Not-so-adiabatic quantum computation for the shortest vector problem

David Joseph,[1] Alexandros Ghionis [●],[2,3] Cong Ling,[1] and Florian Mintert[3]

[1]*Electrical and Electronic Engineering Department, Imperial College London, South Kensington, London SW7 2BU, United Kingdom*
[2]*Physics Department, King's College London, London WC2R 2LS, United Kingdom*
[3]*Physics Department, Imperial College London, South Kensington, London SW7 2BU, United Kingdom*

Since quantum computers are known to break the vast majority of currently used cryptographic protocols, a variety of new protocols are being developed that are conjectured, but not proved, to be safe against quantum attacks. Among the most promising is lattice-based cryptography, where security relies upon problems like the shortest vector problem. We analyze the potential of adiabatic quantum computation for attacks on lattice-based cryptography, and give numerical evidence that even outside the adiabatic regime such methods can facilitate the solution of the shortest vector and similar problems.

I. INTRODUCTION

The advent of quantum computers heralds an age of new computational possibilities. Two paradigms of quantum computing are gate model and adiabatic quantum computation (AQC): the gate model closely resembles current computing architecture, replacing bits with qubits and retaining control over the smallest building-blocks of the system, and in AQC the solution for the problem to be solved is encoded into the ground state of a Hamiltonian [1,2]. Typically one cannot prepare this ground state directly, otherwise the problem would be straightforward to solve. One therefore begins with a physical system with a Hamiltonian whose ground state one knows how to prepare. The adiabatic theorem then guarantees that a sufficiently slow change from this initial Hamiltonian to the problem Hamiltonian H_P lets the system evolve into the ground state of the latter.

Both paradigms have been demonstrated to be equivalent [3], though there is not a general way of mapping from one paradigm to the other. The most impactful quantum algorithm discovered thus far is that of Shor for integer factorization and discrete logarithm computation [4]. Quantum computing is expected to have far reaching consequences, influencing materials science [5], development of medicines [6], and many other disciplines. Crucially for information security though, large-scale quantum computers—through application of Shor's algorithm—will make obsolete most currently operational cryptosystems by solving the underlying mathematical problems that are intractable on classical hardware.

———

A. Cryptography

When two parties (Alice and Bob) want to communicate securely over an insecure channel they must use public key cryptography. In this case Alice has a public/private key pair. Anyone can encrypt messages using the public key, but only Alice can decrypt these messages as only she knows the private key. This means Bob can communicate securely without having to already share a secret with Alice. Generally speaking, the public key is derived from the secret key in a manner which is not easily reversible. Public key cryptography is not efficient, and so is mostly used for exchanging an initial secret securely, from which point onwards Alice and Bob can use more efficient private key cryptography, which is not relevant to this paper.

Some of today's most prevalent public key cryptosystems are RSA, Diffie-Hellman key exchange, and ElGamal, the security of which rely on the hardness of integer factorization and discrete logarithm computation [7–9]. These are public key cryptosystems. Reverse engineering the secret key from only the public key and other public information amounts to cracking the cryptosystem, and this is what Shor's algorithm allows us to do for the schemes listed above.

These developments have necessitated the creation of entire new families of cryptosystems and the corresponding field of post-quantum cryptography [9]. The security of each family is based on the hardness of one of a handful of "contender problems." One of these families is lattice-based cryptography (or LBC). LBC is the most promising area, accounting for almost half of the remaining candidate systems in the NIST Post-Quantum Cryptography Standardization process. Lattice-based constructions derive their security from the shortest vector problem (or SVP; more in Sec. II) and other closely related problems [10]. At present these problems are only conjectured hard; i.e., there is no proof that quantum computers cannot solve them in polynomial time (BQP), there is only an absence of algorithms that can do so either provably or heuristically. It is therefore essential to analyze the security of post-quantum cryptosystems, so as to either

verify or disprove their resilience against attacks that may be aided by quantum logical elements.

B. Quantum computing in lattice-based cryptography

So far, most efforts towards quantum attacks on cryptographic protocols have concentrated on gate model quantum algorithms and their applications as subroutines to preexisting algorithms [11,12] which predominantly fall under either "sieving" or "enumeration." Sieving takes a large basket of vectors and iteratively combines them to obtain smaller and smaller vectors, whereas enumeration evaluates all vectors in a ball around the origin. Central to these gate model algorithms is the quantum Fourier transform (QFT). The most popular approach has been the use of Grover search to quadratically speed up search of unsorted lists in these algorithms. In 2015, however, it was observed that Grover search could not be applied to enumeration [11], but recently a quantum tree algorithm [13] was utilized to achieve square-root speed-up of lattice enumeration with discrete pruning [14]. QFT is also a key component of quantum hidden subgroup algorithms which have also been applied to the shortest vector problem on ideal lattices (these are structured lattices embedded in algebraic number fields) [15,16].

Solving this lattice problem is in essence backwards engineering a private key from the public key and other public information (i.e., reversing the trapdoor process previously mentioned), hence compromising any cryptosystems based on the hardness of this problem (and other related problems). The learning-with-errors cryptosystem [17], and many LBC trapdoor functions [18], for example, can be shown to be at least as hard as solving various lattice problems. The concept behind learning with errors is the addition of Gaussian noise to a lattice equation, which is otherwise easy to solve via systems of linear equations. An important innovation was the introduction of the smoothing parameter [19], which describes how much noise can be added before the structure of the lattice is lost and the problem becomes meaningless. This technique of using noise to obfuscate solutions could make LBC a fruitful field in which to apply optimization algorithms such as those enabled by AQC. At present the lattice community is still a long way from breaking these cryptosystems, as they tend to use lattices in hundreds or even thousands of dimensions, and the best algorithms at the time of writing scale exponentially in the dimension parameter.

The appeal of focusing on gate-model quantum algorithms is the rigorous complexity analyses that can be performed to give theoretical scaling. So far none of these gate-model algorithms threaten LBC. There has not yet been any work done on adiabatic quantum algorithms for LBC. Even though time complexity is generally difficult to estimate for this class of algorithms, they seem particularly suitable for attacks on LBC for two reasons: first, because lattice problems can be formulated as optimization problems, as we will demonstrate for a quantum setting; second, while a major drawback of AQC is the prohibitive time cost of achieving adiabaticity, this may not be a problem here as, up to a threshold, approximate solutions are also admissible. This is significant as it means it is *not* necessary to achieve adiabaticity, thereby potentially avoiding the major time constraints associated with AQC. In

this paper, we therefore employ AQC-style algorithms, but with subadiabatic time parameters.

In this work we demonstrate a mapping from the Euclidean norm of a vector to the energy of an ultracooled bosonic gas in a potential trap. To do so we use a generalized Bose-Hubbard Hamiltonian to describe the energy of the quantum system. We then present an AQC algorithm for solving one of the central lattice problems and analyze its performance on several instances of low dimensional lattices.

C. Structure

Section II introduces lattices and explains the shortest vector problem for which the algorithm is designed. It then covers the necessities regarding adiabatic quantum computing (AQC). Section III outlines the Hamiltonian we will use, and then we build the mapping from lattice vector norms to system Hamiltonian, ultimately combining this into one SVP algorithm. In Section IV we analyze both analytical scaling and simulation results.

II. PRELIMINARIES

Vectors are denoted by lowercase bold letters, matrices by upper case bold letters, and Hamiltonians by H. The length of a vector is defined in terms of a norm. For a vector $\mathbf{x} = (x_1, \ldots, x_N) \in \mathbb{R}^N$ we write $\|\mathbf{x}\|_p = (x_1^p + \cdots + x_N^p)^{\frac{1}{p}}$. Any value of $p \geqslant 1$ can be taken, but common choices are $p = 2$ and the infinity norm with $\|\mathbf{x}\|_\infty = \lim_{p \to \infty} \|\mathbf{x}\|_p = \max_i\{\|x_i\|\}$, i.e., the infinity norm l_∞. For any choice of p there are two shortest vectors (as lattices are symmetric about the origin). One vector is the negative of the other, and so we refer to "the" and not "a" shortest vector for simplicity. The length of this vector is denoted $\lambda_1(\mathcal{L})$.

When talking about approximation factors, we say $\gamma = \text{poly}(N)$ if γ grows asymptotically as $O(N^k)$ for some constant k, and $\gamma = \exp(N)$ if γ grows asymptotically as $O(k^N)$ for some constant k. Similarly, we say an algorithm takes polynomial time if it requires $\text{poly}(N)$ operations to complete, and exponential time if it requires $\exp(N)$ polynomial time operations to complete.

The dot product of two N-dimensional vectors is the canonical inner product on Euclidean space given by $\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + \cdots + x_N y_N$.

A. Lattices

Lattices simply put are a repeating pattern of points in space. In two dimensions, this looks similar to Fig. 1, which shows that the same lattice can be described by multiple different bases (red arrows and green arrows are just two different bases; there are infinitely many different bases for any given lattice, in fact). The volume of a lattice is equal to the magnitude of the determinant of the basis $|\det(\mathcal{L})|$, and represents the amount of ambient space inside its fundamental parallelepiped [10]. All bases for the same lattice must therefore have the same determinant, up to sign, but some contain much longer vectors than others, as can be seen by comparing the lengths of the red arrows with the lengths of the green arrows in Fig. 1. A lattice is described by one
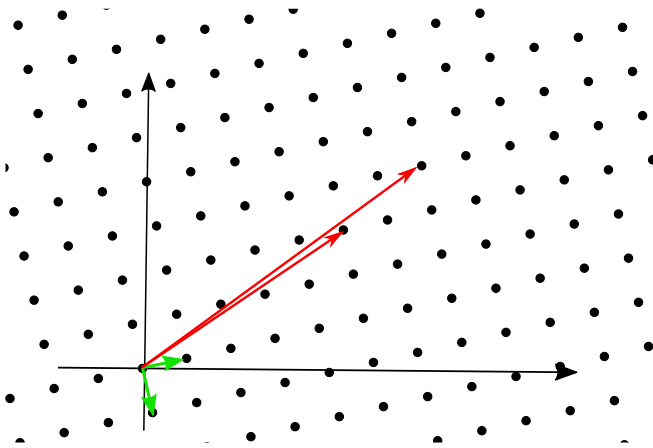
FIG. 1. A good basis (green) has short and nearly orthogonal vectors while a bad basis (red) is the opposite. The black arrows represent the x, y axes in $\mathbb{R}^2$, and the black dots an arbitrary 2D lattice.

basis vector for each dimension, and linear combinations of these basis vectors span the entire lattice; in Fig. 1 both red and green bases contain two linearly independent vectors. Mathematically, a lattice is a discrete additive subgroup of $\mathbb{R}^n$.

Definition 1. Lattice L is the set of integer combinations of k basis vectors $\mathbf{b}_i$, $1 \leqslant i \leqslant k$:

$$L = \left\{ \sum_{i=1}^{k} x_i \mathbf{b}_i \right\} = \{ \mathbf{x} \cdot \mathbf{B} : \mathbf{x} \in \mathbb{Z}^k \},$$

where the $\mathbf{b}_i$ are linearly independent and the lattice is embedded in the ambient space $\mathbb{R}^N$ for some $N \geqslant k$. The lattice is said to be full rank if $N = k$. Cryptographically, full rank lattices are the most relevant, and also the hardest for a particular dimension ambient space. Because of this, for the rest of this paper we will deal only with full rank integer lattices, i.e., those lattices for which the basis vectors have integer coordinates $\mathbf{b}_i \in \mathbb{Z}^N$. Throughout the rest of the paper we will treat $\mathbf{B}$ as a row basis,

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_N \end{bmatrix}. \tag{1}$$

There is no standardized convention in the cryptographic community (column bases vs row bases) and this choice is generally down to the author's preference. These lattices are in fact (a subset of) Euclidean lattices.

The central problem that we set out to address is the shortest vector problem (SVP) which is simply the task of finding the shortest nonzero lattice vector.

Definition 2. Shortest vector problem: Let $\lambda_1(\mathcal{L})$ denote the length of the shortest nonzero vector in a lattice L. Given a basis $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_N\}$ describing L, find the shortest nonzero vector such that

$$\lambda_1(\mathcal{L}) = \min\{ \|\mathbf{v}\| : \mathbf{v} \in L \setminus \{\mathbf{0}\} \}.$$

Given a lattice L determined by a basis $\mathbf{B}$, every vector $\mathbf{v}$ in the lattice can be described as a linear combination of the basis vectors $\mathbf{v} = \mathbf{x} \cdot \mathbf{B}$ as in Definition 1. We will call this linear

combination $\mathbf{x}$ the *coefficient vector* and denote the coefficient vector that achieves the shortest vector $\mathbf{x}_{\min}$ so that

$$\|\mathbf{x}_{\min} \cdot \mathbf{B}\| = \lambda_1(\mathcal{L}). \tag{2}$$

In the following we will denote the coordinates of a coefficient vector $\mathbf{x}$ as x_i and the coordinates of $\mathbf{x}_{\min}$ as $x_{\min}^i$ to avoid confusion of subscripts.

A variant of SVP is γ-approximate SVP.

Definition 3. SVP$_\gamma$: Given a basis $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_N\}$ describing a lattice L, find $\mathbf{v}$ such that

$$\|\mathbf{v}\| \leqslant \gamma \cdot \lambda_1(\mathcal{L}),$$

where $\gamma = \mathrm{poly}(N)$.

Cracking this problem is also conjectured hard, and solving it would be considered fatal for LBC. It is this problem that this work targets, because the quantum algorithm from Sec. III returns short vectors in the lattice with varying probabilities, as discussed in Sec. IV. As such, even if a quantum algorithm for finding *the* shortest vector is not feasible, finding somewhat short vectors may scale significantly better.

The format of the bases we work with to tackle the shortest vector problem have an important bearing on the speed with which we can accomplish the task. With that in mind we will outline the forms of basis that we utilize in this work.

A note on lattice bases. In LBC there is much talk of "good" bases and "bad" bases. It is important to distinguish the two and discuss their significance in solving the central problems and compromising lattice-based cryptosystems. A "good" basis is composed of short vectors which are approximately orthogonal to each other. The conditions of shortness and orthogonality are essentially the same, but they mean that good bases *already* contain short vectors. In a lattice-based cryptosystem one would generate a good basis as a private key (for example, in NTRU [20] and GGH [21], but not in LWE [17]), and scramble it (making it "worse" and the vectors less orthogonal) to create a bad basis, which would serve as a public key. The instances that are of interest to us are those of bad bases, from which we hope to derive short vectors.

Two types of bases that are relevant to us are Hermite normal form bases, which are useful in that they are upper triangular and allow us to perform some useful manipulation later in the paper, and the Lenstra-Lenstra-Lovász (LLL) - reduced bases, which are used as a benchmark and a starting point for many lattice algorithms.

Definition 4. Hermite normal form (HNF): For any integer lattice row basis $\mathbf{B}$ of rank N there exists a unique upper triangular basis $\mathbf{H}$ which satisfies the following conditions:

(1) $\mathbf{H}_{ij} = 0$ for $i > j$.

(2) The first nonzero term from the left (the pivot) is strictly to the right of the first nonzero term of the row above, and is positive.

(3) The elements beneath the pivot are zero, and the elements above the pivot are reduced modulo the pivot.

HNF bases form a good starting point for some of the work in Appendix B. They are generally quite bad bases but have some nice properties which we will use.

LLL-reduced bases [22] are better. The LLL algorithm runs in polynomial time, reducing bad bases to better ones; it outputs vectors which are exponentially larger than those
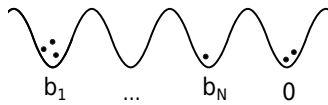
FIG. 2. The energy of the system of particles trapped in a potential landscape depicted above is proportional to the square of the Euclidean norm of the vector $\mathbf{v} = (3, \ldots, 1, 2) \cdot \mathbf{B}'$.

which would be considered solutions to SVP_γ, but is used as a benchmark in LBC cryptanalysis.

Definition 5. LLL-reduced basis: Given a basis $\mathbf{B} = \{\mathbf{b}_0, \ldots, \mathbf{b}_N\}$, define its Gram-Schmidt orthogonal basis $\mathbf{B}^* = \{\mathbf{b}_0^*, \ldots, \mathbf{b}_N^*\}$ and the Gram-Schmidt coefficients

$$\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle}.$$

Then the basis $\mathbf{B}$ is LLL-reduced if there exists $\delta \in (0.25, 1]$ such that

(1) Size reduction: For $1 \leqslant j < i \leqslant N$, $|\mu_{i,j}| \leqslant 0.5$.

(2) Lovász condition: For $k = 1, \ldots, N$, $\delta \|\mathbf{b}_{k-1}^*\|^2 \leqslant \|\mathbf{b}_k^*\|^2 + \mu_{k,k-1}^2 \|\mathbf{b}_{k-1}^*\|^2$.

LLL-reduced bases are not unique: there are potentially many different bases satisfying these conditions for any given lattice. In this respect they are different from HNF bases, which are unique for each lattice.

Definition 6. The Gram matrix $\mathbf{G}$ of a row basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_N\}$ is given by

$$\mathbf{G} = \mathbf{B} \cdot \mathbf{B}^T.$$

The Gram matrix will be enough to define H_P entirely, as $\mathbf{G}_{ij}$ is the dot product $\mathbf{b}_i \cdot \mathbf{b}_j$ and so will be used regularly in the following work.

B. Hamiltonian evolution

The model system that we will use in the following for AQC algorithms is based on the Bose-Hubbard Hamiltonian describing bosonic particles in potential landscapes with sufficiently well pronounced minima that can be identified as sites [23–26], and in practice these sites often form a periodic structure, as depicted in Fig. 2.

The explicit Hamiltonian

$$H_t = f(t)H_0 + g(t)(H_I + H_S) \tag{3}$$

is composed of a tunneling term H_0, an interaction term H_I, and an onsite-energy term H_S, where S represents "site." Here the sum of the interaction and onsite Hamiltonians give the problem Hamiltonian, $H_P = H_I + H_S$. The annihilation (a_i), and creation ($a_i^\dagger$) operators

$$a_i |n_i\rangle = \sqrt{n_i} |n_i - 1\rangle,$$
$$a_i^\dagger |n_i\rangle = \sqrt{n_i + 1} |n_i + 1\rangle, \tag{4}$$
$$a_i^\dagger a_i = \hat{n}_i,$$

decrease or increase the particle number at site i by 1. These operators are used to define the tunneling term

$$H_0 = -\sum_i (a_i a_{i+1}^\dagger + a_{i+1} a_i^\dagger), \tag{5}$$

where index i runs over the sites in the potential landscape. It will later become clear that the number of sites corresponds directly to the dimension of the lattice in which a short vector is sought.

The interaction Hamiltonian

$$H_I = \sum_i v_{ii} \hat{n}_i (\hat{n}_i - 1) + \sum_{i \neq j} v_{ij} \hat{n}_i \hat{n}_j \tag{6}$$

and onsite Hamiltonian

$$H_S = \sum_i \mu_i \hat{n}_i \tag{7}$$

together define the problem Hamiltonian, and the interaction constants v_{ij} and onsite energies μ_i will be determined by the underlying Euclidean lattice. In particular, it will be essential to consider not only onsite interactions $\sum_i v_{ii} \hat{n}_i (\hat{n}_i - 1)$ and interactions between neighboring sites but also long-range interactions.

With the choice of $f(0) = 1$ and $g(0) = 0$, the system Hamiltonian contains initially only the tunneling term. It has comparatively simple eigenstates, and the system can thus be initialized in its ground state. As soon as the values of $f(t), g(t)$ differ from their initial values, the system state will start to evolve in time, but the system will remain in the instantaneous ground state of its current Hamiltonian if the values of $f(t), g(t)$ change sufficiently slowly [27]. In general, f, g need not be continuous, but, to achieve adiabaticity, continuity is necessary (but not sufficient). Given the validity of such adiabatic dynamics, the system will thus end up in the ground state of the problem Hamiltonian $H_I + H_S$ at the final point in time with $f(T) = 0$ and $g(T) = 1$.

Since, besides the requirement of sufficiently slow changes, there are no further restrictions on $f(t)$ and $g(t)$, there is a continuum of possible sweeps. Knowledge of the spectrum of the underlying Hamiltonian could be used to find functions that make the adiabatic approximation particularly good. Since, however, the AQC should be applicable to the case in which finding this spectrum is beyond computational capabilities, we will not assume any suitably chosen functions, but simply a linear sweep

$$H_t = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} (H_I + H_S) \tag{8}$$

throughout the rest of this paper.

III. QUANTUM ALGORITHM

In this section we formulate the quantum SVP algorithm and detail the mapping from vector norms to the Hamiltonian of Eq. (3).

A. Problem Hamiltonian to l^2 norm

The interaction term has an explicit distinction between the interaction $v_{ii} \hat{n}_i (\hat{n}_i - 1)$ of particles at the same site and the interaction $v_{ij} \hat{n}_i \hat{n}_j$ between particles at different sites. This distinction is necessary because $\hat{n}_i$ particles interact only with the remaining $\hat{n}_i - 1$ particles at the same site i, whereas $\hat{n}_i$ particles at site i interact with all $\hat{n}_j$ particles at site j. On the other hand, there is the onsite interaction term, and the onsite energies can always be chosen such that they compensate for

the difference between the onsite and offsite interactions, i.e., such that

$$H_I + H_S = \sum_{i,j} \tilde{v}_{ij} \hat{n}_i \hat{n}_j, \qquad (9)$$

which maps to the l^2 norm of a vector in a natural fashion.

A vector $\mathbf{v} \in L \setminus \{\mathbf{0}\}$ can be written as a unique combination of the basis vectors $\mathbf{b}_i$:

$$\mathbf{v} = x_1 \mathbf{b}_1 + \cdots + x_N \mathbf{b}_N,$$
$$\mathbf{v} = \mathbf{x} \cdot \mathbf{B}, \qquad (10)$$

remembering that $\mathbf{B}$ is a row basis for L, where $\mathbf{b}_i$ are the rows. Expanding the square of the Euclidean norm of this vector term by term, we have

$$\|\mathbf{v}\|^2 = (x_1 \mathbf{B}_{11} + \cdots + x_N \mathbf{B}_{1N})^2$$
$$+ \cdots$$
$$+ (x_1 \mathbf{B}_{N1} + \cdots + x_N \mathbf{B}_{NN})^2. \qquad (11)$$

This can be expressed as

$$\|\mathbf{v}\|^2 = (x_1^2 \mathbf{b}_1 \cdot \mathbf{b}_1 + \ldots + x_N^2 \mathbf{b}_N \cdot \mathbf{b}_N)$$
$$+ 2(x_1 x_2 \mathbf{b}_1 \cdot \mathbf{b}_2 + \cdots + x_{N-1} x_N \mathbf{b}_{N-1} \cdot \mathbf{b}_N). \qquad (12)$$

Referring to (13), this form neatly fits that of the problem Hamiltonian H_P with the identification of $\tilde{v}_{ij}$ with the scalar product $\mathbf{b}_i \cdot \mathbf{b}_j$ of two basis vectors, and of $\hat{n}_i$ with the integer expansions coefficients x_i. Generally, an experimentally observable expectation value of particle number at any given site does not need to be an integer, but at the end of the algorithm, where the tunneling term is vanishing, any local particle number is indeed well defined without quantum fluctuations, so that the identification of $\hat{n}_i$ with x_i is justified.

In terms of Definition 6, the problem Hamiltonian for the AQC thus reads

$$H_P = \sum_{ij}^N \mathbf{G}_{ij} \hat{n}_i \hat{n}_j, \qquad (13)$$

where there are N sites, with the energy of a single particle at site i corresponding to the length of basis vector $\mathbf{b}_i$. All the interaction constants $\mathbf{G}_{ij}$ are defined by the basis $\mathbf{B}$, and any non-negative number of particles can be found at any site, subject to availability of particles. Running this algorithm with K particles, they could theoretically occupy the sites in any non-negative combination summing to K, resulting in a Hilbert space of D Fock states [28], where D is

$$D = \frac{(K + N - 1)!}{K!(N-1)!}, \qquad (14)$$

and the Fock state with the lowest energy H_P has a configuration of particles that, when interpreted as the coefficient vector $\mathbf{x}$, gives the shortest possible vector norm $\|\mathbf{v}\|$ under the constraint that $x_i \geqslant 0$, $\forall\, 1 \leqslant i \leqslant N$.

See Appendix C for a worked-though example demonstrating the theory outlined up to this point.

B. Adaptation to negative coefficients

The mapping so far transforms the Euclidean norm squared of general lattice points $\mathbf{v} = \mathbf{x} \cdot \mathbf{B}$ into the problem Hamiltonian energy where the x_i is the number of particles at each site $\hat{n}_i$. The dilemma that this presents is that this only permits non-negative values for each of the x_i. This is not a problem, however, as we show next how to modify the physical system such that H_P generalizes so as to return solutions that relate to negative x_i values.

The solution that we propose is to add Nm extra particles to the system (m particles for each site), and then by a change of variables use these particles as an offset, thereby permitting negative coefficients x_i. The coefficients x_i can now take values as low as $-m$, which occurs if the particle number at site i is zero. If there are m particles at site i then $x_i = 0$ and so on. The new particle number can be written $n_i = \hat{n}_i + m$. Denote this new problem Hamiltonian H_P'. Upon substituting this change of variables into Eq. (13) the new problem Hamiltonian H_P' becomes

$$H_P' = \sum_{ij} \mathbf{G}_{ij} (\hat{n}_i + m)(\hat{n}_j + m),$$

$$H_P' = H_P + 2m \sum_i \hat{n}_i \left(\sum_j \mathbf{G}_{ij} \right) + m^2 \sum_{ij} \mathbf{G}_{ij}. \qquad (15)$$

To obtain the desired minimization of the problem Hamiltonian H_P from H_P', the onsite energy needs to be reduced by a function of the column sum of the interaction matrix. The final term, being constant, can be corrected at a later stage so as to return the correct short lattice vectors but would not affect the energy spectrum of the Hamiltonian (other than a constant shift) or, consequently, which configuration of particles minimizes the system energy.

To guarantee that the shortest vector lies in the solution set, the offset m must be larger than the infinity norm of the coefficient vector $\mathbf{x}_{\min}$. That is, $m \geqslant \|\mathbf{x}_{\min}\|_\infty$ where $\mathbf{x}_{\min} \cdot \mathbf{B} = \lambda_1(\mathcal{L})$.

C. Multirun quantum SVP

Above we have defined a mapping from the Euclidean length of a vector to the energy of an ultracooled bosonic gas trapped in a potential landscape. But choosing the parameters for total particle number K and length of time evolution T then performing the quantum algorithm are not enough, on their own, to obtain the shortest vector. One run of the algorithm described above contains K particles, but the Fock states in the solution space may not correspond to the required linear combination $\mathbf{x}_{\min}$.

Take for example a two-dimensional (2D) lattice basis for which the shortest vector is determined by $\mathbf{x}_{\min} = (3, 0)$; then $\lambda_1(\mathcal{L})$ is found by

$$\|(3, 0) \cdot \mathbf{B}\| = \lambda_1(\mathcal{L}). \qquad (16)$$

To obtain the shortest vector using the algorithm detailed above (assuming no prior knowledge about $\mathbf{x}_{\min}$, and for simplicity setting offset $m = 0$) one would first run with particle number $K = 1$, then with $K = 2$, $K = 3$, and then possibly repeat a few more times to be sure the shortest vector has indeed been found. In this way, a search for the shortest vector consists of running the algorithm many times, each time incrementing K by 1 until confident that there are no shorter
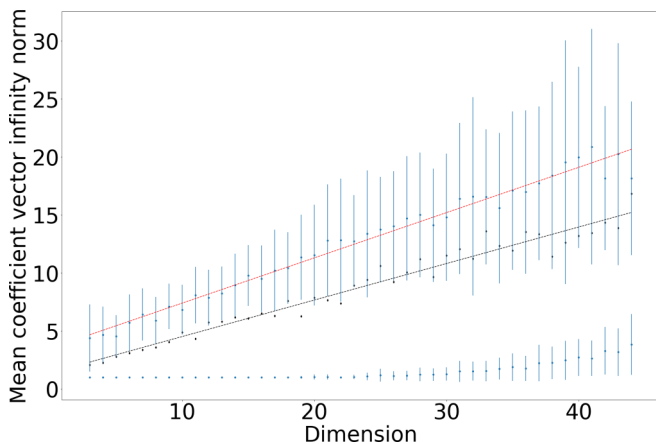
FIG. 3. Mean $\|\mathbf{x}_{\min}\|_\infty$ with error bars averaged over 80 random integer lattices in each dimension $3 \leqslant N \leqslant 45$ calculated on the Hermite normal form basis (upper distribution with red dashed best fit) and LLL reduced bases (lowest distribution). The mean absolute coefficient vector sum ($k = |\sum_{i=1}^{N} x_i|$)—with best fit as the black dashed line—demonstrates that $N(m + 1)$ particles' scales are enough to ensure a good chance of containing λ_1 in the solution set, and also represents the expected number of runs in multirun.

vectors to be found. The output of this algorithm, if performed adiabatically, will be a collection of coefficient vectors $\mathbf{x}$ and the resulting lattice vectors $\mathbf{v} = \mathbf{x} \cdot \mathbf{B}$, each of which returns the shortest lattice vector possible for a particular choice of K, and among these samples will be the sought after shortest vector of length $\lambda_1(\mathcal{L})$.

The multirun algorithm ensures that with well chosen m, $K_{\max}$, and sweep times the $\lambda_1(\mathcal{L})$ will definitely be correctly identified. This is after performing the sweeps for particle numbers $K_i = Nm + i$ up to $K_c = K_{\max} = Nm + c$. The number of runs required is therefore c, which should approximate the absolute coefficient vector sum $|\sum_{i=1}^{N} x_i|$. As can be seen in black in Fig. 3, this grows linearly in N, meaning $O(N)$ repetitions would be expected in order execute multirun, with each repetition having a different number of particles and being carried out adiabatically.

In the next part we present a more eloquent all-in-one algorithm where many different runs from this algorithm are combined into one larger run. It offers an $O(N)$ improvement in space but at the cost of $\lambda_1(\mathcal{L})$ no longer corresponding to the ground state, but instead to the first excited state.

Appendix C illustrates what one of the runs would look like for a 2D lattice with no offset ($m = 0$).

D. Single-run quantum SVP

The aim of this algorithm is to generalize multirun into one overarching algorithm (single run) that encompasses all of the repetitions executed during the multirun algorithm. Whereas before the coefficient vectors (in 2D) $(1, 0)$ and $(1, 1)$ could be obtained only from separate runs, now the aim is to include all possible coefficient vectors in one solution space. Instead of repeating sweeps with a different particle number K times, only one sweep is performed, the solution space of which includes all possible solutions from the multirun version.

What we propose is to introduce an extra site to the potential landscape, corresponding to the zero vector. Label this

site $N + 1$. This new site should act as a "particle reservoir" and *not* influence the energy of the system directly. For the Euclidean lattice, one appends the zero vector to the basis $\mathbf{B}$, as defined in Eq. (18). Denote the particle number for the single-run version K_S. If the process is run with $K_S \geqslant K_{\max}$ total particles, then the set of configurations where no particles are in site $N + 1$ correspond to one run of the previous algorithm with K_S particles; the set of configurations with one particle in site $N + 1$ correspond to a run of the previous algorithm but with $K_S - 1$ particles, and so on. It is important to remember that the ground state is no longer the shortest vector, but the zero vector (m particles in the first N sites and all the remaining particles in site $N + 1$ returns $\mathbf{0}$ in the lattice, which has the lowest energy of all solutions). As such, an adiabatic evolution is no longer desirable, because this solution is trivial: At least one excitation is required to obtain a nonzero lattice vector. We analyze the implications of this in the following sections.

Accordingly, the single-run problem Hamiltonian H_P' looks as follows:

$$H_P' = \sum_{ij}^{N+1} \mathbf{G}_{ij}'(\hat{n}_i + m)(\hat{n}_j + m), \tag{17}$$

where $\mathbf{G}_{ij}'$ is defined as $\mathbf{B}'\mathbf{B}'^T$, for

$$\mathbf{B}' = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_N \\ \hline \mathbf{0} \end{bmatrix}. \tag{18}$$

The problem Hamiltonian in Eq. (17) will be the one used for the rest of the paper, unless otherwise stated, including all numerical simulations.

The size of the Hilbert space D_S for single-run quantum SVP (letting $K_S = K_{\max}$ to make the two modes of computation directly comparable) is, either by use of the hockey-stick identity or by direct application of Eq. (14),

$$D_S = \frac{(K_S + N)!}{K_S!N!} = \sum_{K=0}^{K_{\max}} \frac{(K + N - 1)!}{K!(N - 1)!}. \tag{19}$$

Having added an extra "particle reservoir" site and accordingly appended $\mathbf{0}$ to the basis, the lowest energy state of the problem Hamiltonian is the unwanted $\mathbf{0}$ state, but the energy of the first excited state will correspond to $\lambda_1(\mathcal{L})$.

IV. RESULTS

Little is known analytically about the time scaling for adiabatic quantum algorithms, beyond a worst case energy gap dependence of $1/\Delta^3$ [29]—with Δ denoting the minimum energy gap between ground state and second lowest eigenstate—whereas with quantum gate algorithms neat closed form scalings are known for a handful of algorithms, for example Shor's exponential speedup for integer factorization and discrete logarithm computation [4] and Grover's quadratic speedup for searching unsorted lists [30]. For adiabatic quantum optimization, it is not yet even known if these algorithms run faster than classical optimization [31]. Due to

time dependence on the minimum energy gap between E_0 and E_1 it is usually found that

$$T = O(\exp(\alpha N^\beta)), \tag{20}$$

though this leaves open the possibility of drastically reducing run time subject to achieving lower values of α, β than their classical analogues.

While it is difficult to estimate the time scaling for this algorithm, or even for which parameter regimes this scaling would be optimal (near adiabatic versus much faster sweeps, for example), we can calculate the qubit space requirements (though we do not directly use qubit-based architecture).

A. Qubit requirements

Let us estimate the required system size. This is a function of number of sites and number of particles K_S. The former is predetermined (it is $N + 1$) but one can choose the latter. The aim is to make the system just large enough (pick K_S) so that $\mathbf{x}_{\min}$ is one of the possible configurations of particles in $N + 1$ sites with high probability.

First m, the offset, must be chosen. We derive this by taking an estimate for the infinity norm of $\mathbf{x}_{\min}$. This can be seen in Fig. 3 for HNF bases (to err of the side of safety, as this will give larger values because HNF bases have long vectors). One can see that the average of the infinity norms grows linearly for HNF bases (see the red best fit line), so we approximate m to be some linear function of N. This means that K_S is already up to $m(N + 1)$ particles, and so K_S scales as $O(N^2)$.

The only other consideration is the absolute value of the sum of the coefficient terms $k = |\sum_{i=1}^{N} x_{\min}^i|$. Heuristically this grows linearly, as shown by the black best fit line in Fig. 3, and is less than m so can be ignored from this point on, as in the instance that all $m(N + 1)$ particles reside in the first N lattice sites (and none in the particle reservoir corresponding to the zero vector) the net coefficient sum would be $m > k$. This is because Nm particles are acting as offset particles, leaving the remaining m as the coefficient sum.

By considering the size of the solution space, we have analytically deduced that the qubit requirements scale as $O(N \log N)$ as shown in Appendix A, which space-wise appears acceptable.

B. Empirical results

The required particle number K is determined by $|\sum_{i=1}^{N} x_{\min}^i|$, which must be estimated. This reflects the fact that on average some coordinates of $\mathbf{x}$ will be positive, and some negative, canceling out, but they will rarely cancel out entirely. The growth in the mean of $|\sum_{i=1}^{N} x_{\min}^i|$ is linear, as demonstrated numerically in Fig. 3, and grows below the estimated offset number m. The significance of this is that $m(N + 1)$ particles is a generous estimate for K_S. This means that taking $K_S = m(N + 1)$ gives a good chance of finding $\mathbf{x}_{\min}$ in the solution set.

To understand this, consider the system with mN particles in the first N sites and m particles in site $N + 1$. The potential solutions are the same as those one would get from performing the multirun version with mN total particles. Now with one more particle in the first N sites and one fewer in site $N + 1$

the solutions are the same as those from multirun with $mN + 1$ total particles. To ensure there are enough particles in the system, one must be confident of achieving up to $mN + |\sum_{i=1}^{N} x_{\min}^i|$ particles in the first N sites. That Fig. 3 demonstrates that $|\sum_{i=1}^{N} x_i| < m$ in general means that the total of $m(N + 1)$ particles is sufficient to find $\lambda_1(\mathcal{L})$.

The growth in $\|\mathbf{x}_{\min}\|_\infty$ with respect to Hermite normal form bases appears heuristically linear, as can be seen by the red dashed best-fit line in Fig. 3.

Another group of cryptographically relevant bases are LLL-reduced bases. These are much better than HNF bases and are easy to obtain. They are often used as a first step in classical SVP routines [12,32]. LLL-reduced bases in low dimensions (<30 or so) were so efficient at finding the shortest vector that there were not enough data points to draw sound conclusions from, as can be seen in the lower blue scatter plot in Fig. 3. We can, however, assert that the LLL-reduced case is upper bounded by the Hermite normal form case and so can approximate m to be a term linear in N. This is a sound assertion because any lattice basis can be transformed into an HNF basis in polynomial time.

C. Numerical analysis

Ideally simulating this quantum SVP algorithm on lattices in many dimensions would give an empirical idea of scaling. Regrettably, simulating quantum systems is computationally very intensive due to the factorial growth of the Hilbert space and so these simulations were only possible for low dimensional lattices. Using the QuSpin Python library we were able to simulate problems with Hilbert space sizes of up to ten thousand eigenstates (20 particles in 5 sites).

Nevertheless it is insightful to consider the distribution over eigenstates (grouped where degenerate) for runs of different time length. We simulated the quantum SVP algorithm on 200, 150, and 100 lattices in two, three, and four dimensions respectively. Using standard "bad" bases from literature does not work well for small dimensions—both HNF and LLL reduction tend to return maximally reduced bases—so we generated our own as follows. For each lattice we generated a basis (call this the "good" basis) and then scrambled it by some randomly generated unimodular matrix to obtain a worse basis. The average increases in basis vector length are factors of 12.06, 10.08, and 10.07 in dimensions two, three, and four respectively under the unimodular transformations. Basis vectors could not be increased by too much otherwise the problems would have become intractable on our hardware. Note also that after generating the Hamiltonians in the QuSpin package we scaled the Hamiltonians so that they all occupied roughly the same spectrum of eigenvalues. The reason for this is that expanding the energy spectrum significantly increases success probabilities for quantum adiabatic algorithms due to the dependence on minimum energy gap. Scaling Hamiltonians has an effect similar to altering the sweep times which we want to analyze, and it is reasonable to expect that implementers of this algorithm would have access to the same energy spectrum regardless of the problem size. As such, we in a sense "fixed" the spectrum and varied the sweep times to isolate the effect of the parameter T. Throughout all numerical simulations natural units were used, setting $\hbar = 1$.
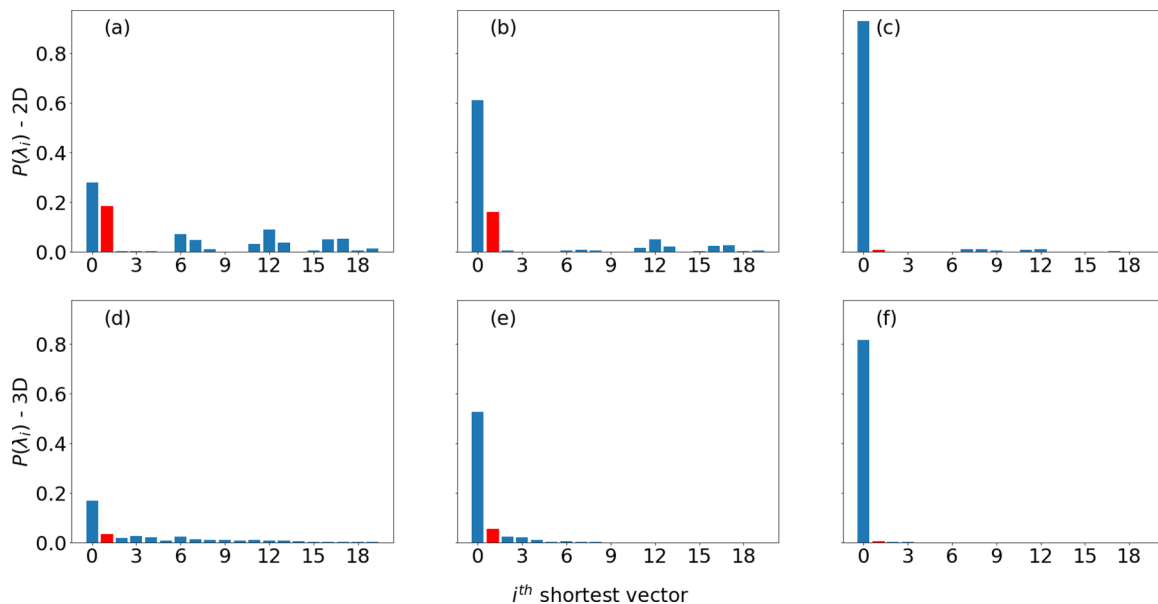
FIG. 4. Probability of returning ith shortest vector averaged over 200 instances of two-dimensional lattices for $T = 1, 10, 100$ in plots (a), (b), and (c) respectively, and over 50 instances of three-dimensional lattices for $T = 1, 10, 100$ in plots (d), (e), and (f) respectively.

Consequently all energies are measured in electronvolts (eV) and all sweep times in eV^{-1}.

Mean distribution over eigenstates. Figure 4 shows the averaged results for the single-run quantum SVP algorithm of Sec. III. Each subplot represents a different choice of parameters, and shows the mean probability of observing the system in an eigenstate corresponding to the zero vector (index 0), the shortest vector $\lambda_1(\mathcal{L})$ (index 1, in red), the second, third, etc. shortest vectors and so on up to the twentieth shortest vector. What is clear to see is a high likelihood of the system being found in low-energy states. For slower sweeps (higher T values) this distribution becomes more concentrated around the lowest-energy states.

Paying particular attention to the red bars [representing the preferred solutions corresponding to $\lambda_1(\mathcal{L})$] one can see that maximizing the height of the red bar requires some nuance: sweep too slow and there is too high a chance of attaining the zero vector at an unacceptable time cost; too fast and the system will become excited to much higher energy levels with unacceptably low probabilities of observing the system in very low energy states.

The top row of Fig. 4 displays the final results for parameter choices $N = 2$, $m = 3$, $T = 1, 10, 100$. This maps to a system of nine particles in three sites. Applying Eq. (14), the total Hilbert space has 55 eigenstates. The bottom row of Fig. 4 shows the results for parameter choices $N = 3$, $m = 4$, $T = 1, 10, 100$. This is for a system of sixteen particles in four sites. The Hilbert space now has 969 eigenstates. In both $N = 2, 3$ slower sweeps result in higher probabilities of the system terminating in the lowest eigenstates, and for $T = 100$ there is a very low likelihood of finding the system in anything but the ground state. It should also be observed that as the system size increases from $N = 2$ to $N = 3$, keeping T constant, the probability of recording the system in any given low-energy state decreases.

Time sweep optimization

While reliable time bounds for finding ground states of quantum adiabatic algorithms with good enough probability are much sought after, and the general rule of thumb is "longer is better," thought is shifting. There are instances, for example, in the MAX 2-SAT problem, in which slow sweeps perform much worse than fast ones [33]. Here we instead consider faster sweeps and how they relate to algorithms where the exact ground state is not necessarily required.

The problem of observing the ground state of a system after a quantum annealing algorithm has been the subject of much research. In the pursuit of "somewhat" low energy states much less is known. The algorithm outlined in Sec. III targets the first excited state, and, furthermore, the ground state (corresponding to the zero vector) is of even less use than eigenstates of energy just above $\lambda_1(\mathcal{L})$, as at least these return a short nonzero contender. Bearing this in mind along with the observations from Fig. 4, we thought to examine how targeting low-but-not-ground states versus ground state might differ.

The solid lines in Fig. 5 represent mean probability of returning $\mathbf{0}$, $\lambda_1(\mathcal{L})$, $\lambda_2(\mathcal{L})$ in blue, red and green respectively. The dashed lines reflect 90% confidence.

Looking first at the probability curves, there are a few interesting observations to be made. In two and three dimensions, exponentially slower sweeps result in higher probabilities of achieving a final ground state, as might be expected. These blue curves will continue to level off past the right of the axes due to maximum probability of any state being 1. But, remarkably, targeting the first excited state appears to experience almost no success penalty for performing faster sweeps. In fact, the nonmonotonicity of the red dashed line indicates that there is some "Goldilocks" zone where evolutions are slow enough to achieve a good distribution over low energy states, but not slow enough that $P(E_0)$ dominates the distribution.
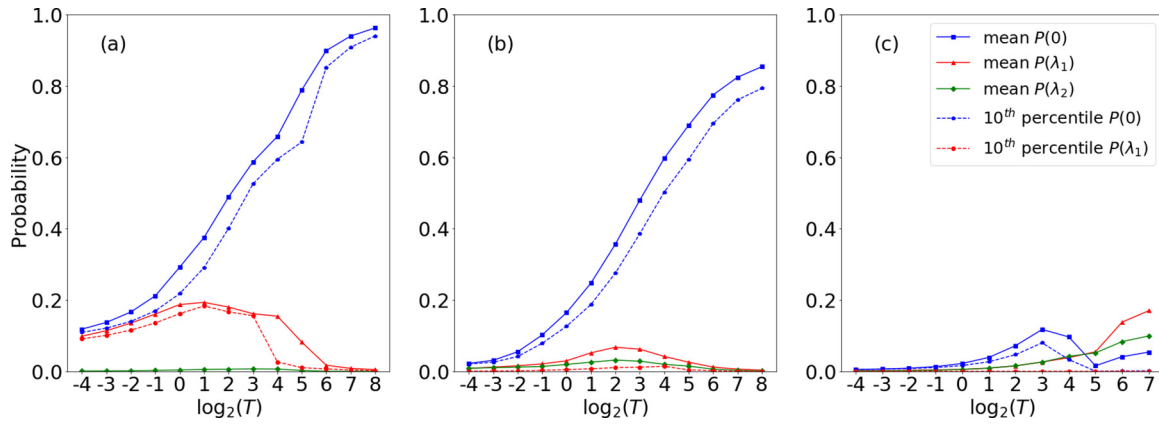
FIG. 5. For quantum SVP simulations on (a) 200 2D "bad" lattice bases, (b) 150 3D "bad" lattice bases, and (c) 150 4D "bad" lattice bases, the mean (solid) and 10th percentile (dashed) probabilities of returning the ith shortest vector are depicted. Blue is for ground state (zero vector), red corresponds to shortest vector [$\lambda_1(\mathcal{L})$], and green is for the second shortest vector [$\lambda_2(\mathcal{L})$].

In Fig. 5 this zone appears around $T = 2$ for 2D lattices and $T = 4$ for 3D lattices.

The case for 4D lattices looks quite different. The solid blue line is overtaken by probabilities for $\lambda_1(\mathcal{L})$ and even $\lambda_2(\mathcal{L})$. To understand better what is happening let us look at the probability distributions at a few different points from plot (c) in Fig. 5, which illustrates results on 4D lattices. To this end, Fig. 6 presents the same information as that in Fig. 4 but for more samples, and should be looked at closely in conjunction with plot (c) of Fig. 5. Again, probabilities corresponding to $\lambda_1(\mathcal{L})$ are highlighted in red. While it is apparent there is some locally different behavior corresponding to the ground state, Fig. 6 shows that, as sweeps become slower, probability density continues to accumulate around the

lowest energy states, and, if slow enough (though beyond our computational capabilities), would concentrate entirely on the lowest eigenstate. This behavior is particularly promising for the quantum SVP algorithm in higher dimensions as again one can see [subplot (f) provides the best example] a significant concentration of probability around the ten lowest eigenstates (out of $> 10\,000$) without this distribution necessarily being dominated by the zero vector. Furthermore, the probability of achieving $\lambda_1(\mathcal{L})$ is considerably high relative to surrounding eigenstates for slower sweeps, as demonstrated by the series of red bars in Fig. 6

In order to explain some of the favorable characteristics it serves to look at a specific instance for the evolution of the system.
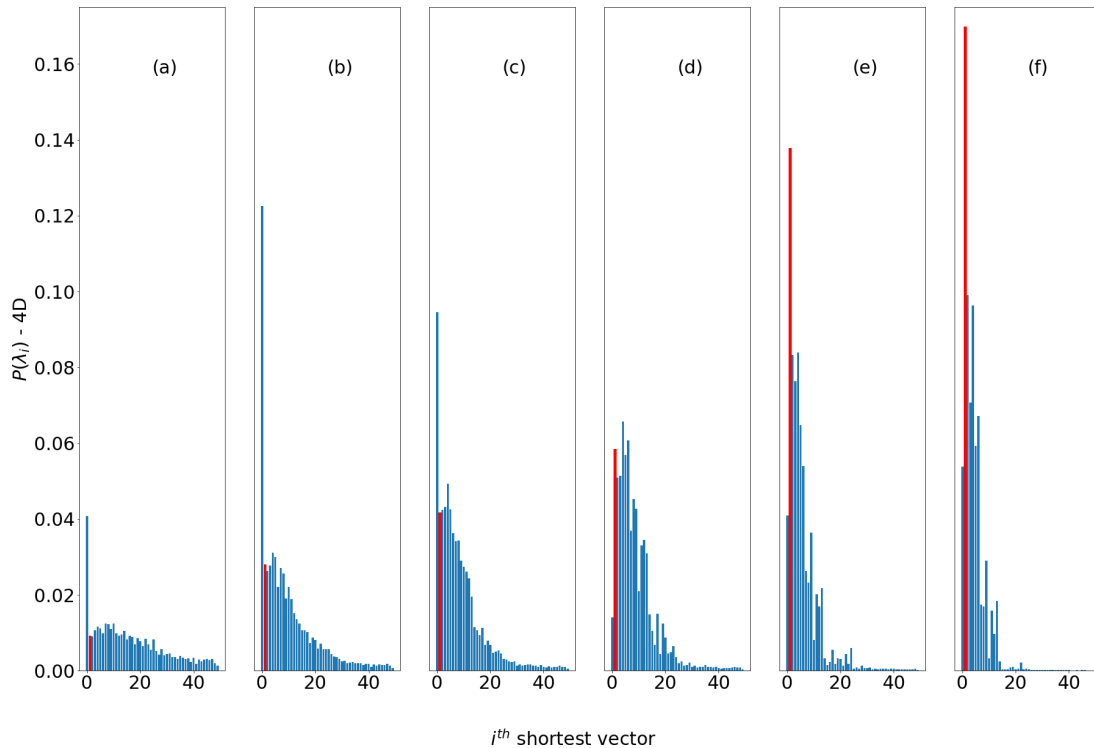

FIG. 6. Mean probability of returning ith shortest vector across 150 instances of four-dimensional lattices for sweep lengths $T = 4, 8, 16, 32, 64, 128$ in subplots (a), (b), (c), (d), (e), and (f), respectively.
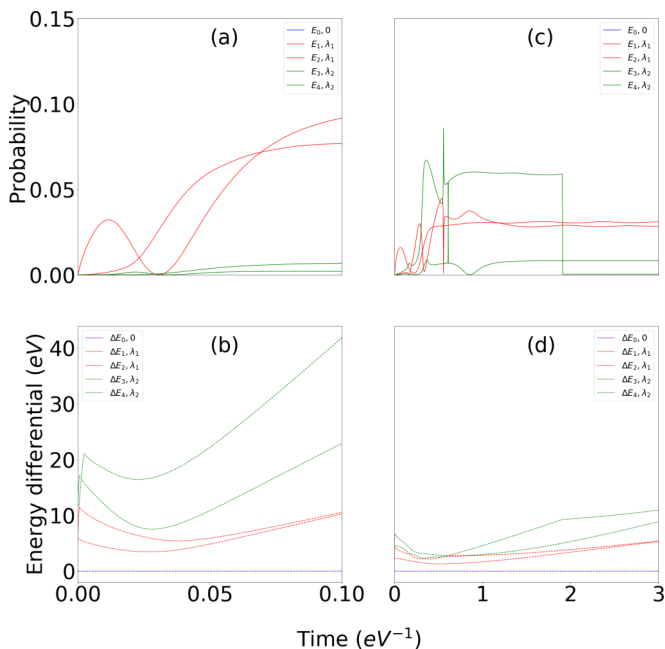
FIG. 7. Algorithm sweep on a 2D (left; 0.1 of $T = 0.25$) and 3D (right; 3 of $T = 4$) lattice basis showing the probability of measuring the system in an eigenstate (top) and the corresponding energy differential (bottom)—both for only the lowest five energy levels—defined as $\Delta E_i^t = E_i^t - E_0^t$. Degeneracies should not occur during the intermediate Hamiltonian, though H_P will contain degeneracies, for example those states corresponding to the two shortest vectors.

Figure 7 shows the energy levels and corresponding measurement probabilities for a typical 2D lattice example, run with parameters $T = 0.25$, $m = 3$. While the ground state solution relates to the unique zero vector, all other vectors have two associated eigenstates reflecting the symmetry of the lattice about zero (for every $\mathbf{v} \in \mathcal{L}$, its negative $-\mathbf{v}$ also belongs to $\mathcal{L}$). This is important as it means that while one excitation is ideal, two excitations still yield $\mathbf{x}_{\min}$ and hence the shortest vector.

A note on the trivial solution. Throughout this piece of work we have assumed the framework of adiabatic quantum computation. The presence of the (useless) zero vector in the solution set means we no longer seek the lowest energy eigenstate. This does not fit the traditional AQC framework, but is advantageous in that it permits faster sweeps. Scaling to larger systems, this could help to circumvent the prohibitive time cost of AQC algorithms. In subadiabatic regimes it is foreseeable that shorter sweep times could be employed at the cost of larger γ approximations for SVP_γ and vice versa.

V. DISCUSSION

We have introduced a quantum optimization framework to the area of computationally hard lattice problems that may underpin tomorrow's cryptosystems. By examining some interesting properties of an AQC-style algorithm when targeting low-but-not-lowest energy states, we have identified the existence of a "Goldilocks" zone for time sweep optimization. This is particularly exciting for cryptanalysis of lattice-based cryptosystems as the underlying problems often

come in approximate—and not exact—form, as with SVP_γ analyzed in this work. Among cryptographers it is thought that the approximate nature of lattice problems strengthens their post-quantum credentials, as the lack of determinism means quantum hidden subgroup algorithms cannot be applied. This "proximity" property, however, may allow subadiabatic algorithms for such problems to overcome the costly time requirements of AQC, while still outputting acceptable solutions. Outside of cryptography, it should be observed that for many real-world problems an approximate solution is fine where exact solutions are intractable, and the 'Goldilocks" zone highlighted in this paper indicates that this may be where AQC-style algorithms will most outperform classical alternatives.

The numerical analysis presented in Sec. IV offers an encouraging insight into how Hamiltonian simulation on higher dimensional instances may perform. The notion of mapping Euclidean distances into Hamiltonian energies is one that has many foreseeable applications in tackling lattice problems: there are similarities, for example, in the formulation of SVP_γ and the approximate closest vector problem [10]. Many lattice problems are closely related, meaning there are several areas one could apply the ideas laid out in this work.

Looking forward, there are many interesting challenges to surmount. A major one is the issue of achieving better theoretical bounds on scaling complexity. One advantage of AQC is that time dependence relies on only one factor (minimum energy gap Δ), meaning the source of time cost is easy to understand. In a subadiabatic regime, however, modeling eigenstate transitions probabilistically could be a natural progression for theoretical analysis of AQC-style algorithms. The development of quantum hardware that can realize this generalized Bose-Hubbard Hamiltonian and assume particle-particle offsite interactions is a target for experimental physicists, and generalizing AQC-style algorithms to run on different hardware—such as coherent Ising machines [34,35]—will become increasingly investigated as progress continues towards a post-quantum world.

APPENDIX A: QUBIT SCALING

With these heuristic scaling assumptions we can derive the following analysis for the single-run algorithm (results for multirun are similar). Using $m = cN$ for linear constant c, there are $m(N + 1)$ particles in the system. Therefore the total particle number for single run (denoted K_S) is $K_S = cN^2 + cN$.

The Hilbert space size with P particles distributed among Q sites is

$$D = \frac{(P + Q - 1)!}{(P)!(Q - 1)!}. \tag{A1}$$

The qubit scaling equivalent is obtained by simply taking the base-2 logarithm of the above expression, with values for P, Q substituted in:

$$D = \frac{(K_S + N)!}{K_S! N!} = \frac{(cN^2 + cN + N)!}{(cN^2 + cN)! N!}. \qquad (A2)$$

By Stirling's approximation this is very close to

$$\frac{1}{\sqrt{2\pi N}} \left(\frac{e}{N}\right)^N \left(\frac{(cN^2 + cN + N)!}{(cN^2 + cN)!}\right), \qquad (A3)$$

which can be written

$$\frac{1}{\sqrt{2\pi N}} \left(\frac{e}{N}\right)^N \prod_{i=1}^{N} (cN^2 + cN + i). \qquad (A4)$$

Bounding the product term, this is much less than

$$\frac{1}{\sqrt{2\pi N}} \left(\frac{e}{N}\right)^N [cN^2 + (c+1)N]^N$$

$$= \frac{1}{\sqrt{2\pi N}} [e(cN + (c+1)]^N, \qquad (A5)$$

leaving a system size in qubit terms of $\log_2 D$ bounded above by $O(N \log N)$. There are no analytical time bounds; this is an active research area in the community. What we can do is provide some analysis for a Grover search algorithm over the same solution space, giving us some post-quantum context for the complexity to be expected. Given that the solution space scales as $N^N = 2^{N \log N}$, search using Grover's algorithm scales as $2^{\frac{1}{2} N \log N}$.

APPENDIX B: BASIS BAND DIAGONALIZATION

In order to guarantee that offsite interaction terms γ_{ij} in the problem Hamiltonian are nonzero only for small $|i - j|$, the basis must be altered to take a banded structure, but using only operations that preserve the basis. To demonstrate, consider the following example: The row span of the matrix

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 0 & b_{22} & b_{23} \\ 0 & 0 & c_{33} \end{bmatrix}$$

returns vectors $\mathbf{v} = n_1 \mathbf{a} + n_2 \mathbf{b} + n_3 \mathbf{c}$ of norms

$$|\mathbf{v}| = (n_1 a_{11})^2 + (n_1 a_{12} + n_2 b_{22})^2 + (n_2 b_{23} + n_3 c_{33})^2. \qquad (B1)$$

Expanding this vector norm and grouping into onsite terms and offsite terms gives

$$|\mathbf{v}| = n_1^2 (a_{11}^2 + a_{12}^2) + n_2^2 (b_{22}^2 + b_{23}^2) + n_3^2 (c_{33}^2)$$

$$+ 2n_1 n_2 a_{12} b_{22} + 2n_2 n_3 b_{23} c_{33}, \qquad (B2)$$

and one can see that while there exist nearest-neighbor interaction terms (γ_{12}, γ_{23}) there is no γ_{13} term and so one can see how the banding structure is necessary to eliminate far away offsite particle-particle interactions.

Our solution is to iteratively eliminate elements far away from the leading diagonal using an argument that relies on taking the greatest common divisor (gcd) of as many elements as needed to help us eliminate elements using Bezout's lemma.

Take a simple example as a taste of what this algorithm is tapping into. Consider a prime determinant integer lattice,

here taking the determinant to be some large p. Let us pick p so that the lattice is in fact the row span of the following matrix. Every prime determinant integer lattice can be represented by a basis very similar in form to the one shown [36]; the pivot p is not necessarily in the bottom right, but this does not affect the reduction:

$$\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_N \end{bmatrix} = \begin{bmatrix} 1 & & & x_1 \\ & \ddots & & \vdots \\ & & 1 & x_{N-1} \\ & & & p \end{bmatrix}. \qquad (B3)$$

For this simple case assume that the gcd of x_{i+1}, x_{i+2} divides x_i. Then by Bezout's lemma there exist some u, v, δ such that

$$\delta(ux_{i+1} + vx_{i+2}) = x_i. \qquad (B4)$$

Now perform the lattice preserving row operations

$$\mathbf{b}_i \rightarrow \mathbf{b}_i^* = \mathbf{b}_i - \delta u \mathbf{b}_{i+1} - \delta v \mathbf{b}_{i+2}. \qquad (B5)$$

This should be performed iteratively from $i = 1$ to $N - 2$. After the first such iteration the above matrix looks as follows:

$$\begin{bmatrix} \mathbf{b}_1^* \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{b}_n \end{bmatrix} = \begin{bmatrix} 1 & -\delta u & -\delta v & & & 0 \\ & 1 & & & & x_2 \\ & & 1 & & & x_3 \\ & & & \ddots & & \vdots \\ & & & & 1 & x_{n-1} \\ & & & & & p \end{bmatrix}. \qquad (B6)$$

If $\gcd(x_{i+1}, x_{i+2})$ does not divide x_i, then extend consideration to x_{i+3}, and so on until finding a group of numbers with gcd dividing x_i. Each extra number means adding an extra band to the matrix so it is ideal to find a set of coprime (or with gcd dividing x_i) entries $x_{i+1}, \ldots, x_{i+j}$ for some small j. Fortunately, for k randomly selected numbers the probability of them being coprime (a stronger condition than is needed) is $1/\zeta(k)$ which fast approaches 1 as k increases. This means that even for high dimensional lattices one can be confident of tight bandings.

There are circumstances where x_i is odd, and several of the entries below it are even. These are the only problematic cases where sticking strictly to the algorithm yields poor results. In these cases it is optimal to instead multiply $\mathbf{b}_i$ by 2 so that x_i can be eliminated efficiently. This is not ideal as it does not preserve the lattice, but instead increases the volume by a factor of 2 (or p if this is extended to some other small primes). What results is a basis for a sublattice. Fortunately these cases are rare enough that the mean volume increase on performing this algorithm over many lattices is very small.

To generalize this to any given HNF basis, this procedure simply needs to be repeated for all dense columns, which will appear only above pivots that are not equal to 1. While technically HNF bases can be dense in the upper triangle, this is not typical. Furthermore, HNF bases are particularly relevant cryptographically [37] as they afford a way of representing a bad basis that can be communicated with $O(N)$ key size [if dense this would be $O(N^2)$]. Thus the property that makes this basis a good candidate for band-diagonalization also makes it a good choice in terms of cryptographic efficiency. Figure 8 shows the average results for the tight banding algorithm on
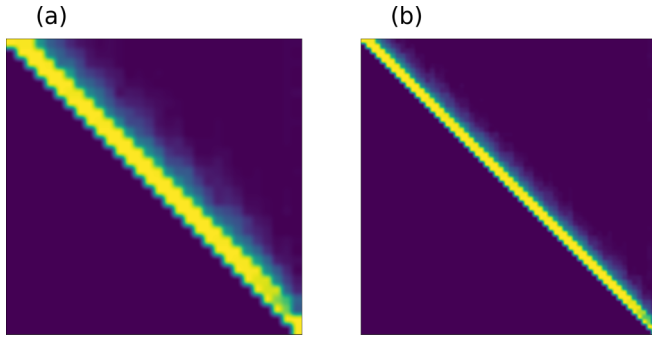
FIG. 8. Banding algorithm performance over 100 instances of 30-dimensional lattices (a) and 60-dimensional lattices (b). The color of each coordinate represents the average relative entry size.
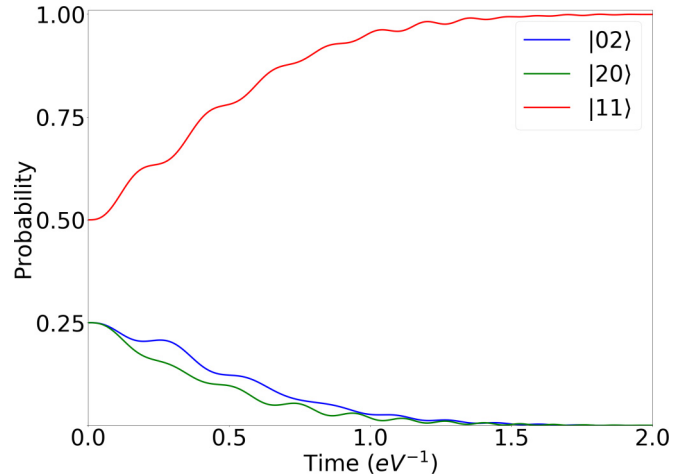

FIG. 9. Each line represents the probability of finding the system in the corresponding Fock state as outlined in Eq. (C1) as time progresses in the evolution.

HNF bases for 100 lattices in 30 and 60 dimensions, with the heat indicating the relative size of coefficients. The size of these coefficients reduces farther away from the leading diagonal. The dark blue squares represent zeros and so it is clear to see both the upper-triangular form of the row bases and also how the magnitude of the coordinates fades quickly to zeros above the leading diagonal. Moreover, increasing the dimension does not adversely affect the ability of the algorithm to produce a tightly banded lattice basis. The mean volume increase in 30 dimensions was 2.98 and in 60 dimensions it was 7.99, meaning that while the algorithm tends not to preserve the lattice exactly, the volume of the basis is increased by a small factor. This is acceptable for solving SVP_γ. The effect of this band-diagonalization algorithm is that in realizing the quantum SVP algorithms described in this paper it is not necessary to consider particle-particle interaction terms for particles at sites which are far away from each other.

APPENDIX C: EXAMPLE RUN

Consider a very simple example to aid intuition in following the algorithm from lattice basis to final result and look at all the steps in between. The system comprises two particles in two lattice sites, with no offset ($m = 0$). The Hilbert space is three dimensional and the Fock states are

$$|02\rangle, |11\rangle, |20\rangle. \tag{C1}$$

Take a good basis $\mathbf{B}_G$, and a slightly longer bad basis $\mathbf{B}_B$. Derive from $\mathbf{B}_B$ the Gram matrix $\mathbf{G}_B$ which defines coefficients for the problem Hamiltonian H_P:

$$\mathbf{B}_G = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad \mathbf{B}_B = \begin{pmatrix} 1 & 2 \\ 0 & -2 \end{pmatrix}, \quad \mathbf{G}_B = \begin{pmatrix} 5 & -4 \\ -4 & 4 \end{pmatrix}. \tag{C2}$$

Calculating coefficients for H_0 from Eq. (5) and H_P from Eq. (13) gives initial and final Hamiltonians of

$$H_0 = \begin{pmatrix} 0 & -\sqrt{2} & 0 \\ -\sqrt{2} & 0 & -\sqrt{2} \\ 0 & -\sqrt{2} & 0 \end{pmatrix}, \quad H_P = \begin{pmatrix} 16 & 0 & 0 \\ 0 & -8 & 0 \\ 0 & 0 & 20 \end{pmatrix}, \tag{C3}$$

and the time-dependent Hamiltonian matrix, using linear time evolution as per Eq. (8), is therefore

$$H(t) = \begin{pmatrix} 16\frac{t}{T} & -\sqrt{2}\left(1 - \frac{t}{T}\right) & 0 \\ -\sqrt{2}\left(1 - \frac{t}{T}\right) & -8\frac{t}{T} & -\sqrt{2}\left(1 - \frac{t}{T}\right) \\ 0 & -\sqrt{2}\left(1 - \frac{t}{T}\right) & 20\frac{t}{T} \end{pmatrix}. \tag{C4}$$

The Hamiltonian ground state for $t = 0$ is

$$|\psi_0\rangle = \tfrac{1}{2}(|02\rangle + \sqrt{2}\,|11\rangle + |20\rangle). \tag{C5}$$

Initialize the system in this state and let it evolve. The probabilities of measuring the system to be in each of the Fock states during the evolution are shown in Fig. 9. The Hamiltonian ground state for $t = T = 2$ is

$$|\psi_T\rangle = |11\rangle, \tag{C6}$$

which is easy to see because

$$(1 \quad 1)\begin{pmatrix} 1 & 2 \\ 0 & -2 \end{pmatrix} = (1 \quad 0), \tag{C7}$$

and $(1, 0)$ is the shortest vector in the lattice. This can be seen by looking at $\mathbf{B}_G$. This example would be one of many runs, each with a different number of particles, that together would constitute a multirun algorithm to solve SVP for the lattice described by $\mathbf{B}_B$ (and also $\mathbf{B}_G$, though one would not have any prior knowledge of $\mathbf{B}_G$).

[1] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, arXiv:quant-ph/0001106.

[2] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, Science **292**, 472 (2001).

[3] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, SIAM J. Comput. **37**, 166 (2004).

[4] P. W. Shor, SIAM Rev. **41**, 303 (1999).

[5] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, Phys. Rev. X **8**, 011044 (2018).

[6] A. Aspuru-Guzik, R. Lindh, and M. Reiher, ACS Central Sci. **4**, 144 (2018).

[7] R. L. Rivest, A. Shamir, and L. Adleman, Commun. ACM **21**, 120 (1978).

[8] W. Diffie and M. Hellman, IEEE Trans. Inf. Theory **22**, 644 (1976).

[9] D. Bernstein, J. Buchmann, and E. Dahmen, *Introduction to Post-Quantum Cryptography* (Springer-Verlag, Berlin, 2009), pp. 1–13.

[10] C. Peikert, Found. Trends Theor. Comput. Sci. **10**, 283 (2016).

[11] T. Laarhoven, M. Mosca, and J. Van De Pol, in *Post-Quantum Cryptography*, 5th International Workshop, PQCrypto 2013, Limoges, France, June 4–7, 2013, edited by P. Gaborit, Lecture Notes in Computer Science Vol. 7932 (Springer-Verlag, Berlin, 2013), pp. 83–101.

[12] O. Regev, SIAM J. Comput. **33**, 738 (2004).

[13] A. Montanaro, Theory Comput. **14**, 1 (2018).

[14] Y. Aono, P. Q. Nguyen, and Y. Shen, in *Advances in Cryptology – ASIACRYPT 2018*, 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, edited by T. Peyrin and S. Galbraith (Springer-Verlag, Berlin, 2018), p. 405.

[15] L. Ducas, Nieuw Arch. Wisk. **5**, 184 (2017).

[16] R. Cramer, L. Ducas, and B. Wesolowski, in *Advances in Cryptology – EUROCRYPT 2017*, 6th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 – May 4, 2017, edited by J.-S. Coron and J. B. Nielsen, Lecture Notes in Computer Science Vol. 10210 (Springer-Verlag, Berlin, 2017), p. 324.

[17] O. Regev, in *STOC '05: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, NY, US, 2005).

[18] D. Micciancio and C. Peikert, in *Advances in Cryptology – EUROCRYPT 2012*, 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15–19, 2012, edited by D. Pointcheval and T. Johansson, Lecture Notes in Computer Science Vol. 7237 (Springer-Verlag, Berlin, 2012), pp. 700–718.

[19] D. Micciancio and O. Regev, SIAM J. Comput. **37**, 267 (2007).

[20] J. Hoffstein, J. Pipher, and J. H. Silverman, in *Algorithmic Number Theory*, Third International Symposiun, ANTS-III, Portland, Oregon, USA, June 21–25, 1998, edited by J. P. Buhler, Lecture Notes in Computer Science Vol. 1423 (Springer-Verlag, Berlin, 1998), p. 267.

[21] O. Goldreich, S. Goldwasser, and S. Halevi, in *Annual International Cryptology Conference*, edited by B. S. Kaliski Jr. (Springer, Berlin, Heidelberg, 1997), pp. 112–131.

[22] A. K. Lenstra, H. W. Lenstra, and L. Lovász, Math. Ann. **261**, 515 (1982).

[23] H. A. Gersch and G. C. Knollman, Phys. Rev. **129**, 959 (1963).

[24] I. Bloch, Nat. Phys. **1**, 23 (2005).

[25] D. Jaksch, C. Bruder, J. I. Cirac, C. W. Gardiner, and P. Zoller, Phys. Rev. Lett. **81**, 3108 (1998).

[26] M. Greiner, O. Mandel, T. Esslinger, T. W. Haensch, and I. Bloch, Nature (London) **415**, 39 (2002).

[27] M. Born and V. Fock, Z. Phys. **51**, 165 (1928).

[28] D. Raventós, T. Graß, M. Lewenstein, and B. Juliá-Díaz, J. Phys. B **50**, 113001 (2017).

[29] S. Jansen, M. B. Ruskai, and R. Seiler, J. Math. Phys. **48**, 102111 (2007).

[30] L. K. Grover, in *STOC '96: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, NY, US, 1996).

[31] A. Lucas, Front. Phys. **2**, 5 (2014).

[32] N. Gama, P. Q. Nguyen, and O. Regev, in *Advances in Cryptology – EUROCRYPT 2010*, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 – June 3, 2010, edited by H. Gilbert, Lecture Notes in Computer Science Vol. 6110 (Springer-Verlag, Berlin, 2010), p. 257.

[33] E. Crosson, E. Farhi, C. Y.-Y. Lin, H.-H. Lin, and P. Shor, arXiv:1401.7320.

[34] R. Hamerly, Y. Haribara, M. M. Fejer, C. Langrock, T. Inagaki, K. Aihara, H. Mabuchi, H. Takesue, A. Marandi, P. L. McMahon, Y. Yamamoto, S. Utsunomiya, S. Tamate, and R. L. Byer, Science **354**, 614 (2016).

[35] R. Hamerly, T. Inagaki, P. L. McMahon, D. Venturelli, A. Marandi, T. Onodera, E. Ng, C. Langrock, K. Inaba, T. Honjo, K. Enbutsu, T. Umeki, R. Kasahara, S. Utsunomiya, S. Kako, K.-i. Kawarabayashi, R. L. Byer, M. M. Fejer, H. Mabuchi, D. Englund, E. Rieffel, H. Takesue, and Y. Yamamoto, Sci. Adv. **5**, eaau0823 (2019).

[36] D. Goldstein and A. Mayer, Forum Math. **15**, 165 (2003).

[37] D. Micciancio, in *Cryptography and Lattices*, International Conference, CaLC 2001, Providence, RI, USA, March 29–30, 2001, edited by J. H. Silverman, Lecture Notes in Computer Science Vol. 2146 (Springer-Verlag, Berlin, 2001), pp. 126–145.