Darmstadt SVP Challenge" [DSvW21]. For the avoidance of doubt, our results do not violate any previous claims on the hardness of lattice problems on quantum computers because in general we may hope for a running time at best $2^{\lambda/2+o(\lambda)}$ for instances encoded in $\lambda = \frac{3}{2}n\log_2 n + O(n)$ qubits.[5]

2 Preliminaries

Lattices A (Euclidean) *lattice* $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^d$, or equivalently the set $\mathcal{L}(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n) = \{\sum_{i=1}^n x_i\boldsymbol{b}_i \ : \ x_i \in \mathbb{Z}\}$ of all integer combinations of n linearly independent vectors $\boldsymbol{b}_1,\ldots,\boldsymbol{b}_n \in \mathbb{R}^d$. Such $\boldsymbol{b}_i$ form a *basis* of $\mathcal{L}$. We say that a matrix $\boldsymbol{B}$ forms a basis of $\mathcal{L}$ if its rows form a basis of $\mathcal{L}$. All the bases have the same number n of elements, called the *dimension* or *rank* of $\mathcal{L}$. The dual $\widehat{\mathcal{L}}$ of a lattice $\mathcal{L}$ is the set of all vectors $\boldsymbol{x} \in \mathrm{span}(\mathcal{L})$ such that $\langle \boldsymbol{x}, \boldsymbol{y}\rangle$ is an integer for all $\boldsymbol{y} \in \mathcal{L}$. If $\boldsymbol{B}$ is a basis of $\mathcal{L}$ then $\widehat{\boldsymbol{B}} := (\boldsymbol{B}\boldsymbol{B}^T)^{-1}\boldsymbol{B}$ is a basis of $\widehat{\mathcal{L}}$. We call $\widehat{\boldsymbol{B}}$ the "dual basis" of $\boldsymbol{B}$.

One of the fundamental algorithmic problems related to lattices is to find a shortest non-zero element of an arbitrary lattice (with respect to its Euclidean norm), given an arbitrary basis of this lattice. This problem is referred to as the shortest vector problem (SVP) and the length of such a vector denoted by $\lambda_1(\mathcal{L})$. It is a central premise of lattice-based cryptography that solving SVP (and its decision variant GapSVP) within a polynomial factor takes super-polynomial time also on a quantum computer [Reg05]. It is well-known that the hardness of the SVP is related to the "quality" of the basis which informally quantifies the length of the vectors in the basis and the angles between. Intuitively, a basis with short and relatively orthogonal vectors is of higher quality. Therefore, a fundamental problem in lattice-based cryptography is to increase the quality of a given basis, a process known as lattice reduction. The celebrated LLL algorithm [LLL82] was the first polynomial-time algorithm that computes a reduced basis of guaranteed quality, namely the first vector is at most exponentially longer than the shortest vector of the lattice. The BKZ algorithm [Sch87] is a generalisation of LLL to obtain more strongly reduced basis at the expense of a higher running time. More precisely, the BKZ algorithm requires one to choose a so-called block size β: the larger the β, the stronger the reduction but the higher the running time (which is at least exponential in β). BKZ internally uses an algorithm to solve (near) exact SVP in lower-dimensional lattices. Therefore, understanding the complexity of SVP is critical to understanding

[4] In Appendix A we outline a less efficient solution to impose the constraint at the Hamiltonian level adding new variables. That approach is applicable to QAOA, quantum annealing and adiabatic quantum computing thus extending our method.

[5] Classically, SVP can be solved in time $n^{1/(2e)n+o(n)}$ and $\mathsf{poly}(d)$ memory or $2^{0.292\,n+o(n)}$ time and memory.

the complexity of BKZ and lattice reduction. This, in turns, is critical to choosing security parameters of cryptographic primitives [ACD+18].

Finally, one of the strongest notion of reduction is that of HKZ reduced basis [Kan83]. The first vector of such a basis is always a shortest vector of the lattice. Furthermore, HKZ basis naturally lend themselves to be computed recursively and enjoy many good properties, especially in conjunction with enumeration [HS07]. Enumeration algorithms list all of the lattice points within a ball of prescribed radius r. One of the most important aspect of enumeration is to correctly chose the enumeration radius r so that it is larger than $\lambda_1(\mathcal{L})$, but not too large since the running time increases rapidly with r. For random lattices, the so-called Gaussian Heuristic gives a good estimate of $\lambda_1(\mathcal{L})$ as $\mathrm{gh}(\mathcal{L}) := \sqrt{\frac{n}{2\pi e}}\det(L)^{1/n}$.

The fastest known (heuristic) quantum algorithm [CL21] for solving SVP is a "sieving algorithm" runs in time $2^{0.257\,n+o(n)}$, uses QRAM [GLM08] of maximum size $2^{0.0767\,n+o(n)}$, a quantum memory of size $2^{0.0495\,n+o(n)}$ and a classical memory of size $2^{0.2075\,n+o(n)}$. The second main class of quantum algorithms for solving SVP are "lattice-point enumeration" algorithms which combine exhaustive search with projections [ANS18]. These algorithms run in time $n^{n/(4e)+o(n)}$ and $\mathsf{poly}(d)$ memory. In many cryptography applications $n^{n/16+o(n)}$ seems plausible [ABF+20]. On classical computers, the current record computation solved the "Darmstadt SVP Challenge", which asks to solve a slightly relaxed version of SVP, in dimension 180 [DSvW21] using sieving on GPUs. Both classes of algorithms rely on Grover's algorithm or random walks which require long running computations on fault-tolerant quantum computers and thus are not suitable on quantum devices of the next decade, the NISQ era.

Variational Quantum Algorithms (VQA)s [CAB+21] are hybrid classical-quantum algorithms that can solve a variety of problems, including optimisation problems. VQAs are believed to be one of the most promising approaches to use and possibly offer quantum advantage on near-term quantum devices – also known as Noisy Intermediate-Scale Quantum (NISQ) devices. The quantum devices that currently exist and those that will become available in the coming 2-5 years are expected to have at most 1000 qubits (intermediate-scale) and have imperfections (noisy). Since the number of qubits is limited, to run computations of interest one cannot "afford" to perform quantum error-correction since this would require an overhead in the order of $\approx 10^3$ physical qubits for every useful logical qubit. VQAs mitigate the effects of noise by a different approach. A computationally expensive subroutine is solved by the quantum device that amounts in estimating the "energy" of a quantum state that arises as the output of a parameterized quantum circuit of short depth (avoiding the excessive accumulation of errors). The preparation and measurement is run multiple times and the output is fed to a classical optimisation part that essentially off-loads part of the computation to a classical device. VQAs appear to offer advantages over classical algorithms in various areas of quantum chemistry [KMW+17,WZdS+20] and is a promising approach for many other areas including combinatorial optimization [Luc14], quantum machine learning [CLKAG21] and quantum metrology [KEJ+20].

Since SVP can be formulated as an optimisation problem, we focus on two most widely used VQAs for combinatorial optimisation, namely Variational Quantum Eigensolver (VQE) [PMS+14,MRBAG16] and Quantum Approximate Optimisation Algorithm (QAOA) [FGG14]. The first step for both of them, as well as for quantum annealing [HKL+20], is to encode the problem (SVP here) to the ground state (smallest eigenvalue) of a Hamiltonian operator $\mathcal{H}$. For QAOA and quantum annealing, the Hamiltonian needs to be an Ising spin Hamiltonian (i.e. involving only Z spins and up to quadratic interaction terms) and for quantum annealing extra limitations due to the connectivity of the spins apply. For VQE the Hamiltonian can take more general form (including higher order terms and/or X spins). The second step, once the Hamiltonian is chosen, is to select an *ansatz state* $|\psi(\theta)\rangle$ – a family of quantum states, parametrized by θ, that are the output of a simple parameterized circuit. Ideally, we would like to ensure that elements of the family considered are close to the ground state of the problem's Hamiltonian. The two classes of anstze that exist are the *hardware efficient* ones that are essentially chosen for the ease that can be implemented at a given quantum hardware, and the *problem specific* that are anstze that use information about the problem for example using the problem's Hamiltonian. The former are less prone to errors and can be used with any Hamiltonian but have no guarantee to be "dense" around the true ground state, while the latter are more sensitive to noise and can be used with specific classes of Hamiltonians but are designed to have states close to the true ground state. The third step is to prepare a state from the ansatz and measure it where this step is repeated multiple times. From these

repetitions an estimate of the expectation $\langle\psi(\theta)|\mathcal{H}|\psi(\theta)\rangle$ is calculated and passed as the cost value for the choice of parameters to a classical optimizer.

The optimizer then calculates new parameter θ and the procedure repeats until some stopping criterion is reached and an estimate for the ground state is produced. For our VQE runs we used hardware efficient anstze. QAOA is by definition a problem specific ansatz since the family is constructed as a discretized version of Quantum Adiabatic Computation. This puts a constraint on the form of the Hamiltonian (which makes it harder to solve SVP with fewer qubits see Section 3.2), but has the theoretical guarantee that for sufficiently deep circuits the solution should be found. Specifically, for Hamiltonians that involve only spin Z terms, to compute the energy/cost $C(\theta)$ of a quantum state $|\psi(\theta)\rangle$ we prepare the state and measure in the computational basis N times. Each run gives outcomes (bit-strings) x_i, and for every outcome we compute the related cost $m_i(x_i)$. Our estimate of the cost of the state is $C(\theta) = \langle\psi(\theta)|\mathcal{H}|\psi(\theta)\rangle \approx \frac{1}{N}\sum_{i=1}^{N} m_i$ and is used by the classical optimizer. For classical combinatorial optimization problems one can find that other methods are performing better. The Conditional Vale at Risk (CVaR) [BNR+20] and the Ascending-CVaR [KW21] are two methods that give better results and in this work we will use the former. In [BNR+20] instead of computing the cost taking the average of the values m_i, they considered ordering the values from the smallest to the larger and counting the α-tail of the distribution. Specifically, α is to be chosen heuristically and the cost is calculated as an average of $\lceil \alpha N \rceil$ lowest measurements outcomes. Suppose $\{\tilde{m}_i\}_{i=1,\ldots,N}$ is a sorted set of $\{m_i\}_{i=1,\ldots,N}$ in non-decreasing order. Then the cost is calculated as: $C_{CVaR_\alpha}(\theta) = \frac{1}{\lceil \alpha N \rceil}\sum_{i=1}^{\lceil \alpha N \rceil} \tilde{m}_i$. We note that finding a ground state of a Hamiltonian is QMA-complete in general, but for specific Hamiltonians the ground *can* be found efficiently – adiabatic quantum computing is a universal model, that finds efficiently the ground state of those Hamiltonians that correspond to BQP problems. Indeed, in our case, since SVP is believed to be outside BQP we do not expect to find the solution efficiently. On the other hand, obtaining a polynomial speed-up is valuable for the cryptanalysis – after all Grover's algorithm also provides such a moderate speed-up. In particular, we do not expect to do better than Grover – query complexity bounds indicate that for our problem we can get at most a small constant improvement to Grover. However, due to the fact that our approach is heuristic, we may still be able to get considerable (larger) speed-up for certain (but not the hardest) instances.

2.1 Related Work

There have been several works focusing on translating SVP into Hamiltonian $\mathcal{H}$ where the ground state corresponds to the shortest lattice vector. Note that we can trivially achieve encodings where the corresponding eigenvalues of $\mathcal{H}$ define the order of lattice vector lengths. The resulting variational quantum formulation is thus capable of solving the approximate SVP, a relaxation of SVP, on which the security of most lattice-based protocols is relied upon. In [JGLM20] the energy gaps between the first three excited states of the problem Hamiltonian are analysed when solving low dimensional SVP via adiabatic computation. The ground state is by their construction a zero state and hence the first excited state is sought. The results suggest the existence of "*Goldilocks*" zones. In such zones the adiabatic evolutions are slow enough that there is a high probability of sampling any of the first three excited states without any strong dominance of any of them. As a consequence, in this case it is possible to obtain the shortest non-zero vector with a small number of measurement samples. This motivates the use of QAOA to find the ground state as it mimics the adiabatic computation. However, the experiments were performed only for lattices up to 4 dimensions. Moreover, as SVP is an integer optimization problem, bounds on the ranges of the (integer) variables need to be defined prior to mapping the problem into a binary optimization problem. The authors make an experimental guess that each of them grows linearly with lattice dimension resulting in guess of $O(n \log n)$ qubit requirement. The qubit requirement was later proved in [JCLM21] for special lattices with what they called "an optimal Hermite Normal Form (HNF)". The density of such lattices is around 44% for lattices generated with entries selected "uniformly at random" [Maz10, Section 5.1]. For such lattices, they show that $\frac{3}{2}n\log_2 n + n + \log\det(\mathcal{L})$ qubits are enough to guarantee that the shortest vector is an eigenvector of the hamiltonian. To confirm the approach, experiments for up to 7-dimensional instances of SVP were performed on D-Wave quantum annealer making use of up to 56 logical qubits. The proof of their bound crucially relies on the special shape of the HNF namely that it has at most one nontrivial column. However, q-ary lattices (which are

ubiquitous in cryptography, see Section 5.1) almost never have such a special Hermite Normal Form since[6] they have k nontrivial colums, where k is the rank of the underlying linear code and typically linear in the dimension (such as $n/2$).

3 Direct approaches to SVP with Variational Quantum Algorithms

Recall that the starting point for VQE and QAOA (but also for quantum annealing) is to encode the problem to the ground state of a Hamiltonian $\mathcal{H}$. As in other combinatorial optimization problems, one first needs to turn the problem to a quadratic unconstrained binary (QUBO) form and then there is a standard method to obtain an Ising Hamiltonian (i.e. a Hamiltonian that involves only Z operators and the interaction terms involve just pairs of qubits). Note that to run VQE one can have more general Hamiltonians and as we will see later, in some cases it is simple to include constraints (or even higher order terms). Coming back to QUBO formulatons the general form of the cost is $C(s_1 s_2 ... s_n) = c + \sum_i c_{ii} s_i + \sum_{i \neq j} c_{ij} s_i s_j$, for string binary variables $s_1 s_2 \cdots s_n$, for some coefficients $c, \{c_{ij}\}_{1 \leq i,j \leq n}$. Since Pauli-Z operators have ± 1 eigenvalues one maps each binary variable s_i to $\frac{I_i - Z_i}{2}$ leading to the Ising Hamiltonian, $H = \sum_i c_{ii} \frac{I_i - Z_i}{2} + \sum_{i \neq j} c_{ij} \frac{I_i - Z_i}{2} \otimes \frac{I_j - Z_j}{2}$ where Z_i and I_i are the Pauli-Z and identity operators respectively, acting on i-th qubit.

3.1 Mapping the Shortest Vector Problem into QUBO formulation

A first approach (also starting point of [JGLM20,JCLM21]), is to consider as variables the coefficients of the basis vectors. Each choice of the coefficients determines a vector in the lattice, and its length can be easily calculated using the basis matrix. There are two major challenges with this approach. The first one is that the coefficients are integers while we want to use a finite number of binary variables. Recall that each binary variable will result to a qubit so we can only use a finite number while making this number as small as possible is crucial to run it in a NISQ device. We therefore need to truncate (limit) the possible coefficients, but in a way that the shortest vector is included in our possible solutions. The second challenge is that if the variables are the coefficients then a possible vector is the vector that all coefficients are zero, a vector that clearly is shorter than the shortest (non-zero) vector we are searching. We would therefore need to impose the constraint $x \neq 0$ at some level.

Going back to the formulation, given an n-dimensional full-rank row-major lattice basis matrix B, we define lattice $\mathcal{L}(B) = \{xB | x \text{ row vector}, x \in \mathbb{Z}^n\}$. The shortest vector problem (SVP) finds the solution $\lambda_1 = \min_{y \in \mathcal{L}(B) \setminus \{0\}} |y|$. Let x be a row vector of coefficients and $G = BB^T$ a gram matrix of the basis. Then $y = xB \implies |y|^2 = xBB^T x^T = xGx^T$ which allows us to reformulate it as a quadratic constrained integer optimization problem:

$$\lambda_1^2 = \min_{y \in \mathcal{L}(B) \setminus \{0\}} |y|^2 = \min_{x \in \mathbb{Z}^n \setminus \{0\}} \sum_{i=1}^{n} x_i G_{ii} + 2 \sum_{1 \leq i < j \leq n} x_i x_j G_{ij}. \tag{1}$$

As a consequence of rank-nullity theorem, we have $\lambda_1 \neq 0$ as required. In order to convert (1) into a binary optimization problem, we need bounds $|x_i| \leq a_i$ for all $i = 1, ..., n$. This has been the core problem of all approaches to map the SVP into a Hamiltonian and we provide tighter and more general bounds in Section 4.

We now define new binary variables $\{\tilde{x}_{ij}\}_{0 \leq j \leq \lfloor \log 2a \rfloor}$ using the initial integer variables x_i and the bound a_i

$$|x_i| \leq a \implies x_i = -a + \sum_{j=0}^{\lfloor \log 2a \rfloor - 1} 2^j \tilde{x}_{ij} + (2a + 1 - 2^{\lfloor \log 2a \rfloor}) \tilde{x}_{i, \lfloor \log 2a \rfloor} \tag{2}$$

where the last term ensures we do not enlarge the search space. By substituting into (1) and ignoring the $x \neq 0$ constraint, the resulting QUBO becomes (3) where c, c_{ij} and $d_{ij,kl}$ are calculated constants

[6] Indeed, the matrix of a q-ary lattice is $\begin{bmatrix} \boldsymbol{I}_{n-k} & \boldsymbol{X} \\ 0 & q\boldsymbol{I}_k \end{bmatrix}$, where $\boldsymbol{X} \in \mathbb{Z}_q^{(n-k) \times k}$ has rank k, is already in HNF.

resulting from the substitution.

$$\min_{\substack{x_{1,0},\ldots,x_{1,\lfloor \log 2a_1 \rfloor} \\ \ldots \\ x_{n,0},\ldots,x_{n,\lfloor \log 2a_n \rfloor}}} c + \sum_{\tilde{x}_{ij}} c_{ij}\tilde{x}_{ij} + \sum_{\tilde{x}_{ij},\tilde{x}_{kl}} d_{ij,kl}\tilde{x}_{ij}\tilde{x}_{kl} \tag{3}$$

3.2 Encoding the $x \neq 0$ constraint

In deriving Equation 3 we ignored the $x \neq 0$ condition in the definition of SVP. The true minimum is zero and is obtained by setting all the integer variables x_1,...,x_n to zero. Instead we want to obtain the second smallest value. In other words, we are seeking the first excited state of the corresponding Hamiltonian, with the extra information that the "true" ground state (zero vector) is known. There are three ways to address this. First is to ignore the constraint, run the optimization to find the ansatz state with greatest overlap with the zero vector, and hope that it has an (ideally large) overlap with the first excited state [JGLM20]. Even if this approach succeeds in finding SVP for small dimensions it is unlikely to work well at large scales. The second solution is to exclude the zero vector by imposing it in a form of a constraint that "penalizes" the zero vector. To map the problem back to a QUBO, however, requires introducing extra variables ($n-2$ in our case), making this approach less practical. Since this approach can work both for QAOA and quantum annealing where the last and more practical approach does not work, we give the details of this mapping in Appendix A.

The third approach, that we will analyse here and used in our numerical experiments, can be used in VQE. One directly targets the first excited state by modifying the classical loop of the VQE i.e. modifying the way that the cost is evaluated. In the general form, assuming that $|\psi_0\rangle$ is the ground state, instead of using the expression $C(\psi) = \langle\psi|H|\psi\rangle$ for the cost, we use $C(\psi) = \langle\psi|H|\psi\rangle \frac{1}{1-|\langle\psi|\psi_0\rangle|^2}$, where the multiplicative factor is the inverse of the probability of not being in the ground state. This gives infinite cost to the ground state, and in general penalizes states with greater overlap with the ground state. It is easy to see that this cost is minimized at the first excited state of the original Hamiltonian. Note that other approaches to find the first excited state exist (see for example [HWB19]).

In our case the ground state is the zero vector. Let $\tilde{N}$ be the cardinality of m_i's (measured bit-strings) that are non-zero. Then $\frac{N}{\tilde{N}}$ is the numerical estimate of the inverse of the probability not being the zero vector. Therefore we will be using the modified cost $C'(\theta) = \left(\frac{N}{\tilde{N}}\right)\frac{1}{N}\sum_{i=1}^{N} m_i$. Equivalently, this means that to compute the cost of a state, we disregard the zero-vector outcomes taking the average value over all the other outcomes. In the measurements of the final ansatz state, at the end of the optimization, we output the measurement sample with the lowest non-zero energy. The advantage of this approach is that the quantum part (states and measurements) are identical to the unconstrained one, and the only difference is in the way that the measurement outcomes are used to assign a cost to different states in the classical optimization part of the VQE.

4 Bounds on lattice enumeration and application to variational quantum algorithms for solving SVP

In the previous section we gave a map of SVP to a QUBO formulation that is suitable for VQE. This map, however, relies on bounds on each $|x_i|$. In this section, we obtain worst case bounds on the $|x_i|$ based on the orthogonality defect of the dual basis. We then give two applications of this result. First, we show how to obtain a recursive algorithm to compute a HKZ reduced basis and thus solving the SVP in passing. This method gives us the best asymptotical bound on the number of qubits required. Second, we estimate the number of qubits required to directly solve the SVP by reducing the dual basis with recursive or classical preprocessing and then applying the NISQ enumeration.

4.1 General bounds on lattice enumeration

As we have seen in Section 3.1, we need to choose a bound on each of the $|x_i|$ to obtain a finite optimisation problem. This bound will then determine the number of qubits required. In order to be sure to find a solution, we first derive a general bound on all the x_i that correspond to lattice points in a ball of radius A. We then use the Gaussian heuristic to choose the radius A so as to ensure that this ball contains at least one nonzero vector for most lattices and therefore a shortest vector.

Lemma 1. *Let $x_1, \ldots, x_n$ be such that $\|x_1 \cdot \boldsymbol{b}_1 + \cdots + x_n \cdot \boldsymbol{b}_n\| \leqslant A$, then for all $i = 1, \ldots, n$ we have $|x_i| \leqslant A\|\hat{\boldsymbol{b}}_i\|$ where $\hat{\boldsymbol{b}}_1, \ldots, \hat{\boldsymbol{b}}_n$ are the rows of $\widehat{\boldsymbol{B}}$ and $\boldsymbol{B}$ is the matrix whose rows are $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$.*

Proof. Let $\delta_{i,j}$ denote the Kronecker delta. Observe that $\left\langle \boldsymbol{b}_j, \hat{\boldsymbol{b}}_i \right\rangle = \delta_{i,j}$ for all i, j. Indeed, $\left\langle \boldsymbol{b}_j, \hat{\boldsymbol{b}}_i \right\rangle = (\boldsymbol{B} \cdot \widehat{\boldsymbol{B}}^T)_{i,j} = (\boldsymbol{I}_n)_{i,j} = \delta_{i,j}$. Now let $\boldsymbol{v} = x_1 \cdot \boldsymbol{b}_1 + \cdots + x_n \cdot \boldsymbol{b}_n$ for some $x_1, \ldots, x_n \in \mathbb{Z}$ be such that $\|\boldsymbol{v}\| \leqslant A$. Then for any i, we have that $\left|\left\langle \boldsymbol{v}, \hat{\boldsymbol{b}}_i \right\rangle\right| = |x_i|$. But on ther other hand, $\left|\left\langle \boldsymbol{v}, \hat{\boldsymbol{b}}_i \right\rangle\right| \leqslant \|\boldsymbol{v}\| \cdot \|\hat{\boldsymbol{b}}_i\|$ which proves the result. □

If we now make use of the Gaussian heuristic to choose the radius A and take $A \coloneqq \mathrm{gh}(L) = \sqrt{n/2\pi e} \cdot \mathrm{vol}(L)^{1/n}$. The total number of qubits that we need will be

$$N = \sum_{i=1}^{n} (\lfloor \log_2(2m_i) \rfloor + 1) \leqslant 2n + \log_2 \prod_{i=1}^{n} m_i \tag{4}$$

where m_i is the bound on the $|x_i|$. Using the bound of Lemma 1 we obtain

$$\prod_{i=1}^{n} m_i \leqslant \left(\sqrt{\frac{n}{2\pi e}}\right)^n \cdot \mathrm{vol}(L) \cdot \prod_{i=1}^{n} \left\|\hat{\boldsymbol{b}}_i\right\| = \left(\sqrt{\frac{n}{2\pi e}}\right)^n \cdot \frac{\prod_{i=1}^{n} \left\|\hat{\boldsymbol{b}}_i\right\|}{\mathrm{vol}(L^*)} = \left(\sqrt{\frac{n}{2\pi e}}\right)^n \delta(L^*)$$

where L^* is the dual of $L = L(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ and $\delta(\cdot)$ denotes the orthogonality defect. This suggest that the critical factor for the enumeration is the reduction of the dual of the lattice and not the lattice itself.

Corollary 1. *The number of qubits required for the enumeration on the basis $\boldsymbol{B}$, assuming the Gaussian heuristic is bounded by $2n + \log_2\left(\left(\frac{C^2 n}{2\pi e}\right)^{n/2} \delta(\widehat{\boldsymbol{B}})\right)$ where $\delta(\cdot)$ denotes the orthogonality defect and C the multiplicative factor used with the Gaussian heuristic[7].*

4.2 Bound on the number of qubits for producing a HKZ basis

In this section, we show that our NISQ enumeration procedure can be used to HKZ reduce a basis in a recursive way using only $\frac{3}{2} n \log_2(n) + O(n)$ qubits, thus solving SVP in passing. This is asymptotically the same as the algorithm in [JCLM21]. However, our algorithm works for any lattice, whereas the algorithm in [JCLM21] only works for lattices that have what they called "an optimal Hermite Normal Form". See Section 2.1 for more details.

Our algorithm is a variant of the classical algorithm for producing a HKZ reduced basis from Kannan [Kan83]. The main difference is that Kannan's algorithm exclusively works on the (primal) basis whereas our algorithm has to reduce both the primal and the dual basis to control the number of qubits required in the enumeration steps. This difference requires us to work slightly differently. In particular, Kannan's algorithm works by recursively producing a "quasi-HKZ" reduced basis, and then applying enumeration on this basis to obtain a full HKZ basis. A quasi-HKZ reduced basis is essentially a LLL basis on which one applies an HKZ reduction to the projection of the *last* $n-1$ vectors (orthogonally to the first). In contrast, our algorithm first produces what we call a "pseudo-HKZ" basis and then runs the enumeration on its dual. This pseudo-HKZ basis is an LLL basis on which one applies an HKZ reduction to the *first* $n-1$ vectors. Although the two notions are close, they are not exactly the same.

We will require the following well-known result about HKZ basis.

Proposition 1 ([WC19]). *The orthogonality defect of a HKZ basis is bounded by $\gamma_n^{n/2} \cdot \prod_{i=1}^{n} \cdot \frac{\sqrt{i+3}}{2}$ where γ_n is Hermites constant in dimension n, and $\gamma_n < \frac{1}{8}n + \frac{6}{5}$.*

Corollary 2. *The orthogonality defect of a HKZ basis is $\leqslant 2^{n \log_2 n - \left(\frac{5}{2} + \frac{1}{2}\log_2 e\right) n + O(\log n)}$.*

Proof. See Appendix D.

[7] The enumeration is done with a radius of $C \cdot \mathrm{gh}(L)$.

input : A basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$
output : An HKZ basis of the same lattice

1 $L := L(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$
2 $(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n) :=$ dual basis of $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$
3 $(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n) :=$ LLL-reduce $(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n)$
4 $(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{n-1}) :=$ HKZ-reduce $(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{n-1})$ `/* the basis is` $(n-1)$`-dimensional */`
5 $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) :=$ dual basis of $(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n)$
6 Call the NISQ enumeration procedure proposed in Section 3 on $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ using the bounds established in Lemma 1 and keep a shortest vector $\boldsymbol{v}$
7 $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) :=$ LLL-reduce $(\boldsymbol{v}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ `/* extract a basis from` $n+1$ `vectors */`
8 Compute the orthogonal projections $(\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_n)$ of $(\boldsymbol{b}_2, \ldots, \boldsymbol{b}_n)$ on $\boldsymbol{b}_1^{\perp}$
9 $(\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_n) :=$ HKZ-reduce $(\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_n)$ `/* the basis is` $(n-1)$`-dimensional */`
10 For each i, set $\boldsymbol{b}_i := \boldsymbol{b}'_i + \alpha_i \boldsymbol{b}_1$ where $\alpha_i \in \left(-\frac{1}{2}, \frac{1}{2}\right]$ is such that $\boldsymbol{b}_i$ belongs to L
11 **return** $(\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n)$

Algorithm 1: HKZ-reduction algorithm using quantum enumeration

Theorem 1. *Assuming the Gaussian heuristic, Algorithm 1 produces an HKZ basis with* $\frac{3}{2}n\log_2(n) - 2.26n + O(\log n)$ *qubits.*

Proof. We first show that the algorithm is correct by induction on n. Let $L = L(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ at the beginning of the algorithm. First note that at the beginning of line 6, the basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ is still a basis of the original lattice L since all the previous operations preserve that property. Assuming the Gaussian heuristic is true (or true with a multiplicative factor C), the enumeration algorithm will indeed find a shortest nonzero vector $\boldsymbol{v}$ at line 6. The reduction at line 7 will allow to extract a basis of L among the $n+1$ vectors $\boldsymbol{v}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$. Since $\boldsymbol{v}$ is a shortest vector, the LLL reduction will ensure that $\boldsymbol{b}_1$ is a shortest vector of L at the beginning of line 8. Let $L' = L(\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_n)$ be the projected lattice obtained after line 8. The reduction at line 9 does not change the lattice spanned by the $\boldsymbol{b}'_i$ which is now a HKZ basis. The lifting operation at line 10 ensures that $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ is a basis of L at the end of the algorithm. Since $\boldsymbol{b}_1$ is a shortest vector of L and $(\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_n)$ is HKZ and a basis of the projected lattice, it follows that the returned basis is HKZ.

We now analyze the qubits requirement of the algorithm. Let $(\bar{\boldsymbol{d}}_1, \ldots, \bar{\boldsymbol{d}}_n)$ be the basis obtained after line 3 and $(\bar{\boldsymbol{d}}_1^*, \ldots, \bar{\boldsymbol{d}}_n^*)$ be its Gram-Schmidt orthogonalization. By the properties of the LLL-reduction, we have $\left\|\bar{\boldsymbol{d}}_n\right\|^2 \leqslant 2^{n-1} \cdot \left\|\bar{\boldsymbol{d}}_n^*\right\|^2$. Let $(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n)$ be the basis obtained after line 4 and $(\boldsymbol{d}_1^*, \ldots, \boldsymbol{d}_n^*)$ be its Gram-Schmidt orthogonalization. Clearly $\boldsymbol{d}_n = \bar{\boldsymbol{d}}_n$ since this vector was not touched by the HKZ reduction. It follows that $\boldsymbol{d}_n^* = \bar{\boldsymbol{d}}_n^*$ because $\boldsymbol{d}_n^*$ (resp. $\bar{\boldsymbol{d}}_n^*$) is the projection of $\boldsymbol{d}_n = \bar{\boldsymbol{d}}_n$ on the orthogonal of $\mathrm{span}(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n) = \mathrm{span}(\bar{\boldsymbol{d}}_1, \ldots, \bar{\boldsymbol{d}}_n)$ which are equal because the HKZ reduction does not change the span. As a result, we have $\|\boldsymbol{d}_n\|^2 \leqslant 2^{n-1} \cdot \|\boldsymbol{d}_n^*\|^2$ after line 4 and therefore

$$\delta(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n) = \delta(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{n-1}) \cdot \frac{\|\boldsymbol{d}_n\|}{\|\boldsymbol{d}_n^*\|} = 2^{n\log_2 n - \left(2 + \frac{1}{2}\log_2 e\right)n + O(\log n)}$$

by Corollary 2. Taking the dual of this basis at line 5 means at line 6, we are in a position to run the enumeration algorithm on a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ whose dual $(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n)$ has a small orthogonality defect. By Corollary 1, the number of qubits required for this enumeration step is bounded by

$$2n + \log_2 \left[\left(\frac{C^2 n}{2\pi e} \right)^{n/2} \delta(\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n) \right] \leqslant \tfrac{3}{2} n \log_2 n - \left(\tfrac{1}{2}\log_2 \pi + \log_2 e - \log_2 C \right) n + O(\log n)$$

Denote by $Q(n)$ the number of qubits necessary to run the algorithm in dimension n. Since we run the enumerations sequentially, $Q(n) = \max(Q(n-1), n\log_2(n) + O(n))$ and therefore $Q(n) = \frac{3}{2}n\log_2(n) - 2.26n + O(\log n)$ for $C = 1$. □

4.3 Solving the SVP directly in the NISQ era

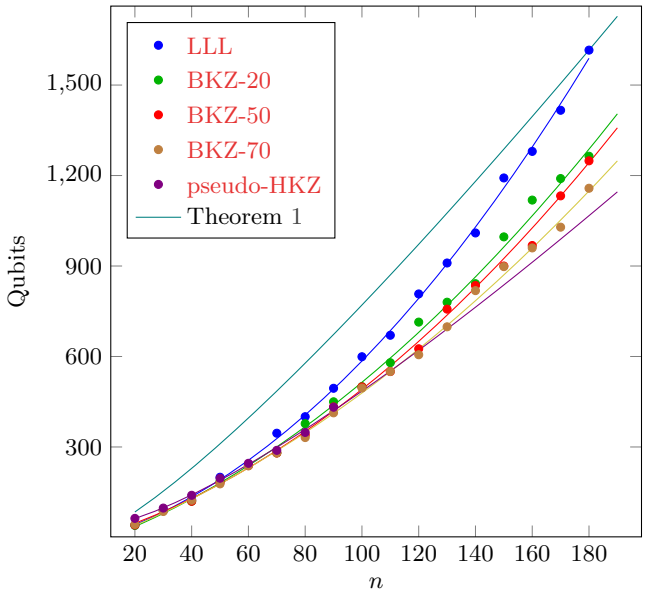
As we have seen in the previous sections, the bound on the x_i and thus the running time of the enumeration fundamentally depend on the quality of the dual basis. The algorithm of the previous

section takes advantage of this fact by recursively reducing the primal and dual basis and always running the enumeration on a dual quasi-HKZ reduced basis, one of the strongest notion of basis reduction. Unfortunately, the bound on the number of qubits that we obtained relies on worst case bounds on the orthogonality defect of such basis. In particular, the linear term in the bound of Theorem 1 is quite pessimistic for most bases.

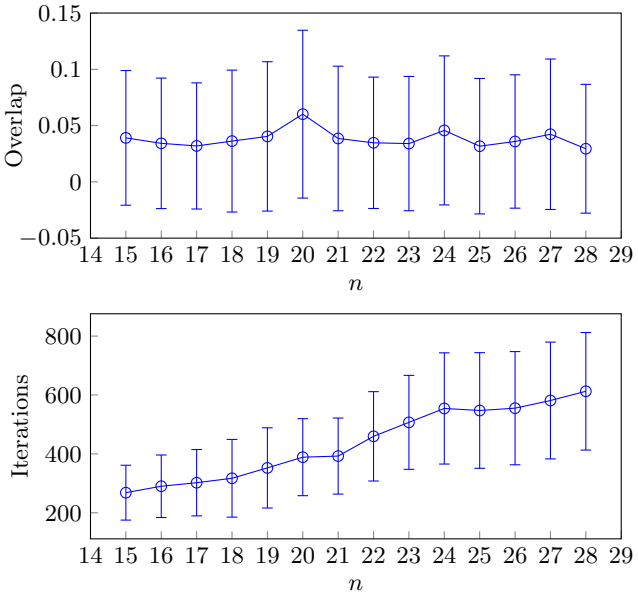
In this section, we perform numerical experiments to understand the number of qubits necessary to run the quantum enumeration depending on the quality of the dual basis. In cryptography, it is common to use LLL or BKZ reduced basis. Using Corollary 1, we draw Figure 1a which shows the number of qubits needed when the dual basis is LLL reduced or BKZ-β reduced with different β. We choose the maximum value of $\beta = 70$, since the running time becomes prohibitively long for larger β. We also performed experiments with dual pseudo-HKZ reduced basis to understand the practical behavior of the algorithm in the previous section[8][9]. The graph was obtained by generating random q-ary lattices (for $q = 65537$ and $k = n/2$), reducing them and then computing N as in (4) using the bounds in Lemma 1. Each experiment was repeated 5 times[10] and the average was taken. We also performed regression using quadratic polynomials for LLL and BKZ reduced basis. This type of regression is expected to fit well since the number of qubits can be shown to be quadratic. On the other hand, as shown in Theorem 1, for pseudo-HKZ reduced dual basis, the number of qubits should grow as $\Theta(n \log_2(n))$ asymptotically. Hence, we fitted with a curve of form $\frac{3}{2}n \log_2(n) + an + b$.

In order to solve the current lattice approximate SVP record (dimension 180), we need approximatively 1157 qubits when the dual is sufficiently reduced ($\beta = 70$ for example). This is considered to be achievable in the not so far future.

Fig. 1: Experimental results



(a) Average number of qubits as a function of the dimension and their regression; input lattices are q-ary; reduction performed the dual lattice; "Theorem 1" is worst case bound with $O(\log(n))$ term omitted; raw data linked in red.



(b) Top: Averaged overlap(inner-product) of state resulting from running the final ansatz circuit with Hamiltonian's 1st excited state (ground state is the zero vector), Bottom: Averaged number of VQE iterations until convergence

5 Experimental results

We run quantum emulation of the VQE algorithm on a classical computer to assess the performance of our SVP solver. Due to the limited number of emulable qubits we considered lattices with rank up

[8] Given the high cost of computing HKZ reduced basis, our experiments are limited to dimension 80.

[9] See Section 4.2 for the definition, a pseudo-HKZ reduced basis is slightly weaker than an HKZ basis.

[10] The q-ary lattices that we generate have fixed determinant, hence the number of qubits only depends on the orthogonality defect. We observed that the orthogonality defect of LLL and BKZ reduced basis varies little between runs, hence the small number of runs.

to 28, much smaller than the cryptographically relevant ranks but much bigger than prior quantum attempts. While we estimated the space-complexity required to solve large dimensions, we cannot reliably extrapolate the performance (accuracy/time) due to the heuristic nature of VQE. The effects of noise and error-mitigation are beyond the scope of this work and are left for future explorations.

5.1 Experimental framework

For our experiments we first sample $\boldsymbol{A} = [[\boldsymbol{I}_{d-k}, \widetilde{\boldsymbol{A}}]; [\boldsymbol{0}, q \cdot \boldsymbol{I}_k]] \in \mathbb{Z}^{d \times d}$ where $\widetilde{\boldsymbol{A}} \leftarrow\!\!\$\, \mathbb{Z}_q^{(d-k)\times k}$ and $\boldsymbol{I}_x \in \mathbb{Z}^{x \times x}$ is the identity. We then consider the lattice $\mathcal{L}(\boldsymbol{A})$ spanned by the rows of this matrix. In particular, we first LLL reduce the entire lattice basis and then consider the sublattice of rank n spanned by the first rows of the reduced basis. Our choice of q-ary lattices $\mathcal{L}(\boldsymbol{A})$ is partly motivated by their ubiquity in cryptography and by their connection to worst-case lattice problems. For example, finding short vectors in the dual of a random q-ary lattice is as hard as finding short vectors in any lattice [Ajt96]. We preprocess these lattices using the polynomial-time LLL algorithm for numerical stability reasons. We extract a n-rank sublattice in dimension d rather than considering full-rank lattices with $n = d$ since LLL will succeed in solving SVP for such small dimensions directly. The VQE receives a $n \times d$ matrix as input and this "leaves some work" for the quantum algorithm to do. Note that even though many instances of these sub-lattices were "almost" solved by the described procedure, it is little advantage for the VQE as the size of the search space was not reduced. Therefore, by using a random guess for initial optimization parameters of an ansatz circuit, our experiments represent scaled-down real instances. Consequently, the hardness of SVP for VQE is in our experiments not even lowered if the shortest vector is already present in the given basis. The search space is defined by bounds on the coefficient vector as discussed in Section 4.1. However, the bounds there are worst-case and asymptotic. In contrast, lattice algorithms tend to perform much better on average than in the worst case, especially in small dimensions. Indeed, Figure 1a illustrates the same phenomenon here. We hence present a different qubit mapping strategy for which we compute probabilities of the shortest lattice vector lying within these bounds and in our quantum experiments we evaluate the probability of finding the ground state of problem Hamiltonian. The advantage is that the two problems: choosing an appropriate qubit mapping and finding the ground state of the problem Hamiltonian, can be tackled separately. The overall success probability of the variational SVP solver is then a product of the probability of encoding the shortest non-zero lattice vector in the search space and the probability of finding the Hamiltonian's ground state.

Naive qubit mapping approach. Given an n-rank sublattice we assign one qubit per coefficient of the coefficient vector, i.e. $|x_i| \leq 1$ for $1 \leq i \leq n$. Using this approach we found that the shortest non-zero lattice vector was included in the search space with the following probabilities: *rank 15*: 80%, *rank 16*: 75%, *rank 17*: 75%, *rank 18*: 74%, *rank 19*: 71%, *rank 20*: 70%, *rank 21*: 64%, *rank 22*: 62%, *rank 23*: 57%, *rank 24*: 55%, *rank 25*: 50%. See Appendix B for analysis of naive qubit mappings with ranks up to 50 that also considers different naive mapping strategies. The results presented in the appendix can be of an interest for all experimental approaches to the *SVP* that use a Hamiltonian formulation.

Quantum emulation setup. We developed distributed variational quantum algorithms emulation software framework[11] by modification of parts of *Xacc* [MLD+20] library to utilize implementations of variational quantum algorithms and useful classical optimizers and we use *QuEST* [JBBB19] for the underlying quantum emulation. The experiments have been run on Ngio 5, a distributed computer provided by Edinburgh Parallel Computing Centre featuring 2x Xeon Platinum 8260 24C 2.4GHz with 12x 16GB DRAM modules. The emulations use state-vector representation of intermediate quantum states and do not consider any effects of noise. Noise would affect the probability of finding the ground state (uncertainty of cost makes the classical optimization harder) (Figure 1b Top), but it would not affect the time requirements (Figure 1b Bottom), neither the number of qubits. As it has been noted, extrapolating our results to lattice dimensions of cryptographic interest is infeasible from our experiments, because the way the probability of success decays cannot be extrapolated (among other reasons because we cannot anticipate the effects that barren plateaus would have). This is why for our current contribution adding noise would not add much in our analysis, beyond perhaps, bringing lower the probabilities of solving the problem in the very low qubit experiments that are possible to emulate.

[11] https://github.com/Milos9304/FastVQA, https://github.com/Milos9304/LattiQ

5.2 Experimental results

To improve the performance in our experiments we used $CVaR_\alpha$ cost for VQE. We chose $\alpha = 0.175$ since this gave better results (see also Appendix C). We were able to solve the SVP with our emulator for lattices with rank up to 28. Our experiments with 128 instances suggest that the success probability of finding the shortest non-zero lattice vector remains roughly constant for lattice instances with ranks not much larger than 28. Figure 1b (top) depicts the averaged overlap of the final ansatz state with the ground state corresponding to the shortest non-zero lattice vector found by a classical enumeration. The overlap represents the probability of sampling the shortest lattice vector with a single measurement of the final ansatz state. From the figure we see that we need ≈ 25 samples to obtain the solution. Moreover, we observed linear time scaling (see Figure 1b bottom). We note, however, that the ranks of cryptographically relevant lattices are larger (≈ 400) and we cannot extrapolate our observations with confidence.

References

ABF+20. Martin R. Albrecht, Shi Bai, Pierre-Alain Fouque, Paul Kirchner, Damien Stehlé, and Weiqiang Wen. Faster enumeration-based lattice reduction: Root hermite factor $k^{1/(2k)}$ time $k^{k/8+o(k)}$. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 186–212. Springer, Heidelberg, August 2020.

ACD+18. Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the {LWE, NTRU} schemes! In Dario Catalano and Roberto De Prisco, editors, *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, volume 11035 of *Lecture Notes in Computer Science*, pages 351–367. Springer, 2018.

Ajt96. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

ANS18. Yoshinori Aono, Phong Q. Nguyen, and Yixin Shen. Quantum lattice enumeration and tweaking discrete pruning. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*, pages 405–434. Springer, Heidelberg, December 2018.

BNR+20. Panagiotis Kl. Barkoutsos, Giacomo Nannicini, Anton Robert, Ivano Tavernelli, and Stefan Woerner. Improving variational quantum optimization using cvar. *Quantum*, 4:256, 2020.

CAB+21. M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, August 2021.

CL21. André Chailloux and Johanna Loyer. Lattice sieving via quantum random walks. Cryptology ePrint Archive, Report 2021/570, 2021. https://eprint.iacr.org/2021/570.

CLKAG21. Alba Cervera-Lierta, Jakob S. Kottmann, and Alán Aspuru-Guzik. Meta-variational quantum eigensolver: Learning energy profiles of parameterized hamiltonians for quantum simulation. *PRX Quantum*, 2:020329, May 2021.

DH76. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

DSvW21. Léo Ducas, Marc Stevens, and Wessel P. J. van Woerden. Advanced lattice sieving on GPUs, with tensor cores. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 249–279. Springer, Heidelberg, October 2021.

FGG14. Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

GLM08. Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, Apr 2008.

HKL+20. Philipp Hauke, Helmut G Katzgraber, Wolfgang Lechner, Hidetoshi Nishimori, and William D Oliver. Perspectives of quantum annealing: methods and implementations. *Reports on Progress in Physics*, 83(5):054401, may 2020.

HS07. Guillaume Hanrot and Damien Stehlé. Improved analysis of kannan's shortest lattice vector algorithm. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, pages 170–186, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

HWB19. Oscar Higgott, Daochen Wang, and Stephen Brierley. Variational Quantum Computation of Excited States. *Quantum*, 3:156, July 2019.

JBBB19. Tyson Jones, Anna Brown, Ian Bush, and Simon C. Benjamin. Quest and high performance simulation of quantum computers. *Scientific Reports*, 9(1), Jul 2019.

JCLM21. David Joseph, Adam Callison, Cong Ling, and Florian Mintert. Two quantum ising algorithms for the shortest-vector problem. *Phys. Rev. A*, 103:032433, Mar 2021.

JGLM20. David Joseph, Alexandros Ghionis, Cong Ling, and Florian Mintert. Not-so-adiabatic quantum computation for the shortest vector problem. *Phys. Rev. Research*, 2:013361, Mar 2020.

Kan83. Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, page 193206, New York, NY, USA, 1983. Association for Computing Machinery.

KEJ+20. B Koczor, S Endo, T Jones, Y Matsuzaki, and SC Benjamin. Variational-state quantum metrology. *New Journal of Physics*, 22(8), 2020.

KMW+17. Ian Kivlichan, Jarrod Mcclean, Nathan Wiebe, Craig Gidney, Aln Aspuru-Guzik, Garnet Chan, and Ryan Babbush. Quantum simulation of electronic structure with linear depth and connectivity. *Physical review letters*, 120, 11 2017.

KW21. Ioannis Kolotouros and Petros Wallden. An evolving objective function for improved variational quantum optimisation, 2021.

LLL82. A.K. Lenstra, H.W. Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.

Luc14. Andrew Lucas. Ising formulations of many np problems. *Frontiers in Physics*, 2:5, 2014.

Maz10. Gérard Maze. Natural density distribution of hermite normal forms of integer matrices. *Journal of Number Theory*, 131:2398–2408, 2010.

MLD+20. Alexander J McCaskey, Dmitry I Lyakh, Eugene F Dumitrescu, Sarah S Powers, and Travis S Humble. XACC: a system-level software infrastructure for heterogeneous quantum–classical computing. *Quantum Science and Technology*, 5(2):024002, feb 2020.

MRBAG16. Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, feb 2016.

NIS. NIST. Post-quantum cryptography standardization. Available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization.

PMS+14. Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1), July 2014.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

Sch87. Claus Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

Sho97. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):14841509, October 1997.

WC19. Jinming Wen and Xiao-Wen Chang. On the KZ reduction. *IEEE Trans. Inf. Theory*, 65(3):1921–1935, 2019.

WZdS+20. Roeland Wiersema, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen. Exploring entanglement and optimization within the hamiltonian variational ansatz. *PRX Quantum*, 1:020319, Dec 2020.

A Techniques to optimize towards first excited state of problem Hamiltonian

Section 3.1 shows that an Ising spin Hamiltonian whose eigenstates correspond to lattice vectors in a certain region can be constructed in a straightforward manner. However this implicitly encodes the zero vector as the ground state of the Hamiltonian. Since the output of the SVP problem is expected to be non-zero, a technique to essentially find the Hamiltonian's first excited state must be utilized. Note that this is equivalent with finding a new Hamiltonian that has as ground state the first excited state of our initial Hamiltonian. In the case of VQE there exists a natural solution to the problem that can be accommodated fully at the classical loop (classical post-processing) of the optimization. This happens by defining a cost function that excludes the contribution of the zero vector eigenstate and consequently guides the optimizer towards the first excited state. However, such modification cannot be done to QAOA nor can be executed on a quantum annealer as the cost function is strictly defined by the optimization strategy itself.

The first approach one might consider is ignoring the $x \neq 0^n$ and proceed with trying to optimize for the ground state. There is a chance that the overlap of the first excited state with final ansatz state (aimed to match closely the ground state/zero-vector) is non-zero and that it gets sampled during the final ansatz measurements. This approach does not increase any requirements on quantum resources, however, especially as the number of qubits increases, it is not expected to yield satisfiable results. The difference in cost between the zero vector and the first excited state (the SVP) increases and the probability of obtaining the first excited state while targeting the ground state decays possibly exponentially fast.

Instead, a more accurate and in the long term, viable solution is to modify the Hamiltonian and impose a penalty for reaching the zero vector (ground state of the "naive" Hamiltonian). Introducing the extra constraint, and ensuring that the new Hamiltonian is still QUBO (so that we can run QAOA or quantum annealing) requires to introduce auxiliary variables and thus requires more qubits. In our case, we succeed to obtain a theoretical guarantee that the optimization result converges to the shortest lattice vector in the limit of optimization parameters for the cost of introducing extra $n-2$ qubits for a lattice with rank n. Such approach hence roughly reduces the abilities of QAOA algorithm or quantum annealer device by a half compared to the VQE algorithm but allows us to benefit from a wider range of quantum variational approaches and hence we include it here for the reference. A trivial solution would be to penalize certain assignment of QUBO binary variables that makes the QUBO formulation zero. As we will argue below, given m QUBO binary variables, the approach would require to introduce $m-2$ new binary variables. Note that $m \geq n$ is the number of binary variables in QUBO. If we let n be rank of a lattice we can see that $m \geq n$ as shown in Section 3.1 where $m \approx n \log(2a)$ if each x_i has the same variable range $2a$ assigned. Suppose that instead of the mapping presented in Equation 2 we instead map each integer variable as

$$|x_i| \leq a \implies x_i = -a + \zeta_i a + \omega_i (a+1) + \sum_{j=0}^{\lfloor \log(a-1) \rfloor - 1} 2^j \tilde{x}_{ij} + (a - 2^{\lfloor \log(a-1) \rfloor}) \tilde{x}_{i,\lfloor \log(a-1) \rfloor} \tag{5}$$

by introducing new binary variables ζ_i, ω_i and $\{\tilde{x}_{ij}\}_{0\leq j\leq \lfloor \log a \rfloor}$ for eachi x_i.The encoding (5) allows us to encode penalization of zero state with much less qubits. It is easy to observe that for the last two terms of (5) it holds that

$$0 \leq \sum_{j=0}^{\lfloor \log(a-1) \rfloor - 1} 2^j \tilde{x}_{ij} + (a - 2^{\lfloor \log(a-1) \rfloor})\tilde{x}_{i,\lfloor \log(a-1) \rfloor} \leq a - 1$$

and consequently

$$x_i = 0 \implies \zeta_i = 1$$

Hence the penalization of the case where all integer variables are penalized for being zero $\forall x_i = 0$ is equivalent to penalization of the case where the same number of binary variables are one $\forall \zeta_i = 1$. The penalization term then becomes (6)

$$L \prod \zeta_i \tag{6}$$

where L is the penalty value. The term (6) can be encoded as (7) where $\{z_i\}_{1\leq i\leq n}$ are extra auxiliary binary variables with bijective correspondence to $\{\zeta_i\}_{0\leq i\leq n}$.

$$L(1 + \sum_{i=1}^{n} z_i(-(1-\zeta_i) + \sum_{k=i+1}^{n}(1-\zeta_k))) \tag{7}$$

Proof ((7) encodes constraint (6)). Let $x_i = 1 - \zeta_i$, $\tau_i = -x_i + \sum_{k=i+1} x_k$ and let $z_i = 1 \iff \tau_i < 0$. Then it is easy to see that values of $z_1, ..., z_n$ minimize (7). If $\forall x_i = 0$, the whole expression is equal to L as required. Otherwise, let $j_1, ..., j_m$ be positions of $m \leq n$ binary ones in the bitstring $x_1x_2...x_n$ sorted in the increasing order. Then $\tau_{j_m} = -1$ and $\tau_{j_l} \geq 0$ for $1 \leq l < m$. Hence $\forall_{i\neq j_m} z_i = 0$, $z_{j_m} = 1$ and the whole expression is 0, hence no penalty is imposed.

Observe that setting $z_n = 1$ and $z_{n-1} = \zeta_n$ does not change the global minimum of (7). Hence $n-2$ additional binary variables $\{z_i\}_{1\leq i\leq n-2}$ are needed to encode the penalization term for an n-rank lattice. The quadratic unconstrained binary optimization problem formulation that can be trivially mapped to Ising spin Hamiltonian hence becomes

$$min_{\substack{\mathbf{x_1},...,\mathbf{x_n} \\ z_1,...,z_{n-2}}} \sum_{i=1}^{n} \mathbf{x_i} G_{ii} + 2 \sum_{1\leq i<j\leq n} \mathbf{x_i x_j} G_{ij} + L(1 + \sum_{i=1}^{n} z_i(-(1-\zeta_i) + \sum_{k=i+1}^{n}(1-\zeta_k))) \tag{8}$$

where $\mathbf{x_i}$ is encoded as in (5). Number of binary variables needed to represent $\mathbf{x_i}$ bounded by a_i is $\lfloor \log a_i \rfloor + 3$. Hence the total number of binary variables in (8) is

$$n_{bin\_vars\_constrained} = \sum_{i=1}^{n}(\lfloor \log a_i \rfloor + 3) + (n-2) = 4n - 2 + \sum_{i=1}^{n} \lfloor \log a_i \rfloor$$

B Probability of including the shortest lattice vector in the search space using naive qubit mapping strategies

As discussed in Section 4 and 5.1, the bounds determined in Section 4.1 are not the best choice for small lattice instances. Given n dimensional lattice $\mathcal{L}$ and m available qubits, choosing a uniform distribution of qubits over elements of the coefficient vector x turns out to achieve a high probability of encoding the shortest non-zero lattice vector as the ground state of a problem Hamiltonian when m is sufficiently large. Figure 2 depicts the probabilities of encoding the shortest non-zero lattice vector as a Hamiltonian's ground state given the number of available qubits and the lattice rank. The results were averaged over 1024 instances prepared using the same methodology as for the quantum emulation experiments described in Section 5.1. Three approaches have been compared: 1. distributing $\left\lfloor \frac{m}{n} \right\rfloor$ qubits to each of the coefficient of x (blue) 2. randomly distributing $\left\lfloor \frac{m}{n} \right\rfloor$ qubits to $n \times \left\lfloor \frac{m}{n} \right\rfloor$ coefficients of x and distributing $\left\lfloor \frac{m}{n} \right\rfloor + 1$ to the remaining coefficients of x (red) 3. Using Lemma 1 with $A = gaussian\_heuristics(\mathcal{L})$ to

obtain a discrete distribution of the bounds over coefficients of x. The distribution is then scaled such that at most m qubits were distributed (green).

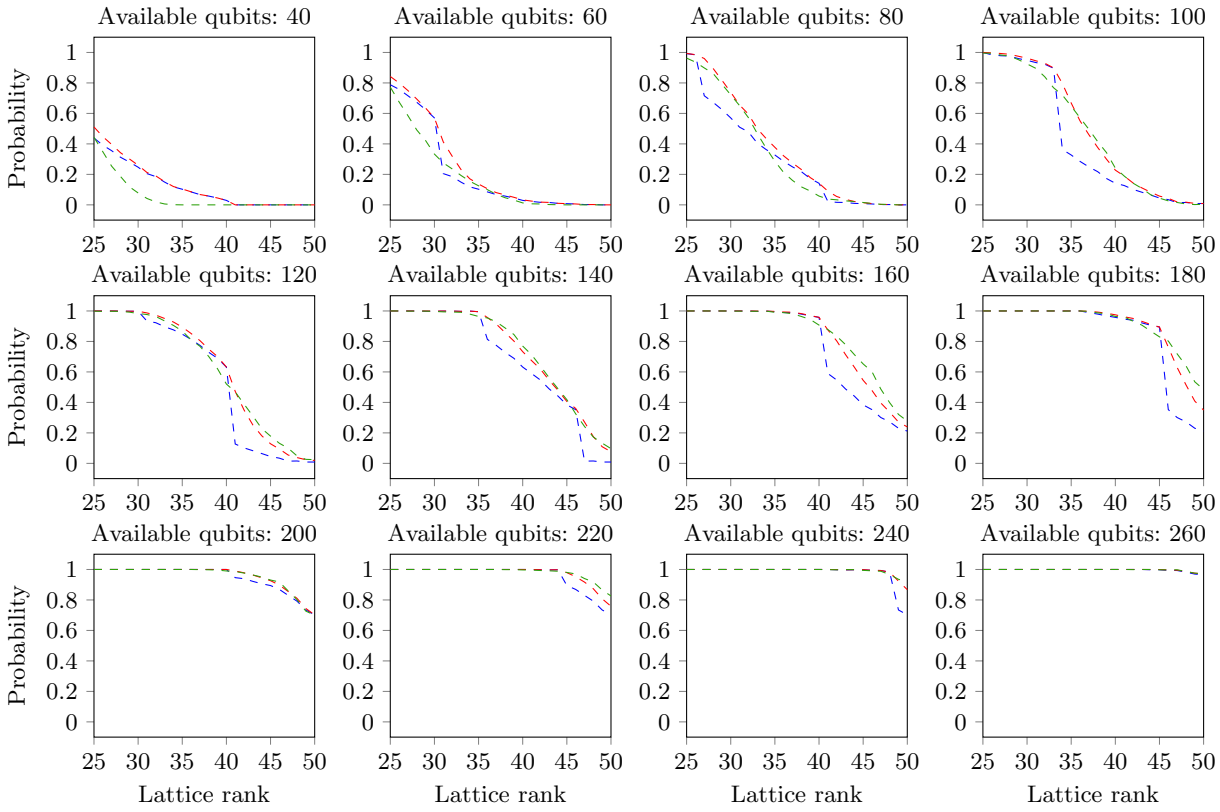


Fig. 2: Probability of encoding the shortest lattice vector as an eigenvector of problem Hamiltonian using **blue:** uniform distribution of qubits, **red:** uniform distribution of qubits and random distribution of the remaining qubits, **green:** scaled distribution of qubits according to Lemma 1 with A being a gaussian heuristics corresponding to the lattice instance

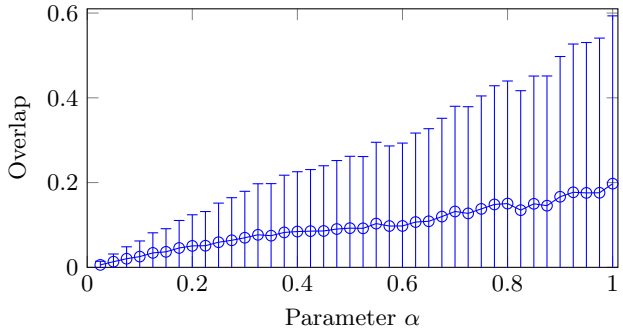
C Conditional value at risk experiment data

We chose the value of $\alpha = 0.175$ to use in the $CVaR_\alpha$ VQE algorithm in our quantum emulation experiments in Section 5. Based on an experiment involving 1024 instances of rank 16 lattices prepared as in Section 5.1 this value gave the best results. The value of α was varied incrementally in small steps. Figures 3a and 3b depict the mean and the median overlaps of the final ansatz state with Hamiltonian's ground state respectively. We can observe that despite positive linear correlation between the mean overlap and parameter α, the median overlap peaks sharply at $\alpha \approx 0.175$, an indication of a rightly-skewed distribution. In other words, we have seen significantly less instances with high overlap as $\alpha >\approx 0.4$ tends towards one, although when such overlaps happen, they are high enough so that their mean increases with α. The inverse of the overlap is the expected number of measurement samples of the final ansatz state to obtain the ground state. Hence a compromise must have been made to achieve as many as possible overlaps which were sufficiently high in the sense that the ground state could be obtained from them by a reasonable number of measurements. In order to make a fair comparison, we considered the probabilities of sampling the ground state if 5000 measurements of the final ansatz were to be performed as depicted in Figure 3c. We chose the value $\alpha = 0.175$ as it achieved around 78% probability of finding the Hamiltonian's ground state and at the same time it achieves the highest median of overlaps (around 0.006). Note that the non-$CVaR_\alpha$ VQE version, i.e. when $\alpha = 1$ resulted in only 21% probability of finding the Hamiltonian's ground state.

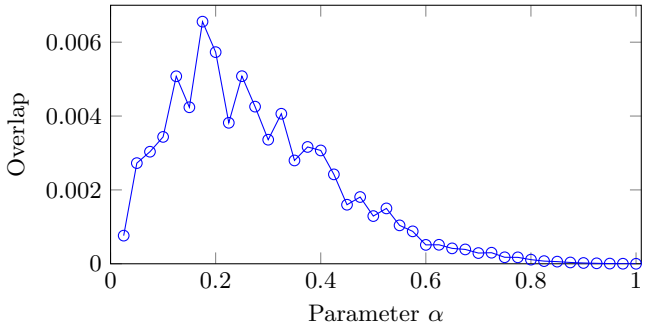
A final thing to note is that when one uses CVaR, the optimizer has incentive to find a quantum state that has α overlap with the true ground state, but no incentive to increase the overlap to higher values than α, since anything above the α-tail is irrelevant for computing the cost. Therefore, while it has been demonstrated that small values of α may increase the chances of finding the solution after a

fixed number of shots/measurements of the final state is carried out, the actual (mean) overlap does not necessarily increase (at least not to values above α). This was one of the motivation to look modify CVaR and define an Ascending-CVaR [KW21].

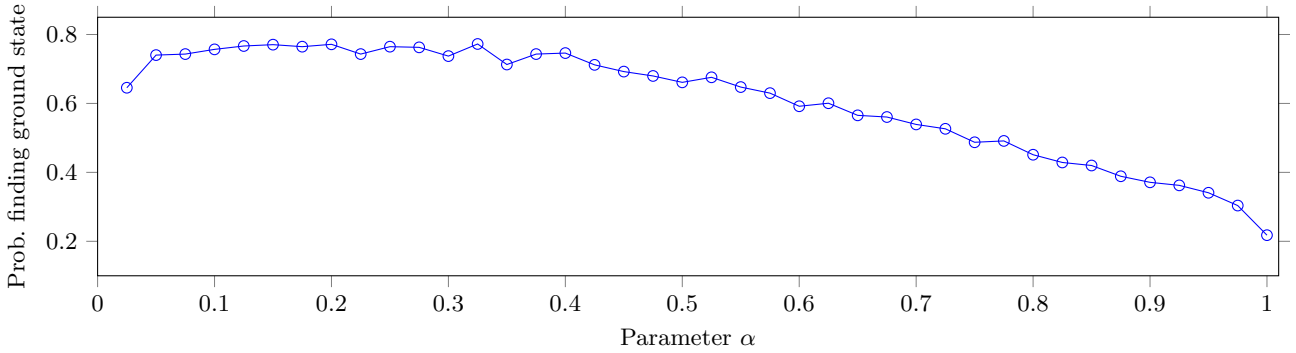
Fig. 3: Experimental determination of CVaR's parameter α



(a) Averaged overlap with the shortest lattice vector as a function of parameter α with standard deviations.



(b) Median overlap with the shortest lattice vector as a function of parameter α.



(c) Probability of sampling the ground state in 5000 samples using CVaR version of VQE as a function of parameter α. Note that $\alpha = 1$ corresponds to the standard non-CVaR version of VQE.

D Proof of Corollary 2

Let δ be the orthogonality defect. Then

$$\begin{aligned}
\log_2 \delta &\leqslant \tfrac{n}{2}\log_2 \gamma_n + \tfrac{1}{2}\log_2 \tfrac{(n+3)!}{3!} - n \\
&\leqslant \tfrac{n}{2}\log_2(\tfrac{1}{8}n + \tfrac{6}{5}) - n + \tfrac{1}{2}\log_2(n+3)! + O(1) \\
&= \tfrac{n}{2}\log_2(n) - \tfrac{5}{2}n + \tfrac{1}{2}\log_2(n+3)! + O(1) \\
&= \tfrac{n}{2}\log_2(n) - \tfrac{5}{2}n + \tfrac{1}{2}\log_2 n! + O(\log n) \\
&\leqslant \tfrac{n}{2}\log_2(n) - \tfrac{5}{2}n + \tfrac{1}{2}\log_2\left(\sqrt{2\pi n}\left(\tfrac{n}{e}\right)^n e^{\frac{1}{12n}}\right) + O(\log n) \qquad \textbf{by standard bounds} \\
&= \tfrac{n}{2}\log_2(n) - \tfrac{5}{2}n + \tfrac{n}{2}\log_2 n - \tfrac{n}{2}\log_2 e + O(\log n).
\end{aligned}$$