entropy

IMPACT FACTOR 2.738

Indexed in: PubMed

Article

Using Variational Quantum Algorithm to Solve the LWE Problem

Lihui Lv, Bao Yan, Hong Wang, Zhi Ma, Yangyang Fei, Xiangdong Meng and Qianheng Duan

Special Issue
Advances in Quantum Computing

Edited by
Dr. Brian R. La Cour and Dr. Giuliano Benenti

MDPI

https://doi.org/10.3390/e24101428

entropy

MDPI

Article

Using Variational Quantum Algorithm to Solve the LWE Problem

Lihui Lv [1,2], Bao Yan [1,2], Hong Wang [1,2,*], Zhi Ma [1,2,*], Yangyang Fei [1,2], Xiangdong Meng [1,2] and Qianheng Duan [1,2]

[1] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
[2] Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China
* Correspondence: redwang@meac-skl.cn (H.W.); ma.zhi@meac-skl.cn (Z.M.)

Abstract: The variational quantum algorithm (VQA) is a hybrid classical–quantum algorithm. It can actually run in an intermediate-scale quantum device where the number of available qubits is too limited to perform quantum error correction, so it is one of the most promising quantum algorithms in the noisy intermediate-scale quantum era. In this paper, two ideas for solving the learning with errors problem (LWE) using VQA are proposed. First, after reducing the LWE problem into the bounded distance decoding problem, the quantum approximation optimization algorithm (QAOA) is introduced to improve classical methods. Second, after the LWE problem is reduced into the unique shortest vector problem, the variational quantum eigensolver (VQE) is used to solve it, and the number of qubits required is calculated in detail. Small-scale experiments are carried out for the two LWE variational quantum algorithms, and the experiments show that VQA improves the quality of the classical solutions.

Keywords: quantum; LWE; QAOA; VQE; KYBER

Citation: Lv, L.; Yan, B.; Wang, H.; Ma, Z.; Fei, Y.; Meng, X.; Duan, Q. Using Variational Quantum Algorithm to Solve the LWE Problem. *Entropy* **2022**, *24*, 1428. https://doi.org/10.3390/e24101428

Academic Editors: Brian R. La Cour and Giuliano Benenti

Received: 30 August 2022
Accepted: 2 October 2022
Published: 8 October 2022

1. Introduction

Lattice theory is a classic subject in mathematical research, and it has critical applications in many fields such as the optimization problem and information coding. In 1996, Ajtai [1] proved that the worst-case hardness of the shortest vector problem (SVP) can be reduced to the hardness of SVP in a class of random lattices, thus providing provable security of lattice-based cryptosystems. Since then, various lattice-based cryptosystems are proposed, such as Ajtai-Dwork [2] and the Number Theory Research Unit [3].

In 2005, Regev proposed an encryption algorithm based on LWE [4]. Compared with previous lattice-based cryptosystems, the ciphertext size and key size of LWE-based cryptosystems are greatly reduced. Therefore, LWE began to be applied to many cryptographic primitives, such as Key-Dependent Message [5], Fully Homomorphic Encryption [6] and so forth. In July 2022, The National Institute of Standards and Technology completed the third round of the Post-Quantum Cryptography standardization process, and four candidate algorithms have been announced. Among them, the public-key encryption algorithm CRYSTALS-KYBER [7] and the digital signature algorithm CRYSTALS-Dilithium [8] are constructed based on the module-LWE problem. Therefore, analyzing LWE algorithms is important to the security of post-quantum cryptography.

The analysis methods of LWE can be classified into combinatorial methods, algebraic methods, lattice methods and the exhaustive search. The combinatorial method mainly refers to an extended application of the Gaussian elimination [9], but it requires a large number of samples. The algebraic method refers to the Arora-Ge algorithm [10], and the complexity is also exponential in the number of LWE dimensions. There are three main lattice methods: the dual method is used to attack decision-LWE instances by solving the short integer solution problem on the dual lattice [1]; the decoding method is used to directly solve the bounded distance decoding problem (BDD) on the original lattice [11,12];

the primary method is used to further reduce the BDD problem to the Unique-SVP problem [13–15]. The exhaustive search is not suitable for practical applications because of its high time complexity.

At the same time, VQA, such as QAOA [16], VQE [17], and FQE [18], has become the most suitable technology to achieve quantum advantage using noisy intermediate-scale quantum (NISQ) devices. Some works have studied how to solve hard lattice problems by VQA. Paper [19] analyzed the energy gaps between the first three excited states of the Hamiltonian when solving SVP with low dimension by quantum adiabatic computation. The conclusion in [19] inspired the use of QAOA to find the ground state. Ref. [20] calculated the number of qubits for special lattices and concluded that $1.5nlogn + n + log(det(\mathcal{L}))$ qubits sufficed to obtain the shortest vector of n-dimensional lattice $\mathcal{L}$. Ref. [21] proposed to solve SVP by VQE and also pointed out that their algorithm was not limited to special lattices.

The work in this paper consists of two aspects. Firstly, we use QAOA to optimize the Nearest Plane algorithm and solve LWE. Secondly, inspired by Ref. [21], we propose a hybrid algorithm using VQE to attack LWE and calculate the number of qubits required to attack specific LWE cryptosystems. For the two LWE algorithm ideas, we conduct small-scale experimental simulations. The experiments show that QAOA improves the quality of classical solutions, and the quality of solutions obtained by VQE is at least equal to that of classical solutions when the memory is big enough.

2. Preliminary

2.1. Lattice Theory

Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ be a set of linearly independent vectors, and the lattice generated by $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ is

$$\Lambda = \mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \{\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2 + \ldots + \alpha_n\mathbf{b}_n | \alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{Z}\}.$$

In cryptography applications, the lattice dimension is n. Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{m*n}$, the q-ary lattice refers to

$$\Lambda_q(\mathbf{A}^T) = \{\mathbf{x} \in \mathbb{Z}^m | \exists \mathbf{y} \in \mathbb{Z}^n, s.t. \mathbf{x} \equiv \mathbf{y}\mathbf{A}^T modq\}.$$

For a lattice $\mathcal{L}$ and its basis matrix $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$, the volume of the lattice is $vol(\mathcal{L}) = \sqrt{det(\mathbf{B}^T\mathbf{B})}$ and the fundamental domain is $\mathcal{P}_{1/2}(\mathbf{B}) = \{\sum_{i=1}^n \alpha_i\mathbf{b}_i | \alpha_i \in [-\frac{1}{2}, \frac{1}{2}]\}$. The distance between $\mathcal{L}$ and vector $\mathbf{v} \in \mathbb{R}^m$ is $dist(\mathbf{v}, \mathcal{L}) = min\{\|\mathbf{v} - \mathbf{y}\| | \mathbf{y} \in \mathcal{L}\}$. The i-th successive minima $\lambda_i(\mathcal{L})$ is the minimum radius of the ball centered at the origin, which contains i linearly independent vectors in the lattice. Let $\mathcal{L}$ be an n-dimensional lattice; then, the Gaussian heuristic states that $\lambda_1(\mathcal{L}) \approx \sqrt{\frac{n}{2\pi e}} vol(\mathcal{L})^{1/n}$.

Definition 1. *(Shortest vector problem, SVP) For a lattice $\mathcal{L}$, the SVP problem asks to find a nonzero lattice vector $\mathbf{v}$ that minimizes the Euclidean nonzero norm $\|\mathbf{v}\|$.*

Definition 2. *(Closest vector problem, CVP) For a lattice $\mathcal{L}$, given a target vector $\mathbf{t} \in \mathbb{R}^m$ that is not in $\mathcal{L}$, the CVP problem asks to find a lattice vector $\mathbf{v}$ that minimizes the Euclidean norm $\|\mathbf{v} - \mathbf{t}\|$.*

Definition 3. *(Unique shortest vector problem, Unique-SVP) For a lattice $\mathcal{L}$ satisfying $\lambda_2(\mathcal{L}) > \gamma\lambda_1(\mathcal{L})$, where $\gamma \gg 1$, the uSVP problem asks to find the shortest nonzero lattice vector.*

Definition 4. *(Bounded distance decoding, BDD) For a target vector $\mathbf{t} \in \mathbb{R}^m$ that is not in the given lattice $\mathcal{L}$, which satisfies $dist(\mathbf{t}, \mathcal{L}) < \gamma\lambda_1(\mathcal{L})$, where $\gamma < 1/2$, the BDD problem asks to find a nonzero lattice vector $\mathbf{v}$ that minimizes the Euclidean norm $\|\mathbf{v} - \mathbf{t}\|$.*

Algorithms for hard problems on lattices usually perform lattice basis reduction as a preprocessing module, because a sufficiently good basis improves the algorithms' success

probability. The LLL (Lenstra–Lenstra–Lovász) algorithm [22] and the BKZ (block–Korkin–Zolotarev) algorithm [23] are two famous basis reduction algorithms.

Before introducing the LLL reduction algorithm, we first explain the Gram–Schmidt orthogonalization. With a lattice basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$, one can calculate its Gram–Schmidt orthogonalization $\mathbf{B}^* = [\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*]$ by the recursion $\mathbf{b}_1^* = \mathbf{b}_1, \mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*$ for $i = 2, 3, \ldots, n$, where the Gram–Schmidt coefficients $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle$. The LLL algorithm was proposed in 1982, and the formal description of LLL reduction is detailed as shown in Algorithm 1.

Algorithm 1 LLL algorithm.

Input: lattice basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n] \in \mathbb{R}^{m \times n}$, a reduction parameter δ.
Output: a δ-LLL reduced basis

Calculate the Gram–Schmidt orthogonalization $\mathbf{B}^* = [\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*]$.
for i = 2, 3,..., n **do**
  for j = i − 1, i − 2,..., 1 **do**
    $\mathbf{b}_i = \mathbf{b}_i - c_{i,j}\mathbf{b}_j$, where $c_{i,j} = \lceil \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle \rfloor$;
  end for
end for
if $\exists i$, s.t.$\delta \|\mathbf{b}_{i-1}^*\|^2 > \|\mu_{i,i-1}\mathbf{b}_{i-1}^* + \mathbf{b}_i^*\|^2$ **then**
  Swap $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$;
  Go to Step 1.
end if
return B.

The BKZ algorithm is derived from the KZ (Korkine–Zolotarev) reduction. BKZ uses the block reduction to improve the LLL algorithm and outputs an (δ, β)-BKZ reduced basis. To be specific, the BKZ algorithm runs the enumeration algorithm on the sub-lattice with block size β and obtains its shortest vector. After inserting the shortest vector into the original basis, LLL reduction with parameter δ is applied on the entire basis to remove the linear dependency. BKZ performs the above steps iteratively until the basis is no longer updated.

2.2. The LWE Problem

Definition 5. *(Learning with errors distribution) Let $n, q > 0$ be integers, and $\alpha \in \{0, 1\}$. Let $\mathbf{s} \in \mathbb{Z}_q^n$ be a secret vector. The LWE distribution $\chi_{\mathbf{s},\alpha}$ refers to $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\mathbf{a} \in \mathbb{Z}_q^n$ is uniformly selected randomly and e is a discrete Gaussian error with standard deviation αq.*

Definition 6. *(Learning with errors problem) Let $n, m, q > 0$ be integers, $\alpha > 0$. Given m samples $(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i), i = 1, 2, \ldots, m$, the search-LWE problem asks to recover the secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, and the decision-LWE problem asks to determine whether the samples are sampled according to $\chi_{\mathbf{s},\alpha}$ or the uniform distribution.*

Now, we review some lattice-based methods for analyzing the LWE problem. In general, the decision-LWE can be solved by the short integer solution strategy, and the search-LWE can be attacked by the BDD strategy or the inhomogeneous short integer solution strategy. Now, we mainly describe the decoding method and the primal method in the BDD strategy.

The LWE problem can be written in a matrix form $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e} mod q$. Given $q \in \mathbb{Z}$, $\mathbf{c} \in \mathbb{Z}_q^m$, $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]^T \in \mathbb{Z}_q^{m \times n}$, the problem recovers $\mathbf{s}$. The basic idea of the decoding method is to regard $\mathbf{c}$ as the target vector and then use the Nearest Plane algorithm to find the closest vector in $\Lambda_q(\mathbf{A})$. Assuming the basis of $\Lambda_q(\mathbf{A})$ is $\mathbf{B}$, before applying the Babai's Nearest Plane algorithm, $\mathbf{B}$ should be preprocessed to a Gram–Schmidt basis $\mathbf{B}^*$. The strategy outputs $\mathbf{s}$ if and only if $\mathbf{e}$ lies in $\mathbf{s} + \mathbb{P}_{1/2}(\mathbf{B}^*)$, which is determined by the quality of the basis. Lindner and Peikert improved Babai's algorithm by admitting a time/success

trade-off. To be specific, in each iteration, the Lindner–Peikert Nearest Plane algorithm chooses several close hyperplanes instead of only the closest hyperplane. The idea stretches $\mathbb{P}_{1/2}(\mathbf{B}^*)$ to a cube-like shape and amplifies the success probability.

The primal method is to solve LWE by reducing BDD to the Unique-SVP problem using an embedding technique. The embedding method is to construct a $(m+1)$-dimensional lattice $\mathbf{B}' = \begin{bmatrix} \mathbf{B} & \mathbf{c} \\ \mathbf{0} & t \end{bmatrix}$. Obviously, the short vector $[-\mathbf{e}, t] \in \mathbb{Z}_q^{m+1}$ is in $\mathbf{B}'$. Therefore, solving the Unique-SVP instance recovers the error vector and the secret vector in passing.

2.3. Variational Quantum Algorithm

VQA is a quantum–classical hybrid algorithm that is considered to be implemented on NISQ devices. Therefore, VQA is expected to demonstrate quantum advantages over classical computers when solving some specific problems. The workflow of VQA is shown in Algorithm 2.

Algorithm 2 VQA algorithm.

Input: An optimization problem.
Output: Parameters in the parameterized quantum circuit.
1: Construct the objective function.
2: Construct the parameterized quantum circuit.
3: Prepare the quantum state and measure the expectation value.
4: Use a classical optimizer to determine new parameters.
5: Iterate the procedure in step 3 and 4 until the convergence of the value.
6: **return** the final parameters.

There are four important modules in VQA [24,25]: the objective function refers to the cost function that needs to be minimized; the parameterized quantum circuit refers to a set of unitary operators that manipulate parameters in the optimization process; the measurement scheme calculates the expectation value; the classical optimizer outputs the parameters that minimize the objective function.

First, VQA encodes the problem into an objective function O. Let the probability of measuring qubit q in state $|0\rangle$ be p^q; then, the objective function of VQA can be expressed as $min_{\boldsymbol{\theta}} O(\boldsymbol{\theta}, \{p(\boldsymbol{\theta})\})$.

Because it is inconvenient to obtain the function value directly by the measurement probability, the expectation value of a Hamiltonian is introduced, and constructing the objective function is equivalent to constructing its corresponding Hamiltonian. The Hamiltonian is a quantum operator that encodes the information of a physical system. Its expectation value corresponds to the energy of a quantum state. The ground state of the Hamiltonian is often used as the minimization target of a VQA problem. In practice, the expectation value of Hamiltonian H

$$\langle H \rangle_{U(\boldsymbol{\theta})} = \langle 0 | U^\dagger(\boldsymbol{\theta}) H U(\boldsymbol{\theta}) | 0 \rangle$$

is used to describe the measurement results of the quantum state produced by $U(\boldsymbol{\theta})$. Therefore, the objective function is

$$min_{\boldsymbol{\theta}} O(\boldsymbol{\theta}, \langle H \rangle_{U(\boldsymbol{\theta})}).$$

If the objective function is defined more compactly, it can be described as $min_{\boldsymbol{\theta}} \langle H \rangle_{U(\boldsymbol{\theta})}$. The objective functions or cost functions constructed in this paper are all in the compact form.

Second, parameterized quantum circuits are a set of unitary operations that depend on parameters. The parameterized quantum circuit acting on quantum state $|\psi_0\rangle$ can be expressed as

$$|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta}) |\psi_0\rangle,$$

where $\boldsymbol{\theta}$ are variational parameters.

Most ansatz U can be classified as problem-inspired or hardware-efficient. The construction of problem-inspired ansatz requires the information of specific problems. For example, the united coupled cluster ansatz in quantum chemistry is constructed by a parameterized cluster operator $T(\boldsymbol{\theta})$ and acts on the ground state $|\psi_{HF}\rangle$ in the way of $|\psi(\boldsymbol{\theta})\rangle = e^{T(\boldsymbol{\theta})-T^{\dagger}(\boldsymbol{\theta})}|\psi_{HF}\rangle$. Ansatz in the QAOA algorithm is also problem-inspired, and its construction is shown in Section 3. Hardware-efficient ansatz is usually expressed as $\prod_{k=1}^{D} U_k(\boldsymbol{\theta}_k)W_k$, where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_D)$, $U_k(\boldsymbol{\theta}_k) = e^{-i\boldsymbol{\theta}_k V_k}$ is a unitary operator derived from Hamiltonian V_k, and W_k is an unparametrized unitary operator.

Third, in order to obtain the information of quantum state, we need to measure it in the computational basis and calculate the expectation value of the objective function. The expectation value of the operator σ^z can be obtained by $\langle\sigma^z\rangle = \langle\psi|\sigma^z|\psi\rangle = |\alpha|^2 - |\beta|^2$, where $|\alpha|^2$ and $|\beta|^2$ are the probabilities to measure $|\psi\rangle$ in state $|0\rangle$ and $|1\rangle$. The measurement defined by σ^x and σ^y is first transformed into the basis of σ^z by $\sigma^x = R_y^{\dagger}(\pi/2)\sigma^z R_y(\pi/2)$, $\sigma^y = R_x^{\dagger}(\pi/2)\sigma^z R_x(\pi/2)$ and then measured on a σ^z basis. Any Pauli string is measured in the same way, except that it is measured on each qubit separately.

QAOA and VQE are two quantum variational algorithms, so they can be used to solve optimization problems. Since a quantum circuit is equivalent to a tensor product, it can be represented on a classical computer, and the expectation value of the cost function can be calculated, but the memory it consumes grows exponentially with the size of the problem. For a quantum computer, repeating the preparation of ansatz state and the quantum measurements, the expectation can be obtained. The quantum resources it consumes increase polynomially with the scale of the problem, thus showing its superiority over classical algorithms.

3. The Decoding Method for Solving LWE

This section applies the decoding method to solve LWE. When solving BDD, we use QAOA to improve Babai's Nearest Plane algorithm.

First, construct a q-ary lattice $\Lambda_q(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}_q^m | \exists \mathbf{x} \in \mathbb{Z}^n, s.t. \mathbf{v} \equiv \mathbf{A}\mathbf{x} mod q\}$, whose lattice basis is equivalent to $\mathbf{B} = [\mathbf{A}|q\mathbf{I}_m]^T \in \mathbb{Z}^{(m+n)\times m}$. Second, perform elementary row transformations on $\mathbf{B}$ and obtain a basis matrix $[\mathbf{b}'_1, \ldots, \mathbf{b}'_m]^T \in \mathbb{Z}^{m\times m}$. Third, solve CVP with the target vector $\mathbf{c}$, and finally output the closest vector $\mathbf{w}$. The last step is to use the Gaussian elimination to recover $\mathbf{s} = \mathbf{A}^{-1}\mathbf{w}$.

Now, introduce the application of QAOA when improving Babai's Nearest Plane algorithm. Babai's Nearest Plane algorithm consists of two steps: first, perform the LLL reduction on the input lattice basis, and then find the linear combination in the reduced basis so that it forms the closest lattice vector to the given target vector. The formal description is detailed as Algorithm 3.

In the loop, $u_j = \lceil \langle \mathbf{b}, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle \rfloor$ only takes one value by the "round to the nearest integer" function. Through experiments, it is found that when the value range is expanded to $\{u_j + x | x = 0, 1, -1\}$, a better solution is often obtained. In a classical algorithm, the process requires an exponential increase in computation with respect to the lattice dimension n. In quantum computing, due to quantum properties, the computing complexity can be greatly reduced. Therefore, we now introduce the method of encoding the random floating in u_j in two qubits and solving the optimization problem by QAOA.

Algorithm 3 Babai's Nearest Plane algorithm.

Input: lattice basis $\mathbf{B}' = [\mathbf{b}'_1, \mathbf{b}'_2, \ldots, \mathbf{b}'_m] \in \mathbb{R}^{m \times m}$, target vector $\mathbf{t} \in \mathbb{Z}^m$
Output: vector $\mathbf{x} \in \mathcal{L}(\mathbf{B}')$, which satisfies $\|\mathbf{x} - \mathbf{t}\| \leq 2^{m/2} dist(\mathbf{t}, \mathcal{L}(\mathbf{B}'))$
1: Perform the LLL reduction on $\mathbf{B}'$ with parameter $\delta = 3/4$.
2: Use the Gram–Schmidt orthogonalization on the reduced basis and obtain $\mathbf{B}^* = [\mathbf{b}^*_1, \mathbf{b}^*_2, \ldots, \mathbf{b}^*_m]$.
3: $\mathbf{b} = \mathbf{t}$.
4: **for** j = m, m − 1,..., 1 **do**
5: $\quad \mathbf{b} = \mathbf{b} - u_j \mathbf{b}'_j$, where $u_j = \lceil \langle \mathbf{b}, \mathbf{b}^*_j \rangle / \langle \mathbf{b}^*_j, \mathbf{b}^*_j \rangle \rfloor$;
6: **end for**
7: **return** $\mathbf{t} - \mathbf{b}$

First, apply Babai's Nearest Plane algorithm to calculate the classical optimal solution, that is, the shortest distance vector $\mathbf{b}_{op} = (b^1_{op}, b^2_{op}, \ldots, b^m_{op})$. Then, the result is improved by QAOA. Let the LLL-reduced basis in Babai's algorithm be $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m]$, and construct the optimization function

$$F(x_1, x_2, \ldots, x_m) = \| \sum_{i=1}^{m} x_i \mathbf{d}_i - \mathbf{b}_{op} \|^2,$$

where $x_i \in \{-1, 0, 1\}, i = 1, 2, \ldots, m$. It is easy to verify that $F(x_1, x_2, \ldots, x_m)$ is a non-negative function. Let $\hat{x}_i = \frac{\sigma^z_{2i-1} + \sigma^z_{2i}}{2}$, which is a quantum operator encoded in the Pauli-Z basis. The eigenvalues of operator $\hat{x}_i$ are -1, 0, 1, which exactly encodes the value of the variable x_i. Therefore, the corresponding problem Hamiltonian is

$$H_C = \sum_{j=1}^{m} | \sum_{i=1}^{m} d_{i,j} \hat{x}_i - b^j_{op} I |^2.$$

Obviously, for an m-dimensional lattice, the number of qubits required to optimize Babai's algorithm is $2m$.

To solve the problem, it is necessary to introduce a mixing Hamiltonian $H_M = \sum_{i=1}^{2m} \sigma^x_i$, where σ^x_i is the Pauli-X operator acting on the ith bit. The quantum circuit of QAOA is defined by the problem Hamiltonian H_C, the mixing Hamiltonian H_M and parameters (γ, β). For D-layer QAOA circuits, there are usually $2D$ variational parameters. The process of using QAOA to solve the optimization problem is shown in Figure 1, and the algorithm description is shown in Algorithm 4.

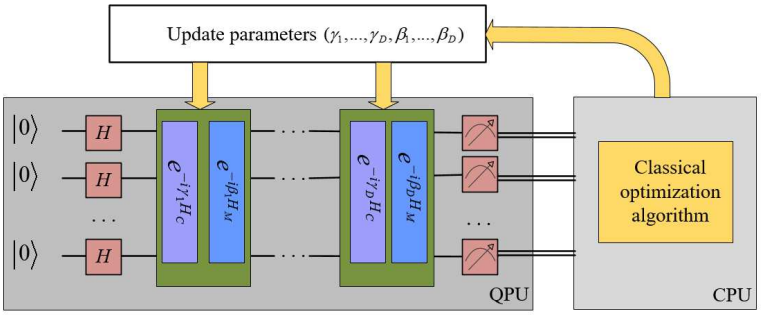

Figure 1. A schematic description of the VQE.

Algorithm 4 QAOA solving optimization.

Input: the problem Hamiltonian H_C, the mixing Hamiltonian H_M.
Output: the ground state $|\Psi_C\rangle$ of H_C.

1: Prepare the quantum register into $|\Psi_0\rangle = |+\rangle^{\otimes m}$.
2: Choose the initial parameters $\boldsymbol{\gamma}$, $\boldsymbol{\beta}$. Perform H_C and H_M alternately and obtain $|\Psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle$.
3: Measure the quantum registers and calculate the cost function.
4: Repeat Step 2 and Step 3 several times and calculate the expectation value of the cost function.
5: Pass the expectation value and parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ to a classical optimizer. Update the parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$.
6: Repeat Steps 2–5 until the result meets a fixed threshold and the parameters are updated to $(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$.
7: **return** $|\Psi_C\rangle = |\Psi(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)\rangle$

Now, we explain the steps in Algorithm 4. Step 1 performs $H^{\otimes m}$ on $|0\rangle^{\otimes m}$, and we obtain $|+\rangle^{\otimes m}$, which is an eigenvector of the Pauli-X operator.

Step 2 applies operators $e^{-i\gamma_k H_C}$ and $e^{-i\beta_k H_M}$, $k = 1, 2, \ldots, D$, alternately. So, we generate a variational wave function

$$|\phi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = e^{-i\gamma_D H_C} e^{-i\beta_D H_M} \ldots e^{-i\gamma_1 H_C} e^{-i\beta_1 H_M} |+\rangle^{\otimes m}. \tag{1}$$

The wave function has $2D$ parameters $\{\gamma_1, \ldots, \gamma_D, \beta_1, \ldots, \beta_D\}$.

The expectation value means

$$\langle \Psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) | H_C | \Psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle, \tag{2}$$

which can be obtained by repeatedly preparing $|\Psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle$ on the quantum processor and measuring it on a computational basis. Then, the classical computer performs classical optimization algorithms to find the optimal parameter. For example, the optimizers use the gradient descent algorithm to minimize the cost function in an iterative manner. The method calculates the first-order derivative of the function to compute the gradient. Then, it moves in the negative direction of the gradient. The termination condition of the gradient descent method is that the slope of the gradient is below a very small threshold. In the actual experiment, the algorithm is terminated by setting the empirical number of iterations.

In fact, classical optimization problems are often mapped to a simple Hamiltonian, which is diagonal in the computational basis. However, it does not mean that the problem is easy to solve or does not require a quantum solver. First, for example, MaxCut is a classical NP-hard problem, and the design of MaxCut problem Hamiltonian is $H = \sum_{ij} \frac{1}{2}(I - \sigma_i^z \sigma_j^z)$ [16]. In computational complexity theory, P is a set of relatively easy problems, and NP indicates hard problems. If MaxCut can be solved by classical computers easily, then P = NP, which completely overturns the theoretical basis of a range of fields. Second, processing classical optimization by QAOA usually requires a mixing Hamiltonian consisting of σ^x or σ^y, so quantum computers still work when solving classical optimization problems.

4. The Primal Method for Solving LWE

In this section, we propose a quantum primal method for solving LWE, where the Unique-SVP problem is solved by VQE. Although the quantum advantage of solving classical optimization by VQE is not as obvious as it is in quantum chemistry, understanding the evolution of the algorithm process is still crucial for improving algorithms running on classical hardware. We detail the number of qubits required and estimate the quantum resources when attacking the KYBER cryptosystem. With the development of quantum computers, resource estimation can also be used as a direction for comparison with pure classical algorithms.

4.1. LWE Algorithm

Algotithm 5 shows the procedure of the LWE algorithm.

Algorithm 5 The LWE algorithm.

Input: LWE samples $(\mathbf{A}, \mathbf{c} = \mathbf{As} + e) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$
Output: secret vector $\mathbf{s} \in \mathbb{Z}_q^n$

1: Construct a q-ary lattice $\Lambda_q(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}_q^m | \exists \mathbf{x} \in \mathbb{Z}^n, s.t. \mathbf{v} \equiv \mathbf{A}\mathbf{x} mod q\}$, whose lattice basis is equivalent to $\mathbf{B} = [\mathbf{A}|q\mathbf{I}_m]^T \in \mathbb{Z}^{(m+n)\times m}$.
2: Perform elementary row transformations on $\mathbf{B}$ and obtain the lattice basis $\mathbf{B}_1 = \begin{bmatrix} \mathbf{I}_n & \mathbf{A}'_{n\times(m-n)} \\ \mathbf{0} & q\mathbf{I}_{m-n} \end{bmatrix} \in \mathbb{Z}^{m\times m}$.
3: Using Kannan's embedding technique, reduce BDD to Unique-SVP and obtain $\mathbf{B}_2 = \begin{bmatrix} \mathbf{B}_1 & 0 \\ \mathbf{c} & M \end{bmatrix} \in \mathbb{Z}^{(m+1)\times(m+1)}$.
4: Process $\mathbf{B}_2$ with VQE and derive a short vector $\mathbf{e}$.
5: **return** $\mathbf{s} = \mathbf{A}^{-1}(\mathbf{c} - \mathbf{e})$

Step 3 expands the q-ary basis by one dimension and embeds the target vector $\mathbf{c}$ and the embedding factor M into matrix $\mathbf{B}_2$. When $M = \|\mathbf{e}\|$, there exists $(\mathbf{e}, -M) \in \mathcal{L}(\mathbf{B}_2)$ [26]. In this case, proposing the first m bits of the vector recovers $\mathbf{e}$. In the experiment, we generally take $M = 1$.

Unique-SVP can be seen as a special case of SVP, and step 4 in Algorithm 5 solves SVP by VQE. The detailed description is shown in Algorithm 6.

Algorithm 6 VQE solving SVP.

Input: the lattice basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_m]^T \in \mathbb{Z}^{(m+1)\times(m+1)}$.
Output: short vector $\mathbf{x}$.

1: Perform BKZ-reduction on $\mathbf{B}$.
2: The SVP problem is encoded to the ground state of the Hamiltonian operator H.
3: Construct parameterized quantum circuits.
4: Repeat preparing an ansatz state $|\Psi(\boldsymbol{\theta})\rangle$ from the parameterized quantum circuit and measuring it in Pauli-Z basis. Calculate the expectation value $C(\boldsymbol{\theta})$.
5: Pass $C(\boldsymbol{\theta})$ and parameters to a classical optimizer. Update the parameter $\boldsymbol{\theta}$ and go to step 4 until the expectation value converges.

The VQE procedure is visualized in Figure 2. Now, we explain the steps in Algorithm 6 in detail. In step 1, the larger the lattice size, the more quantum resources it occupies. In order to reduce the required qubits, a new basis matrix is first obtained by performing the BKZ reduction.

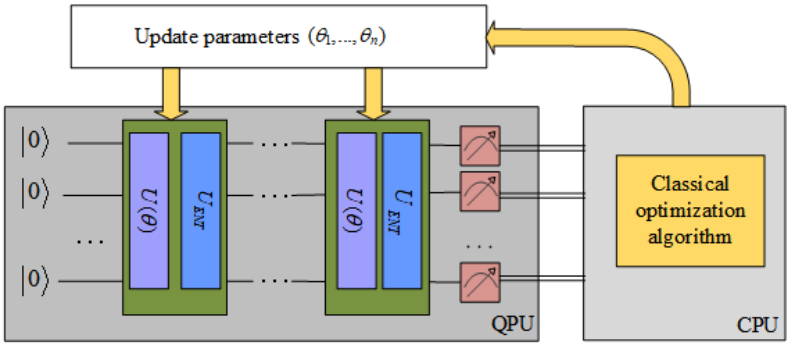

Figure 2. A schematic description of the VQE.

Step 2 constructs the problem Hamiltonian. For Lattice **B**, SVP is to find a nonzero vector **x** satisfying $min_{\mathbf{x}\in\mathcal{L}(\mathbf{B})}\|\mathbf{x}\|$. Let the row vector of coefficients be **z** and $\mathbf{z} \neq \mathbf{0}$; then, we have $\mathbf{x} = \mathbf{zB}$. Let $\mathbf{G} = \mathbf{BB}^T$; then, we have $\|\mathbf{x}\|^2 = \mathbf{zBB}^T\mathbf{z}^T = \mathbf{zGz}^T$. According to Algorithm 5, the dimension of the lattice is $m' = m + 1$. So, the SVP problem is equivalent to

$$min_{\mathbf{x}\in\mathcal{L}(\mathbf{B})}\|\mathbf{x}\|^2 = min_{\mathbf{z}\in\mathbb{Z}^{m'}}(\sum_{i=1}^{m'} z_i^2\mathbf{G}_{ii} + 2\sum_{0\leq i<j\leq m'} z_i z_j \mathbf{G}_{ij}). \tag{3}$$

Before mapping the SVP problem into a Hamiltonian, we first introduce the method of reducing numbers in the integer interval $[-d, d]$ to a Boolean variable polynomial. Let $t = \lfloor logd \rfloor$, introducing $t+1$ Boolean variables $\beta_0, \beta_1, \beta_2, \ldots, \beta_t$; the number in the interval can be expressed as $\sum_{i=0}^{t-1} 2^i\beta_i + (2d + 1 - 2^t)\beta_t - d$. Therefore, for the coefficient vector **z**, if each entry satisfies $|z_i| \leq d_i$, $i = 1, 2, \ldots, m'$, it can be expressed by Boolean variables $\beta_{i0}, \ldots, \beta_{it_i}$. Substituting the Boolean variable polynomials into (3), we have

$$min_{\beta_{10},\ldots,\beta_{1t_1},\ldots,\beta_{m'0},\ldots,\beta_{m't_{m'}}}(h + \sum_{ij} h_{ij}\beta_{ij}^2 + \sum_{ij\neq kl} l_{ij,kl}\beta_{ij}\beta_{kl}),$$

where $h, h_{ij}, l_{ij,kl}$ are calculated constants. Because β_{ij} are Boolean variables, the above equation is equivalent to

$$min_{\beta_{10},\ldots,\beta_{1t_1},\ldots,\beta_{m'0},\ldots,\beta_{m't_{m'}}}(h + \sum_{ij} h_{ij}\beta_{ij} + \sum_{ij\neq kl} l_{ij,kl}\beta_{ij}\beta_{kl}). \tag{4}$$

In the above formula, it is required to find the parameter vector

$$\boldsymbol{\beta} = (\beta_{10}, \ldots, \beta_{1t_1}, \ldots, \beta_{m'0}, \ldots, \beta_{m't_{m'}}) \tag{5}$$

to minimize the function

$$\sum_{ij} h_{ij}\beta_{ij} + \sum_{ij\neq kl} l_{ij,kl}\beta_{ij}\beta_{kl}.$$

Encoding the cost function into a Hamiltonian requires a mapping $\beta_{ij} \to (1 - \gamma_{ij})/2$, where $\gamma_{ij} \in \{-1, 1\}$. Then, substitute $\gamma_{ij} \to \sigma_{ij}^z$ and $1 \to I_{ij}$ to obtain the problem Hamiltonian

$$H = \sum_{ij} h_{ij}\frac{I_{ij} - \sigma_{ij}^z}{2} + \sum_{ij\neq kl} l_{ij,kl}\frac{I_{i,j} - \sigma_{ij}^z}{2} \otimes \frac{I_{kl} - \sigma_{kl}^z}{2},$$

where $ij, kl \in \{10, \ldots, 1t_1, m'0, \ldots, m't_{m'}\}$ and σ_i^z is the Pauli-Z operator acting on the ith bit. The Hamiltonian acts on a Hilbert space spanned by $QNum$ qubits, and it can also be written as a sum over many local interactions.

To find the ground state of H, step 3 generates a hardware-efficient trial wavefunction, which is more suitable for available quantum devices [27]. Let $|\Psi(\boldsymbol{\theta})\rangle = (U(\boldsymbol{\theta})U_{ENT})^D|\Psi_0\rangle$ and the reference state is set to $|00..0\rangle$. $U(\boldsymbol{\theta})$ are a group of single-qubit rotations determined by rotation angles $\boldsymbol{\theta}$. U_{ENT} are entangling drift operations generating sufficient entanglement. D defines the level of the quantum circuit. Obviously, with the increase of D, the convergence speed increases, but the fidelity decreases.

Step 4 calculates $C(\boldsymbol{\theta}) = \langle\Psi(\boldsymbol{\theta})|H|\Psi(\boldsymbol{\theta})\rangle$. Each iteration requires measuring N times and the cost obtained for the i-th time is C_i. Then, the expectation value is

$$C(\boldsymbol{\theta}) = \langle\Psi(\boldsymbol{\theta})|H|\Psi(\boldsymbol{\theta})\rangle = \frac{1}{N}\sum_{i=1}^{N} C_i. \tag{6}$$

If the Hilbert space is too large, because the interaction is local, the Hamiltonian can be split into a summation over many terms. The expectation calculations for one term are relatively simple, and we can speed up the computation by parallelizing the quantum

expectation-value estimation algorithm [28]. After calculating the expectation of each item on the quantum processor, multiply it by the weight and sum on the classical processor to obtain the final expectation value.

However, the shortest vector is $\mathbf{0}$ in this algorithm, so the restriction $\mathbf{x} \neq \mathbf{0}$ needs to be added. The idea is to increase C when appearing as $\mathbf{0}$. We assume that among the N measurements, there are N_0 results that are not $\mathbf{0}$, $C = \frac{1}{N_0}\sum_{i=1}^{N} C_i$. Obviously, the larger the N_0, the smaller the C.

Step 5 uses the classical optimization algorithm to update $\boldsymbol{\theta}$ until the expectation value converges and the process is similar to QAOA.

We give a toy example to illustrate the process on the quantum processor. For a more convenient description, the LWE dimension is further limited, and the example also supports simple experiments on the IBM quantum system. Let $q = 3.n = 1, m = 2$. The samples are $s + e_1 = 1 mod 3, 2s + e_2 = 2 mod 3$. The LLL-reduced matrix after Kannan's embedding is

$$\begin{bmatrix} 0 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 2 & 0 \end{bmatrix}.$$

To simplify the model, suppose $z_i, i = 1, 2, 3$, are already Boolean variables. Then, the SVP problem can be reduced into finding the minimum value of $C = z_1 + 2z_2 + 5z_3 + 2z_2z_3$. The problem Hamilton is

$$H = 4.5I_1 \otimes I_2 \otimes I_3 - 0.5Z_1 \otimes I_2 \otimes I_3 - 1.5I_1 \otimes Z_2 \otimes I_3 - 3I_1 \otimes I_2 \otimes Z_3 + 0.5I_1 \otimes Z_2 \otimes Z_3. \quad (7)$$

Now, construct a hardware-efficient Ansatz consisting of several parameterized single-qubit rotation operations and controlled-NOT gates. Using the parameterized circuit shown in Figure 3, any 3-qubit quantum state $|\Psi(\boldsymbol{\theta})\rangle$ can be prepared, and different quantum states can be output by adjusting the six parameters.

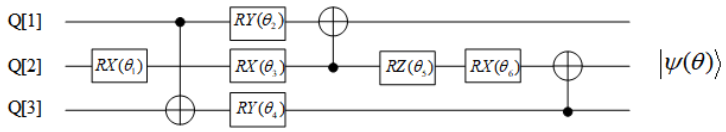

Figure 3. Quantum circuit for 3 qubits.

After preparing the ansatz state and measuring it repeatedly, we calculate the expectation value. Then, we perform the optimization process on the classical processor. Iterate the above process, and finally, the parameters corresponding to the optimal result are $(0, \pi, 0, 0, 0, \pi)$ and $[z_1, z_2, z_3] = [1, 0, 0]$. So, $[e_1, e_2] = [0, 0]$, $s = 1$.

4.2. Algorithm Analysis

First, we analyze the range of d_i in the restriction condition $|z_i| < d_i, i = 1, 2, \ldots, m'$. Let $\widetilde{\mathbf{B}} = (\mathbf{B}^{-1})^T = [\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_{m'}]^T$; then, there exists $\langle \mathbf{b}_i, \widetilde{\mathbf{b}}_i \rangle = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$. Let the shortest vector $\mathbf{v} = \sum_{i=1}^{m'} t_i \mathbf{b}_i$; then, $|\langle \mathbf{v}, \widetilde{\mathbf{b}}_i \rangle| = |t_i| \leq \|\mathbf{v}\| \|\widetilde{\mathbf{b}}_i\|$. Due to the Gaussian heuristic, $\|\mathbf{v}\| = \sqrt{\frac{m'}{2\pi e}} vol(\mathcal{L})^{1/m'}$, we have $|t_i| \leq \sqrt{\frac{m'}{2\pi e}} vol(\mathcal{L})^{1/m'} \|\widetilde{\mathbf{b}}_i\|$.

For an m'-dimensional matrix $\mathbf{B}$, its orthogonality defect $\delta(\mathbf{B}) = \frac{\prod_{i=1}^{m'} \|\mathbf{b}_i\|}{|det(\mathbf{B})|}$. Obviously, for $\mathbf{B}$, there exists $\delta(\mathbf{B}) \geq 1$ and $\delta(\mathbf{B}) = 1$ if and only if $\mathbf{B}$ is an orthogonal matrix. Therefore, the total number of qubits can be expressed as

$$QNum = \sum_{i=1}^{m'} (\lfloor log d_i \rfloor + 1) \leq m' + log(d_1 d_2 \ldots d_{m'}), \quad (8)$$

where $log(d_1 d_2 \dots d_{m'}) \leq 0.5m'log(\frac{m'}{2\pi e}) + log(vol(\mathcal{L}) \prod_{i=1}^{m'} \|\widetilde{\mathbf{b}}_i\|) = 0.5m'log(\frac{m'}{2\pi e}) + log(\delta(\widetilde{\mathbf{B}}))$. For a KZ-reduced matrix $\mathbf{B}$, its orthogonality defect satisfies [29]

$$\delta(\mathbf{B}) \leq (\frac{1}{8}m' + \frac{6}{5})^{m'/2} (\prod_{i=1}^{m'} \frac{\sqrt{i+3}}{2}) \leq (\frac{1}{8}m' + \frac{6}{5})^{m'/2} (m'+3)^{m'/2} (\frac{1}{2})^{m'}.$$

So,

$$log(\delta(\widetilde{\mathbf{B}})) \leq \frac{m'}{2} log(\frac{1}{8}m' + \frac{6}{5}) + \frac{m'}{2} log(m'+3) - m' \leq m'log(m'+3) - m'.$$

Substituting into Equation (8), we have

$$\begin{aligned} QNum &\leq m' + (\frac{m'}{2} log(m') - \frac{m'}{2} log(2\pi e) + m'log(m'+3) - m') \\ &= \frac{m'}{2} log(m') - \frac{m'}{2} log(2\pi e) + m'log(m'+3) \end{aligned} \tag{9}$$

Therefore, the maximum number of qubits is $O(m'logm')$. Now, we review the value of $d_i, i = 1, 2, \dots, m'$ when using VQE for enumeration. In practice, each z_i is represented by $QNum/m'$ qubits and the range of d_i is $[2^{(QNum/m')-1}, 2^{(QNum/m')} - 1]$, where $d_i \in \mathbb{Z}$.

In Kannan's embedding, the lattice dimension is $m+1$, where m is the sample number. In most cases, an LWE-based scheme produces only $m = poly(n)$ LWE samples (and the polynomial bound can be as small as $m = \Theta(n)$). In the LWE-based cryptosystem proposed in paper [12], $m = \sqrt{nlg(q)/lg(\delta)}$ and δ here means the root-Hermite factor. The theoretical worst-case reduction for LWE requires $\alpha q \geq 2\sqrt{n}$ [4], so we set $\alpha q = 2\sqrt{n}$. Now, we analyze the average number of qubits required, and its LWE parameters are shown in Table 1.

Table 1. LWE parameters.

meirent	n	10	20	30	40
	q	2053	2053	2053	2053
	αq	6.3246	8.9444	10.954	12.649
	m	34	65	91	127
	δ	1.069	1.0365	1.0280	1.0191

There are 4 groups of parameters in the table. For each group, 10 experiments are performed, and the average value of the cost function C is obtained. Finally, we calculate the average number of qubits required, and the result is illustrated in Figure 4. The four curves with different colors represent that the preprocessing method for the lattice basis is LLL, BKZ-20, BKZ-40 and BKZ-80, respectively. By the regression analysis, taking BKZ-20 as an example, we have

$$QNum = 92.54nlogn - 612.27n + 1343.8logn - 1234.37.$$

For example, for a 40-dimensional LWE problem, the maximum number of qubits required is 1126, which is a scale that is considered achievable in the near future. With the further development of quantum computers, LWE with larger dimensions can also be solved successively.

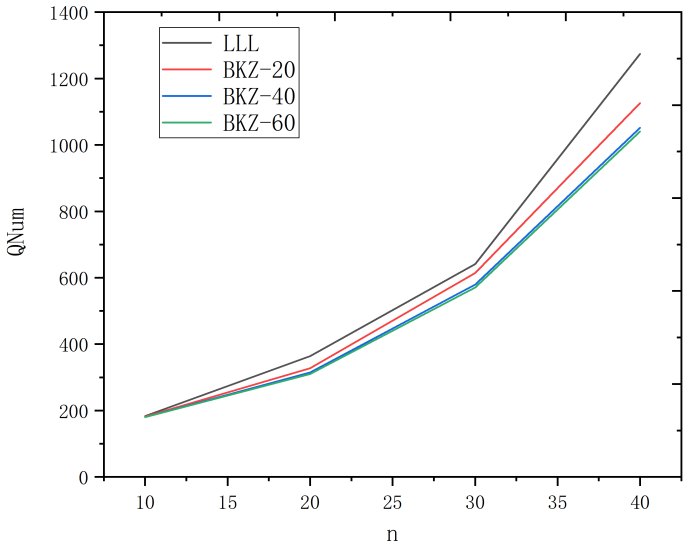

Figure 4. Average number of qubits required for different LWE dimensions.

4.3. Attacks on Existing Cryptosystems

In this section, we calculate the number of qubits required for a VQE attack on the KYBER cryptosystem. KYBER is a key encapsulation mechanism based on the module-LWE problem, which means it is based on Ring $R = \mathbb{Z}[X]/(X^{256} + 1)$. KYBER has three modes to satisfy 128/192/256-bit security, respectively. The parameters are listed in Table 2.

Table 2. KYBER parameters.

	n	k	q
KYBER512	256	2	3329
KYBER768	256	3	3329
KYBER1024	256	4	3329

In the table, n, k, q represents the maximum degree of polynomial, the number of polynomials in each vector and the modulus. The most famous attack on the MLWE problem does not utilize the special structure of a lattice, so we still analyze it as an LWE problem. Paper [7] mentioned that the number of samples is between 0 and $(k+1)n$. To analyze the worst case, let $m = (k+1)n$. Therefore, in the primal attack, the lattice dimension $d = m+1 = (k+1)n+1$. Using the conclusion in Section 4.2, for the above three parameter settings, the required maximum qubits are 13,768, 19,538, and 25,482, respectively.

Although the quantum computers made at this stage are all NISQ devices, after IBM launched the 127-QubitEagle processor in 2021, it plans to launch the 1121-QubitCondor processor in 2023. At the same time, the IBM team also fully considered the future million-qubit system when designing the world's largest dilution refrigerator "Goldeneye", which is an important part of the IBM's roadmap for scaling quantum technology.

5. Algorithm Implementation and Experimental Results

5.1. Using QAOA Algorithm to Improve the Decoding Method

In this section, we discuss the quantum advantage of the algorithm introduced in Section 3. Since it is difficult to estimate the computing complexity of QAOA, the QAOA process is regarded as a black box; that is, it is assumed that QAOA returns the solution to the optimization problem in a limited time. Now, without considering the actual complexity of QAOA, we only analyze the results of the algorithm through small-scale experiments.

The LWE instance is $(\mathbf{A}, \mathbf{c} = \mathbf{As} + \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$. Thus, after reducing to the BDD problem, the target vector is $\mathbf{c}$. Algorithm 3 outputs a classical closest vector $\mathbf{w}$, and the error vector can be obtained by $\mathbf{e} = \mathbf{c} - \mathbf{w}$. Then, Algorithm 4 updates vectors $\mathbf{w}$ and $\mathbf{e}$ by

QAOA. The result quality $r = \|\mathbf{e}\|$, which means the norm of the error vector. It is obvious that the smaller the r, the higher the quality.

Taking the dimension of the secret vector as $n = 3$ and $n = 5$, the experiment generates 50 groups of random LWE samples, respectively. Each group forms an LWE instance. For each instance, after obtaining the closest vector by Babai's algorithm and calculating the result quality r, we use QAOA for optimization to obtain a new approximate closest vector and calculate the quality. Figure 5 shows the comparison of r between classical solutions and solutions after quantum optimization when $n = 3$, and Figure 6 illustrates the comparison when $n = 5$.

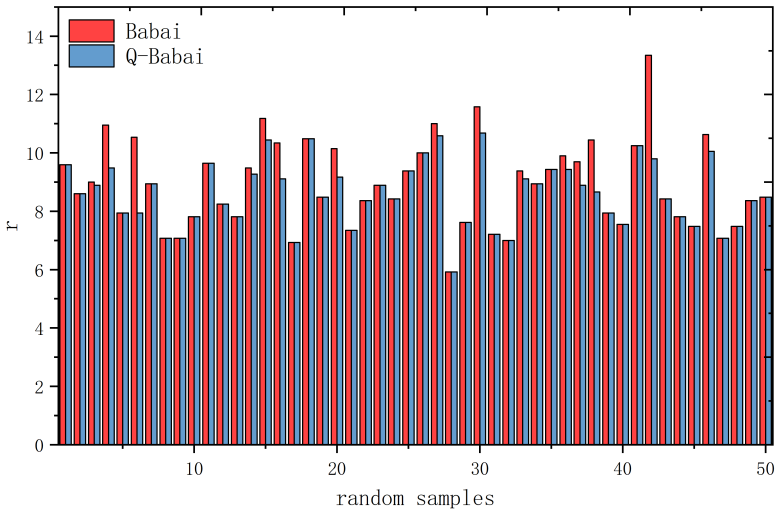

Figure 5. Quantum advantage demonstration of 50 random lattice samples when $n = 3$.

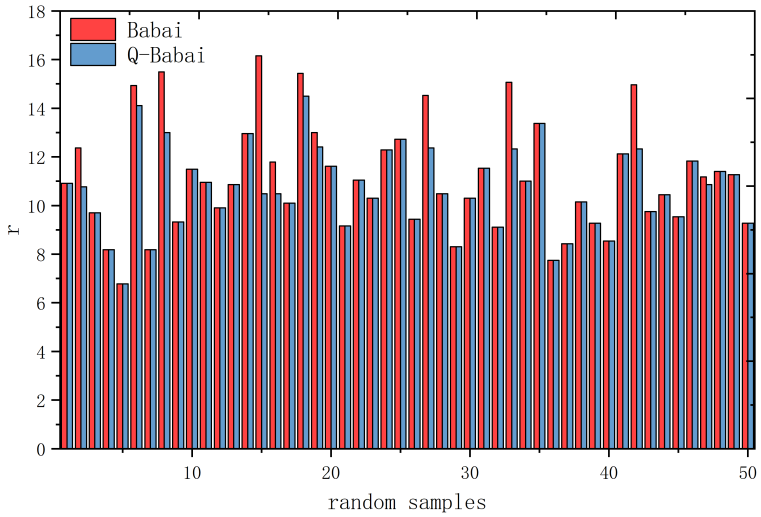

Figure 6. Quantum advantage demonstration of 50 random lattice samples when $n = 5$.

In Figures 5 and 6, the horizontal axis represents 50 groups of random samples, and the vertical axis represents the result quality r. The red columns represent the results of the classical Babai's algorithm, and the blue columns represent the results after quantum optimization. According to the definition $r = \|\mathbf{e}\|$, a smaller r indicates a closer vector and higher quality. As evident in the figures, quantum results have higher quality than classical results in many cases, while in other cases, the results are the same. Therefore, the conclusion that can be drawn from the experiment is that quantum results obtained by QAOA are no worse than their classical counterparts.

5.2. Using VQE Algorithm to Realize the Primal Method

In this section, we present the experiments of solving LWE by the primal method. When quantum simulation is performed in a classical computer, the underlying quantum simulation uses QuSET [30], and the front-end interface to implement the algorithm uses C++. In the experiment, better results can be obtained by using the Conditional Vale at Risk (CVaR) method [31]. Specifically, assuming $C_1, C_2, \ldots C_n$ are sorted in non-decreasing order and in each loop, $C = \frac{1}{\lceil pN \rceil} \sum_{i=1}^{\lceil pN \rceil} C_i$, where $0 < p < 1$. Paper [21] proposes that $p = 0.175$ gives better results.

On the simulation platform, due to memory constraints, the maximum lattice dimension does not exceed 30, which means the LWE dimension n is much smaller than 30. If the input lattice matrix already contains the shortest vector, since the initial parameters of VQE are random and the algorithm still outputs the shortest vector after several iterations, it verifies the correctness of the algorithm. Therefore, when the VQE input is the reduced basis or the shortest vector can be obtained by simple vector addition or subtraction of the input matrix, the solution obtained by VQE is the same as that of the classical algorithm.

When the input is an arbitrary basis, the actual experimental results of the VQE are of poorer quality. The reason is that the simulation platform occupies classical memory, and the qubits for representing entries of the coefficient vector are limited. So, the correct coefficient vector cannot be accurately obtained. As the number of available qubits increases in the future, its coefficient representation will become more and more accurate, and the solution quality of the VQE algorithm will be better.

6. Discussion and Conclusions

VQA uses a classical optimizer to train parameterized quantum circuits, and it is one of the most promising quantum algorithms to achieve quantum supremacy. When researchers envision applications for quantum computers, it is almost impossible to bypass VQA algorithms. In this paper, we first present two LWE attacking tools, using QAOA to improve Babai's algorithm when solving BDD and utilizing VQE to solve Unique-SVP. The two algorithms combine classical optimization techniques and variational quantum techniques, providing ideas for solving LWE when the quantum resources are limited. Second, we estimate the number of qubits required for both algorithms. Third, for the two algorithms, experimental simulations are carried out, respectively. The experimental results show that for the first algorithm, QAOA improves the result quality of classical algorithms, and for the second algorithm, when the memory is large enough, the quality of quantum solutions is at least comparable to that of the classical solutions. How to further reduce the number of qubits by using the structure of the modular lattice is the direction that needs to be studied in the future.

Author Contributions: Formal analysis, L.L., B.Y. and H.W.; supervision, Z.M.; implement, Y.F. and X.M.; data analysis, Q.D.; writing—original draft preparation, L.L. and B.Y.; writing—review and editing, H.W. and Z.M.; funding acquisition, Z.M. All authors were involved in refining the ideas and writing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grants No. 61972413, 61901525, 62002385).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ajtai, M. Generating hard instances of lattice problems (extended abstract). In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing (STOC '96), New York, NY, USA, 22–24 May 1996; pp. 99–108.
2. Ajtai, M.; Dwork, C. A public-key cryptosystem with worst-case/average-case equivalence. In Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing (STOC '97), New York, NY, USA, 4–6 May 1997; pp. 284–293.
3. Hoffstein, J.; Pipher, J.; Silverman, J.H. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory*; Buhler, J.P., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1423.
4. Regev, O. On lattices, learning with errors, random linear codes, and cryptography. In Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing (STOC '05), New York, NY, USA, 22–24 May 2005; pp. 84–93.
5. Applebaum, B.; Cash, D.; Peikert, C.; Sahai, A. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In *Advances in Cryptology-CRYPTO 2009*; Halevi, S., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5677.
6. Brakerski, Z.; Vaikuntanathan, V. Efficient Fully Homomorphic Encryption from (Standard) LWE. In Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, Palm Springs, CA, USA, 22–25 October 2011.
7. Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-KYBER: Algorithm Specifications and Supporting Documentation. 2021. Available online: https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf (accessed on 15 February 2022).
8. Bai, S.; Ducas, L.; Kiltz, E.; Leopoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Dilithium: Algorithm Specifications and Supporting Documentation (Version 3.1). Available online: https://pq-crystals.org/dilithium/data/dilithiumspecification-round3-20210208.pdf (accessed on 30 January 2022).
9. Blum, A.; Kalai, A.; Wasserman, H. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM* **2003**, *50*, 506–519. [CrossRef]
10. Arora, S.; Ge, R. New Algorithms for Learning in Presence of Errors. In *Automata, Languages and Programming. ICALP 2011*; Aceto, L., Henzinger, M., Sgall, J., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6755.
11. Babai, L. On Lovász' lattice reduction and the nearest lattice point problem. In *STACS 1985*; Mehlhorn, K., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1984; Volume 182.
12. Lindner, R.; Peikert, C. Better Key Sizes (and Attacks) for LWE-Based Encryption. In *Topics in Cryptology—CT-RSA 2011*; Kiayias, A., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6558.
13. Albrecht, M.R.; Fitzpatrick, R.; Göpfert, F. On the Efficacy of Solving LWE by Reduction to Unique-SVP. In *Information Security and Cryptology—ICISC 2013*; Lee, H.S., Han, D.G., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8565, pp. 293–310.
14. Kannan, R. Minkowski's Convex Body Theorem and Integer Programming. *Math. Oper. Res.* **1987**, *12*, 415–440. [CrossRef]
15. Bai, S.; Galbraith, S.D. An Improved Compression Technique for Signatures Based on Learning with Errors. In *Topics in Cryptology—CT-RSA 2014*; Benaloh, J., Ed.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2014; Volume 8366.
16. Farhi, E.; Goldstone, J.; Gutmann, S. A quantum approximate optimization algorithm. *arXiv* **2014**, arXiv:1411.4028.
17. Peruzzo, A.; Mcclean, J.; Shadbolt, P.; Yung, M.H.; Zhou, X.Q.; Love, P.J.; Aspuru-Guzik, A.; O'Brien, J.L. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **2014**, *5*, 4213. [CrossRef]
18. Wei, S.; Li, H.; Long, G. A Full Quantum Eigensolver for Quantum Chemistry Simulations. *Research* **2020**, *2020*, 1486935. [CrossRef] [PubMed]
19. Joseph, D.; Callison, A.; Ling, C.; Mintert, F. Two quantum Ising algorithms for the shortest-vector problem. *Phy. Rev. A* **2021**, *103*, 032433. [CrossRef]
20. Joseph, D.; Ghionis, A.; Ling, C.; Mintert, F. Not-so-adiabatic quantum computation for the shortest vector problem. *Phys. Rev. Res.* **2020**, *2*, 013361. [CrossRef]
21. Albrecht, M.R.; Prokop, M.; Shen, Y.; Wallden, P. Variational quantum solutions to the Shortest Vector Problem. *IACR Cryptol. ePrint Arch.* **2022**, *2022*, 233.
22. Lenstra, A.K.; Lenstra, H.W.; Lovász, L. Factoring polynomials with rational coefficients. *Math. Ann.* **1982**, *261*, 515–534. [CrossRef]
23. Schnorr, C.-P.; Euchner, M. Lattice basis reduction: Improved practical algorithms and solving subset sumproblems. *Math. Program.* **1994**, *66*, 181–199. [CrossRef]
24. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [CrossRef]
25. Bharti, K.; Cervera-Lierta, A.; Kyaw, T.H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Heimonen, H.; Kottmann, J.S.; Menke, T.; et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **2022**, *94*, 015004. [CrossRef]
26. Lyubashevsky, V.; Micciancio, D. On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem. In *Advances in Cryptology-CRYPTO 2009*; Halevi, S., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; pp. 577–594.
27. Nikolaj, M.; Panagiotis, B.; Bishop, L.S.; Chow, J.M.; Cross, A.; Egger, D.J.; Filipp, S.; Fuhrer, A.; Gambetta, J.M.; Ganzhorn, M. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Sci. Technol.* **2018**, *3*, 030503.

28. Mcclean, J.R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **2016**, *18*, 023023. [CrossRef]
29. Wen, J.; Chang, X.-W. On the KZ Reduction. In Proceedings of the IEEE International Symposium on Information Theory (ISIT 2015), Hong Kong, China, 14–19 June 2015; Volume 65, pp. 1921–1935
30. Jones, T.; Brown, A.; Bush, I.; Benjamin, S.C. QuEST and High Performance Simulation of Quantum Computers. *Sci. Rep.* **2019**, *9*, 10736. [CrossRef]
31. Barkoutsos, P.K.; Nannicini, G.; Robert, A.; Tavernelli, I.; Woerner, S. Improving Variational Quantum Optimization using CVaR. *Quantum* **2020**, *4*, 256. [CrossRef]