Quantum algorithm for solving the hidden subgroup problems

Hefeng Wang

MOE Key Laboratory for Nonequilibrium Synthesis and Modulation of Condensed Matter,
Xi’an Jiaotong University, Xi’an, 710049, China
School of Physics, Xi’an Jiaotong University and Shaanxi Province Key Laboratory
of Quantum Information and Quantum Optoelectronic Devices, Xi’an, 710049, China

Many problems for which quantum algorithms achieve exponential speedup over classical algorithms can be reduced to the Abelian hidden subgroup problems (HSP). However, there is no efficient quantum algorithm for solving the non-Abelian HSP. We proposed a quantum algorithm for efficiently solving a type of search problems with a hierarchical structure through a multistep quantum computation process. In this work, we apply the algorithm for solving the HSP and problems that can be reduced to the Abelian and the non-Abelian HSP. We demonstrate that both the Abelian and the non-Abelian HSP can be reduced to the hierarchical structured search problems, therefore they can be solved efficiently by using our algorithm.

I. INTRODUCTION

In quantum computation, a problem can be solved by finding the ground state of the corresponding problem Hamiltonian that encodes the solution to the problem, e.g. adiabatic quantum computing (AQC) [1]. In Ref. [2], we proposed a quantum algorithm for finding the ground state of a problem Hamiltonian through a multistep quantum computation process, and apply the algorithm for solving a type of search problems with a hierarchical structure in polynomial time. In this algorithm, we construct a sequence of intermediate Hamiltonians \( \{H_i, i = 0, \ldots, m-1\} \) by decomposing the problem into its structure to form a Hamiltonian evolution path from the initial Hamiltonian \( H_0 \) to the problem Hamiltonian \( H_P = H_m \). Then we start from the ground state of the initial Hamiltonian, evolve it through the ground states of the intermediate Hamiltonians sequentially, finally reach the ground state of the problem Hamiltonian in \( m \) steps. In each step, the ground state of an intermediate Hamiltonian is obtained deterministically through quantum resonant transition (QRT) [3, 4], and the state is protected through quantum entanglement, so that it can be used repeatedly without making copies of the state, therefore we can perform a multistep quantum computation process and we do not need to run the algorithm from beginning once it fails to obtain the ground state of an intermediate Hamiltonian. This algorithm can be run efficiently if (i) the energy gap between the ground and the first excited states of each Hamiltonian is not exponentially small, and (ii) the overlaps between the ground states of any two adjacent Hamiltonians are not exponentially small. We have demonstrated that the conditions for efficiently running this algorithm are not equivalent to that of the common AQC algorithms, where the adiabatic evolution path is constructed as a linear interpolation between the initial Hamiltonian and the problem Hamiltonian, or through a sequence of adiabatic evolution Hamiltonians constructed by linear interpolation of pairs of adjacent intermediate Hamiltonians following the same Hamiltonian evolution path as in our algorithm. Detailed comparisons between our algorithm and the AQC algorithms are shown in Ref. [2].

We have applied this algorithm for efficiently solving the hierarchical structured search problems, where the search space of the problem is reduced in a polynomial rate by using \( m \) [in order of \( O(\log N) \)] oracles, and \( N \) is the size of the search space of the problem. The conditions of the algorithm are satisfied in this case, and the hierarchical structured search problems can be solved in \( m \) steps in polynomial time. Many problems for which quantum algorithm achieves exponential speedup over its classical counterparts can be reduced to the Abelian hidden subgroup problems (HSP), e.g. the graph isomorphism (GI) problem [5]. It has been realized that the structure of a problem is the key for whether or not it can be solved efficiently on a quantum computer [6]. In this work, we analyze the spectrum structure of the HSP. We find that the HSP can be reduced to the hierarchical structured search problems, and we apply this algorithm for solving the HSP and problems that can be reduced to both the Abelian HSP and the non-Abelian HSP efficiently.

This paper is organized as follows: in Sec. I, we describe the algorithm and its application for solving a special case of the hierarchical structured search problems; then in Sec. II, we apply the algorithm for solving the HSP and problems that can be reduced to the HSP, including the Deutsch-Jozsa problem, the Simon’s problem, the factoring problem, the discrete logarithm problem, the period-finding problem and the order-finding problem that can be reduced to the Abelian HSP, and the graph isomorphism problem and the poly(n)-unique shortest vector problem (SVP) in a lattice that can be reduced to the non-Abelian HSP; we close with a discussion section.

* wanghf@mail.xjtu.edu.cn
II. THE ALGORITHM AND ITS APPLICATION FOR SOLVING THE HIERARCHICAL STRUCTURED SEARCH PROBLEMS

In the following, we describe a special case of the algorithm for finding the ground state of a problem Hamiltonian via a multistep quantum computation process [2], where the ground state eigenvalues of the intermediate Hamiltonians and the overlaps between the ground states of any two adjacent Hamiltonians are known in advance. The algorithm requires \( n+1 \) qubits with one probe qubit and an \( n \)-qubit register \( R \) representing a problem of dimension \( N = 2^n \). We use a step of the algorithm to illustrate the running procedures of the algorithm.

In the \( i \)th step of the algorithm, we are given the Hamiltonian \( H_{i-1} \), its ground state eigenvalue \( E_0^{(i-1)} \) and the ground state \( |\varphi_0^{(i-1)}\rangle \) obtained from the previous step, and the Hamiltonian \( H_i \) and its ground state eigenvalue \( E_0^{(i)} \), the goal is to prepare the ground state \( |\varphi_0^{(i)}\rangle \) of \( H_i \) by using the QRT method. The algorithm Hamiltonian of the step is

\[
H^{(i)} = -\frac{1}{2} \omega \sigma_z \otimes I_N + H_R^{(i)} + c \sigma_x \otimes I_N, \tag{1}
\]

where

\[
H_R^{(i)} = \alpha_i |1\rangle \langle 1| \otimes H_{i-1} + |0\rangle \langle 0| \otimes H_i, \quad i = 1, \ldots, m, \tag{2}
\]

\( I_N \) is the \( N \)-dimensional identity operator, and \( \sigma_x, \sigma_z \) are the Pauli matrices. The first term in Eq. (1) is the Hamiltonian of the probe qubit, the second term contains the Hamiltonian of the register \( R \) and describes the interaction between the probe qubit and \( R \), and the third term is a perturbation. The parameter \( \alpha_i \) is used to rescale the energy levels of \( H_{i-1} \), and the ground state energy of \( \alpha_i H_{i-1} \) is used as a reference energy level to the ground state eigenvalue of \( H_i \), and \( c \ll 1 \).

The initial state of the \( n+1 \) qubits is set as \( |1\rangle |\varphi_0^{(i-1)}\rangle \), which is an eigenstate of \( H_R^{(i)} \) with eigenvalue \( \alpha_i E_0^{(i-1)} \). By setting \( \alpha_i = \left( E_0^{(i)} - \omega \right) / E_0^{(i-1)} \), the condition \( E_0^{(i)} - \alpha_i E_0^{(i-1)} = \omega \) for a resonant transition between the probe qubit and the transition between states \( |\varphi_0^{(i-1)}\rangle \) and \( |\varphi_0^{(i)}\rangle \) is satisfied. The probe qubit decays to its ground state \( |0\rangle \) and the register \( R \) evolves to the ground state \( |\varphi_0^{(i)}\rangle \) of \( H_i \) when the resonant transition occurs. We can set the optimal evolution time of the \( i \)th step of the algorithm as \( t_i = \pi / (2c d_0^{(i)}) \) at which the probability for the system to be evolved to the state \( |0\rangle |\varphi_0^{(i)}\rangle \) reaches its maximum, since the overlap \( p_0^{(i)} = \langle \varphi_0^{(i-1)} | \varphi_0^{(i)} \rangle \) between the ground states of \( H_{i-1} \) and \( H_i \) is known already.

The procedures of the \( i \)th step of the algorithm are as follows:

(i) Initialize the probe qubit to its excited state \( |1\rangle \) and the register \( R \) in state \( |\varphi_0^{(i-1)}\rangle \).

(ii) Set \( \alpha_i = \left( E_0^{(i)} - \omega \right) / E_0^{(i-1)} \), and implement the time evolution operator \( U(t_i) = \exp (-iH^{(i)} t_i) \).

(iii) Read out the state of the probe qubit. The system is approximately in the state

\[
\sqrt{1 - p_0^{(i)}} |\varphi_0^{(i-1)}\rangle + \sqrt{p_0^{(i)}} |\varphi_0^{(i)}\rangle
\]

as the resonant transition occurs, where \( p_0^{(i)} = \sin^2 \left( ct_i d_0^{(i)} \right) \) is the decay probability of the probe qubit, it is close to one since the evolution time \( t_i \) is optimal. The state \( |\varphi_0^{(i-1)}\rangle \) is protected in this entangled state. By performing a measurement on the probe qubit, if the probe decays to its ground state \( |0\rangle \), it means that the system evolves to the state \( |0\rangle |\varphi_0^{(i)}\rangle \); if the probe qubit stays in state \( |1\rangle \), the register \( R \) remains in state \( |\varphi_0^{(i-1)}\rangle \), then we repeat procedures (ii)-(iii) until the probe qubit decays to its ground state \( |0\rangle \). In this way, the ground state \( |\varphi_0^{(i)}\rangle \) of \( H_i \) can be obtained deterministically. By protecting the state \( |\varphi_0^{(i-1)}\rangle \) through quantum entanglement, the state can be used repeatedly without copying it. We do not need to run the algorithm from beginning once it fails to obtain the desired state in a middle step, but just repeat procedures of the step until the desired state is obtained. Therefore, this algorithm circumvents the restriction of the no-cloning theorem [7, 8] to realize a multistep quantum computation process without making copies of an intermediate state. The success probability of the algorithm is [2]

\[
P_{\text{succ}} \geq \prod_{i=1}^{m} \left[ 1 - a_i^2 c^2 \right] \geq \left[ 1 - (a_{\text{max}} c)^2 \right]^m,
\]

where \( a_i^2 = \left[ 1 - \left( d_0^{(i)} \right)^2 \right] / \left[ E_1^{(i)} - E_0^{(i)} \right]^2 \) and \( E_1^{(i)} \) is the eigenvalue of the first excited state of \( H_i \), and \( a_{\text{max}} \) is the maximum value of \( a_i \). The coefficient \( c \) can be set such that \( a_{\text{max}} c < 1 / \sqrt{m} \), then \( P_{\text{succ}} > 1 / e \) in the asymptotic limit of \( m \). The evolution time of the \( i \)th step is proportional to \( \pi / (2c d_0^{(i)}) \), therefore the total runtime of the algorithm scales as \( O(\sum_{i=1}^{m} \pi / (2c d_0^{(i)}) \) (\( m \)). The algorithm can be run efficiently if the overlap \( d_0^{(i)} \) and the energy gap \( E_1^{(i)} - E_0^{(i)} \) are not exponentially small.

This algorithm has been applied for solving the hierarchical structured search problems [2] which can be described as follows: the problem contains \( N \) items and can be decomposed by using \( m \) oracles to construct \( m \) Hamiltonians \( H_P \), as

\[
H_P = - \sum_{q_i \in \Pi_i} |q_i\rangle \langle q_i|, \tag{3}
\]

and

\[
H_P = H_m = H_P = - |q\rangle \langle q|, \tag{4}
\]

where the set \( \Pi_i \) contains \( N_i \) marked items in all the \( N \) items and \( |q_i\rangle \) are the marked states associated with the marked items, and \( |q\rangle \) is the marked state that defines the problem Hamiltonian of the problem, the set \( \Pi_m \) only
contains the target state $|q\rangle$ of the problem. These sets satisfy the relation $\Pi_1 \supset \cdots \supset \Pi_{m-1} \supset \Pi_m$ with known sizes $N_1, \ldots, N_{m-1}, N_m = 1$, respectively. The goal is to find the marked item in the set $\Pi_m$.

To solve the hierarchical structured search problems by using this algorithm, we construct $H_0 = -|\psi_0\rangle\langle\psi_0|$ where $|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle$, and a sequence of intermediate Hamiltonians as

$$H_i = \frac{N_i}{N} H_0 + \left(1 - \frac{N_i}{N}\right) H_{P_i}, \quad i = 1, \ldots, m-1. \tag{5}$$

If the ratio $N_i/N_{i-1}$ $(i = 1, \ldots, m)$ are not exponentially small, where $N_0 = N$, then the conditions of our algorithm are satisfied. If $N_i$ are already known, the ground state eigenvalues of the intermediate Hamiltonians and the overlaps between the ground states of two adjacent Hamiltonians can be calculated analytically [2]. The hierarchical structured search problems can be solved efficiently with optimal runtime through the evolution path in Eq. (5) step by step.

III. APPLICATION OF THE ALGORITHM FOR THE HIDDEN SUBGROUP PROBLEMS

We now apply this algorithm for solving the HSP and problems that can be reduced to the HSP. The HSP encompass most of the quantum algorithms that are exponentially faster than their classical counterparts. The Deutsch-Josza problem, the Simon’s problem, the factoring problem, the discrete logarithm problem, the period-finding problem and the order-finding problem can be reduced to the Abelian HSP. The graph isomorphism problem, the discrete logarithm problem, the period-finding problem, the Simon’s problem, the factorization problem, the quantum black box for performing the unitary transform $U_f(g)|h\rangle = |g\rangle|h \oplus f(g)\rangle$, for $g \in G$, $h \in X$, and $\oplus$ a binary operation on $X$, using information gained from evaluation of $f$, find a generating set for $K$. The function $f$ hides the subgroup $K$ and has distinct values on different cosets of $K$.

For a finite group $G$, a classical algorithm calls a routine evaluating $f(g)$ once for each group element $g$, and determine $K$ with $|G|$ function calls. Quantum algorithms can reduce the time cost to $O(poly(log |G|))$ for the finite Abelian groups. Suppose the function $f$ maps the subgroup $K = S_1$ and its cosets $S_2, \ldots, S_M$ to distinct constants $h_1, \ldots, h_M \in X$, respectively, where $M = |G|/|K|$, the standard quantum algorithm for solving the finite Abelian HSP is first querying the function $f$ to prepare the coset state: $\frac{1}{\sqrt{|K|}} \sum_{i=1}^{M} |S_i\rangle\langle S_i|$, where $|S_i\rangle = \frac{1}{\sqrt{|K|}} \sum_{k_i \in S_i} |k_i\rangle$; then performing the quantum Fourier transform over $G$ to the coset state, measuring the resulting state, and repeat the whole process to output a generating set of the hidden subgroup $K$ from the measurement results [3]. The non-Abelian HSP is more difficult than the Abelian case, and there is no known efficiently quantum algorithm so far. In the following, we show that both the Abelian and the non-Abelian HSP can be reduced to the hierarchical structured search problems and they can be solved efficiently in the same way by using our algorithm.

The difficult case of HSP is when each coset contains polynomial number of group elements, i.e., $|K| \ll |G|$. If we obtain the state of the hidden subgroup $|S_i\rangle = \frac{1}{\sqrt{|K|}} \sum_{k_i \in K} |k_i\rangle$ that is mapped to the integer $h_i \in X$, then we can obtain the group elements of the subgroup by measuring the state. By using the oracle $O_f : |j\rangle|0\rangle \mapsto |j\rangle|f(j)\rangle$, where $|j\rangle$ is the state associated with the group element $g_j \in G$, the problem Hamiltonian of the HSP is:

$$H'_{P_i}|j\rangle = h_i|j\rangle, \quad j \in S_i, \quad i = 1, \ldots, M. \tag{6}$$

Without loss of generality, we assume that $h_i$ are integers with increasing order for $i = 1, \ldots, M$. In Fig. 1, we plot the spectrum of the HSP, i.e., the dimension of the cosets v.s. the function $f(g_j)$. To apply our algorithm for solving the HSP, we use $m = \log_2 M$ oracles to divide the group elements of $G$ according to their corresponding eigenvalues by using the method of bisection, and construct Hamiltonians $H_{P_i}$ as shown in Eq. (3), then we construct the intermediate Hamiltonians $H_i$ as shown in Eq. (5). We prepare a division set $\{v_1 = h_{M/2}, v_2 = h_{M/2^2}, \ldots, v_m = h_1\}$, and construct the Hamiltonians $H_{P_i}$ by using $m$ different oracles as:

$$H_{P_i}|k\rangle = \begin{cases} -1 \cdot |k\rangle, & \text{if } h_k \leq v_i \\ 0 \cdot |k\rangle, & \text{if } h_k > v_i \end{cases}, \quad i = 1, \ldots, m. \tag{7}$$

FIG. 1. Spectrum of the hidden subgroup problems.
This can be achieved by using an oracle that recognizes whether the integer \( h_k \) is larger or less than \( v_i \). It is a comparison logic circuit and can be implemented efficiently on a quantum computer \( \text{[6, p264] [11]} \).

The number of computational basis states associated with the integers that are less than or equal to \( v_i \) in the \( i \)th step is \( N_i \) as shown in Eq. (7), by using the method of bisection, the ratio \( N_i/N_{i-1} = 1/2 \). Therefore by constructing the intermediate Hamiltonians as shown in Eq. (5), the state of the hidden subgroup \( |S_i\rangle = \frac{1}{\sqrt{|K|}} \sum_{k_i \in K} |k_i\rangle \) can be obtained efficiently in \( m \) steps with evolution time \( t = \pi/(\sqrt{2}c) \) in each step. The total runtime of the algorithm for solving the HSP scales as \( O(\log |G|) \) by using \( m \) oracles.

This algorithm has been applied for solving the Deutsch-Jozsa problem \( \text{[2]} \). In the following we apply it for solving problems that can be reduced to the Abelian HSP, and the GI problem and the poly(\( n \))-unique SVP that can be reduced to the non-Abelian HSP.

### A. Simon’s problem

The Simon’s algorithm \( \text{[12]} \) is the first quantum algorithm that is demonstrated to be exponentially faster than any probabilistic classical algorithm in solving a black-box problem. In this problem \( \text{[12, 13]} \), there is an \( n \)-bit integer \( a \) such that for any two \( n \)-bit inputs \( i, j \), a black-box function \( f : \{0,1\}^n \rightarrow \{0,1\}^{n-1} \) outputs the integers \( f(i) = f(j) \) if and only if \( i \oplus j = a \), where \( \oplus \) denotes the bitwise XOR operation. The task is to find \( a \) by querying the function \( f \). It requires \( O(2^{n/2}) \) queries of \( f \) for classical algorithms to solve this problem, while the Simon’s algorithm solves this problem with \( O(n) \) queries of \( f \) on a quantum computer.

In this problem, the black box function represents a two-to-one mapping, two states \( |i\rangle, |j\rangle \) with \( i \oplus j = a \) form a set corresponds to an integer in \( \{0,1\}^{n-1} \). The period \( a \) can be calculated if we can obtain such a set deterministically and measure the state. The problem Hamiltonian for the Simon’s problem can be constructed by using the black-box function \( f \). Let \( h_k \) be the integer associated with the state \( |k\rangle \) \( (k = 0, \ldots, N-1, h_k \in [0,N/2-1]) \), then \( h_i = h_j \) if and only if \( i \oplus j = a \). We apply our algorithm for finding the set that contains two states associated with the integer \( h_k = 0 \), we prepare a division set \( \{v_1 = N/4-1, v_2 = N/8-1, \ldots, v_{n-2} = 1, v_{n-1} = 0\} \), and use \( m = \log (N/2) = n-1 \) oracles to construct Hamiltonians \( H_P \) as

\[
H_P |k\rangle = \begin{cases} 
-1 \cdot |k\rangle, & \text{if } h_k \leq v_i \\
0 \cdot |k\rangle, & \text{if } h_k > v_i 
\end{cases}, \quad i = 1, \ldots, n-1.  \tag{8}
\]

Then we construct a sequence of intermediate Hamiltonians \( H_i \) as shown in Eq. (5). Our algorithm can be performed efficiently for obtaining the state of the set associated with integer 0 in \( n-1 \) steps to find the period \( a \) directly with runtime scales as \( O(n) \) by querying \( n-1 \) different oracles.

### B. The factoring problem

Factoring an integer \( Z = x \cdot y \) on a quantum computer can achieve exponential speedup over the best known classical algorithm by using Shor’s algorithm \( \text{[14]} \). An integer \( a < Z \) and co-prime with \( Z \) is used in the algorithm, the order of \( a \) is defined as the smallest integer \( r \) that satisfies \( a^r = 1 \pmod{Z} \), which can be found efficiently through quantum Fourier transform. The factors of \( Z \) can be calculated as \( \gcd(a^{r/2} \pm 1, Z) \). The cost of the algorithm scales as \( O(L^3) \), where \( L = \lceil \log_2 Z \rceil \).

Our algorithm can also be applied for solving the factoring problem. The problem Hamiltonian for factoring an integer \( Z \) is defined as

\[
H_P |k\rangle = h_k |k\rangle = a^k \pmod{Z} |k\rangle, \quad k = 0, 1, \ldots, N-1, \tag{9}
\]

where \( N = 2^n \geq Z \) contains a few periods of the integer \( a \). The ground state eigenvalue of \( H_P \) is 1, and the corresponding eigenstate are in form of \( |p \cdot r\rangle \), where \( p = 0, 1, \ldots, \lfloor (N-1)/r \rfloor \). The order \( r \) of \( a \) can be determined if the ground state of the problem Hamiltonian is obtained, which is in form of \( \frac{1}{\sqrt{N/r}} \sum_{p=0}^{N/r} |p \cdot r\rangle \).

The eigenvalues of the problem Hamiltonian are integers and distribute uniformly in the period of \( a \), since \( k \) ranges from 0 to \( N-1 \) uniformly. The ratio between the degeneracies of any two energy levels of the problem Hamiltonian is about one. We construct a division set \( \{v_1 = \lceil \frac{Z}{2} \rceil, v_2 = \lceil \frac{Z}{4} \rceil, \ldots, v_L = \lceil \frac{Z}{2^L} \rceil = 1\} \) (it could be \( v_{L-1} = \lceil \frac{Z}{2^L} \rceil = 1 \), then the division set contains \( L-1 \) elements, for convenience, we assume there are \( L \) elements). The Hamiltonians \( H_P \) can be constructed by using \( L \) oracles as

\[
H_P |k\rangle = \begin{cases} 
-1 \cdot |k\rangle, & \text{if } h_k \leq v_i \\
0 \cdot |k\rangle, & \text{if } h_k > v_i, \quad i = 1, \ldots, L.  \tag{10}
\end{cases}
\]

Then we construct a sequence of intermediate Hamiltonians as shown in Eq. (5) and let \( H_P = H_{P_L} \). The ratio \( N_i/N_{i-1} \) for the factoring problem Hamiltonian is about \( 1/2 \) in each step of the algorithm, therefore the total runtime of the algorithm for obtaining the ground state of the problem Hamiltonian \( H_P \) is proportional to the number of steps \( L \) of the algorithm, and scales as \( O(\log N) \).

The order \( r \) of \( a \) can be obtained by measuring the ground state of the problem Hamiltonian, thus solving the factoring problem efficiently in \( L \) steps.

### C. The discrete logarithm problem

The discrete logarithm problem is as follows \( \text{[3]} \): for a cyclic group \( G \) generated by an element \( g \), given an
element \( x \in G \), find the discrete logarithm of \( x \) with respect to \( g \), \( \log_g x \), which is the smallest nonnegative integer \( j \) such that \( g^j = x \).

The discrete logarithm problem can be reduced to a period-finding problem on the additive group of pairs of integers modulo \( N \). Consider a function \( f(x_1, x_2) = a^{x_1} + x_2 (\text{mod } N) \), where all the variables are integers, and \( r \) is the smallest positive integer for which \( a^r \equiv 1 \pmod{N} \). This function is \( 2 \)-tuple periodic, since \( f(x_1 + l, x_2 - l) = f(x_1, x_2) \). Given \( a \) and \( b = a^r \), the discrete logarithm problem is to determine the integer \( s \). By applying the quantum order-finding algorithm \( [6] \), one can obtain \( r \) with one query of a quantum black box \( U \) that performs the unitary transform \( U|x_1⟩|x_2⟩|y⟩ \rightarrow |x_1⟩|x_2⟩|y + f(x_1, x_2)⟩ \), and \( O\left( \log^2 \frac{N}{r} \right) \) other operations. Then the integer \( s \) can be obtained from \( r \), thus solving the problem.

Our algorithm can also be applied for solving the discrete logarithm problem. For the periodic function \( f(x_1, x_2) \), we define the problem Hamiltonian as

\[
H'_p|x_1⟩|x_2⟩ = \sum_{x_1, x_2} b_{x_1, x_2}|x_1⟩|x_2⟩ = \sum_{x_1, x_2} a^{x_1} + x_2 (\text{mod } N)|x_1⟩|x_2⟩ = a^{x_1} + x_2 (\text{mod } N)|x_1⟩|x_2⟩, \tag{11}
\]

where \( x_1, x_2 = 0, \ldots, N - 1 \). The ground state eigenvalue of \( H'_p \) is 1, and the corresponding ground state is in form of \( |x⟩ = \frac{\sum_{x_1, x_2} b_{x_1, x_2} |x_1⟩|x_2⟩}{\sqrt{\sum_{x_1, x_2} b_{x_1, x_2}}} \), where \( k = 0, \ldots, \lfloor \frac{(N - 1)}{r} \rfloor \). The eigenvalues of \( H'_p \) distribute uniformly in the period of \( a \), since \( x_1 \) and \( x_2 \) range from 0 to \( N - 1 \) uniformly. We construct a division set as shown in solving the factoring problem by using the method of bisection, and construct the Hamiltonians \( H_p \) and a sequence of intermediate Hamiltonians \( H_1 \) as shown in Eqs. (3) and (5), respectively. The ground state of \( H_p \) can be obtained by using our algorithm in \( \log^2 \frac{N}{r} \) steps. Then by measuring the qubits to obtain states \( |x_1⟩ \) and \( |x_2⟩ \), one can calculate the integer \( s \), thus solving the discrete logarithm problem.

D. The period-finding problem and the order-finding problem

The period-finding problem can be described as follows [6]: suppose \( f \) is a periodic function producing a single bit as output and such that \( f(x + r) = f(x) \), for some unknown \( 0 < r < 2^L \), where \( x, r \in \{0, 1, \ldots\} \). Given a quantum black box \( U \) that performs the unitary transform \( U|x⟩|y⟩ \rightarrow |x⟩|y + f(x)⟩ \), how many black box queries and other operations are required to determine \( r \)?

Quantum algorithm solves this problem using one query and \( O(L^2) \) other operations [6]. For the period-finding problem, the states \( |x + kr⟩ \), \( k = 0, \ldots, 2^L/r \) have the same function value, its spectrum is the same as shown in Fig. 1. Therefore it can be solved by obtaining the states \( |x + kr⟩ \) that correspond to the ground state eigenvalue of the function \( f \). Our algorithm can be applied to solve this problem with \( O(\log N) \) queries of \( f \).

The order-finding problem is [6]: for positive integers \( y \) and \( N, y < N \), that have no common factors, the order of \( y \) modulo \( N \) is defined to be the least positive integer \( r \), such that \( y^r = 1 \pmod{N} \). The problem is to determine the order for some specified \( y \) and \( N \). Order-finding is believed to be a hard problem on a classical computer, and is used as the basis for the RSA crypto system. As shown in solving the factoring problem, the order-finding problem can be solved efficiently by using our algorithm in \( L \) steps, with \( L = \log_2 N \) being the number of bits needed to specify \( N \).

E. The graph isomorphism problem

The GI problem is considered as one of the few natural problems in the complexity class of NP that could be classified as neither NP-complete nor P. It is an important problem in mathematics and computer science since 1970s, for a review on the GI problem, see Ref. [16] and references therein. In a recent work [17], it has been shown that the GI problem is solvable in quasi-polynomial time by using a classical algorithm.

The GI problem can be described as follows: given two undirected graphs \( A_1(V_1, E_1) \) and \( A_2(V_2, E_2) \) with vertex sets \( V_1, V_2 \), and edge sets \( E_1, E_2 \), respectively, and \( |V_1| = |V_2| = n \), the graph \( A_1 \) is isomorphic to \( A_2 \), if there exists a bijection \( \phi : V_1 \rightarrow V_2 \) such that for all \( x, y \in V_1 \), \( (x, y) \in E_1 \) if and only if \( (\phi x, \phi y) \in E_2 \). The GI problem can be reduced to the graph automorphism problem, which is a relabeling of vertices of a graph that preserves the graph edges. The graph automorphism problem for a graph \( A(V, E) \) is to find a set of generators for the group of automorphisms \( Aut(A) \) of \( A \).

The following reduction of the GI problem to the graph automorphism problem is due to Jozsa [18]. We construct a graph \( A \) as the disjoint union of the graphs \( A_1 \) and \( A_2 \), having \( 2n \) vertices labeled \( 1, 2, \ldots, n, n + 1, \ldots, 2n \) where \( 1, 2, \ldots, n \) label \( A_1 \) and \( n + 1, \ldots, 2n \) label \( A_2 \), the automorphism group \( K = Aut(A) \) which embeds into the symmetric group \( S_{2n} \). Any automorphism of \( A \) must either permute the vertices of \( A_1 \) and \( A_2 \) separately, or swap their vertices entirely. Let \( Q \) denotes the group \( S_n \times S_n \), which contains all permutations that map \( A_1 \) and \( A_2 \) into themselves, separately, and let \( \sigma = (1 \ 2 \ n + 1 \ 2n + 2 \ \ldots) \) be the permutation of \( 1, 2, \ldots, 2n \) that swaps the vertices of \( A_1 \) and \( A_2 \) in their listed order. The automorphism group \( K \) is a subgroup of the group \( Y = Q \cup \sigma Q \). If \( A_1 \) and \( A_2 \) are not isomorphic, then \( K = Aut(A) \) lies entirely in \( Q \). Otherwise, if \( A_1 \) and \( A_2 \) are isomorphic, then exactly half of the members of \( K \) are in \( Q \), and the other half are in \( \sigma Q \). We can easily check if an element \( \pi \) of \( K \) lies in \( Q \) or \( \sigma Q \) by evaluating \( \pi(1) \). So if we randomly sample elements of \( K \), we can determine if \( A_1 \) and \( A_2 \) are isomorphic with high probability. Then solving the HSP on the symmet-
ric group leads to a solution to the GI problem \[19\]. For graph \(A\), the automorphism group \(K\) is the hidden subgroup of the group \(Y\). The function \(f: Y \rightarrow X\), where \(X\) is the set of graphs created by permutations of \(A\) by \(f(\pi) = \pi A\). And \(f\) is constant on the cosets of \(K\) in \(Y\), it separates the cosets of \(K\), and is efficiently computable. The GI problem is reduced to a non-Abelian HSP where \(f\) hides the automorphism group \(K\) of graph \(A\).

Based on the above analysis, if we obtain the state of the hidden subgroup \(K\), then we can check if an element \(\pi\) of \(K\) lies in \(Q\) or \(\sigma Q\) by evaluating \(\pi(1)\) to determine if \(A_1\) and \(A_2\) are isomorphic with high probability. Our algorithm can be applied to obtain the state of the hidden subgroup \(K\) efficiently.

The group \(Y\) has \(|Y| = 2(n!)^2\) group elements. By applying the hidden function \(f\), the group elements of \(Y\) are mapped to at most \(2(n!)^2\) distinct integers from 1 to \(2(n!)^2\). By using the oracle \(O_f: |j\rangle \mapsto |j\rangle f(j)\), where the state \(|j\rangle\) is associated with a permutation \(\pi_j \in Y\), the problem Hamiltonian of the GI problem is:

\[ H_P' |j\rangle = h_j |j\rangle, \quad j = 1, \ldots, 2(n!)^2. \]  

We use a division set \(\{v_1 = (n!)^2, v_2 = \frac{v_1}{2}, \ldots, v_m = \frac{v_{m-1}}{2} = 1\}\) to construct \(m = \log_2 \left(\frac{2(n!)^2}{2}\right)\) oracles to run our algorithm step by step to find the state of the hidden subgroup. We construct the Hamiltonians \(H_P\) by using the above oracles as:

\[ H_P |k\rangle = \begin{cases} -1 |k\rangle, & \text{if } h_k \leq v_i \\ 0 |k\rangle, & \text{if } h_k > v_i \end{cases}, \quad i = 1, \ldots, m. \]  

Then we construct a sequence of intermediate Hamiltonians as shown in Eq. (5), let \(H_P = H_P^{m}\) and run the algorithm by setting the initial state as \(|v_0\rangle = \frac{1}{\sqrt{2}} \sum_{j=0}^{2(n!)^2-1} |j\rangle\). The ratio \(N_i/N_{i-1}\) for this problem is either 1 or 1/2 in each step. The state of the hidden subgroup \(|K\rangle = \frac{1}{\sqrt{|K|}} \sum_{k \in K} |k\rangle\) can be obtained in \(m\) steps with finite evolution time in each step. The total runtime of the algorithm for obtaining the ground state of the problem Hamiltonian \(H_P\) is proportional to \(m\). By performing measurement on the state \(|K\rangle\), we obtain an element \(\pi\) in the subgroup \(K\), then we can evaluate \(\pi(1)\) to determine if \(A_1\) and \(A_2\) are isomorphic.

F. The poly(\(n\))-unique shortest vector problem

Another set of the non-Abelian HSP is the lattice problems. An \(n\)-dimensional point lattice is a discrete subset of \(\mathbb{R}^n\) closed under addition and subtraction. It is generated by a set of integer linear combination of \(n\) linearly independent vectors \(b_1, \ldots, b_n\) as:

\[ B = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z} \right\}, \]  

where \(\mathbb{Z}\) denotes the integer set. The shortest vector problem is: given a lattice described by the basis vectors, find the shortest nonzero vector in the lattice.

The SVP is extremely difficult in \(n\) dimensions, all known algorithms for the SVP require exponential time in \(n\). Approximate solutions have been developed, i.e., finding some vector that is not too much larger than the shortest one. A vector is \(f(n)\)-unique if it is a factor of \(f(n)\) shorter than all other nonparallel vectors. Approximating the shortest vector to within a constant, that is, the \(O(1)\)-unique SVP is known to be NP-hard under a randomized reduction \[20\]. The \(O(2^n)\)-unique SVP is solvable in polynomial time by applying the LLL algorithm \[21\]. The decision version of the poly(\(n\))-unique SVP is known to be in NP and coNP \[22\ \[23\].

It has been found \[24\] that the poly(\(n\))-unique SVP can be reduced to the two-point problem on lattice, and the two-point problem can be reduced to the dihedral coset problem, which can be reduced to the dihedral HSP. Therefore an efficient quantum algorithm for the HSP of the dihedral group \(D_N\) will provide an efficient way of solving the poly(\(n\))-unique SVP.

The dihedral group \(D_N\) has \(2N\) elements, it is a group of symmetries of an \(n\)-sided regular polygon. There are \(2N\) ways one can apply rotations or reflections in a distinct way. Every element of \(D_N\) can be represented as a tuple \((r_1, r_2)\) where \(r_1 \in \{0, 1\}\) representing the number of reflections and \(0 \leq r_2 < N\) representing the number of rotations. The dihedral coset problem is to find a constant integer \(l\), given a collection of states in the form

\[ \frac{1}{\sqrt{2}} (|0\rangle |x\rangle + |1\rangle (x + l) \mod N) \]  

As shown in \[24\], the dihedral HSP can be reduced to the case where the subgroup is of the form \(K = \{(0,0), (1,l)\}\). A function \(f\) is defined to hide the subgroup \(K\) by mapping \(K\) to a constant (say 0), and each of the other \(N-1\) cosets of \(K\) to a distinct integer. The dihedral group \(D_N\) has exponential number \((\log N)\) of subgroups of order two, and is infeasible to solve by classical algorithms. While the dihedral HSP has the same structure as that of the Simon's problem in our algorithm. And the state of the hidden subgroup \(K\) can be obtained by applying our algorithm in runtime \(O(\log N)\) in the same way as that of in solving the Simon’s problem. The constant \(l\) can be determined by measuring the state of the hidden subgroup \(K\), and solving the dihedral coset problem, thus obtaining the solution to the poly(\(n\))-unique SVP.

IV. DISCUSSION

In this work, we study the spectrum structure of the hidden subgroup problems, and find that these problems can be reduced to a type search problems with a hierarchical structure, which can be solved efficiently through a multistep quantum computation process. Therefore the
HSP can be solved efficiently in the same way in polynomial time step by step. An oracle is used in each step of the algorithm to reduce the search space of the problem in a finite rate. Our algorithm shows that a multistep quantum computation process by using a number of oracles can be more efficient than a quantum computation process of using just one oracle.

We have shown that the HSP can be reduced to the hierarchical structured search problems, some problems in the HSP, e.g. the discrete logarithm problem and the GI problem, are believed to be in the complexity class of NP-intermediate, i.e. problems that are in neither the class of P nor the class of NP-complete, assuming P≠NP. It is likely that the hierarchical structured search problems also belong to the complexity class of NP-intermediate, we leave this as an open question for future study.

ACKNOWLEDGMENTS

This work was supported by the Fundamental Research Funds for the Central Universities (Grant No. 11913291000022), and the Natural Science Fundamental Research Program of Shaanxi Province of China (Grant No. 2022JM-021).

[1] E. Farhi, J. Goldstone, and S. Gutmann, A numerical study of the performance of a quantum adiabatic evolution algorithm for satisfiability, arXiv: quant-ph/0007071.
[2] H. Wang, S. Yu and H. Xiang, Efficient quantum algorithm for solving structured problems via multistep quantum computation, Phys. Rev. Res. 5, L012004 (2023).
[3] H. Wang, Quantum algorithm for obtaining the eigenstates of a physical system, Phys. Rev. A 93, 052334 (2016).
[4] Z. Li, X. Liu, H. Wang, S. Ashhab, J. Cui, H. Chen, X. Peng, and J. Du, Quantum Simulation of Resonant Transitions for Solving the Eigenproblem of an Effective Water Hamiltonian, Phys. Rev. Lett. 122, 090504 (2019).
[5] A. M. Childs and W. van Dam, Quantum algorithms for algebraic problems, Rev. Mod. Phys. 82, 1 (2010).
[6] M. A. Nielsen and I. L. Chuang, Quantum computation and quantum information (Cambridge Univ. Press, Cambridge, England, 2000).
[7] W. K. Wootters and W. H. Zurek, A single quantum cannot be cloned, Nature 299, 802 (1982).
[8] D. Dieks, Communication by EPR devices, Phys. Lett. A 92, 271 (1982).
[9] C. Dürr and P. Høier, A quantum algorithm for finding the minimum, arXiv: quant-ph/9607014.
[10] W. P. Baritompa, D. W. Bulger and G. R. Wood, Grover’s quantum algorithm applied to global optimization, SIAM J. Optim. 15, 1170 (2005).
[11] J. M. Martyn, Z. M. Rossi, A. K. Tan and I. L. Chuang, Grand unification of quantum algorithms, PRX Quantum, 2, 040203 (2021).
[12] D. R. Simon, On the power of quantum computation, in Proceedings of the 35th Annual Symposium on Foundations of Computer Science, edited by S. Goldwasser (IEEE) 1994, pp. 116–123.
[13] I. Hen, Period finding with adiabatic quantum computation, EPL, 105 50005 (2014).
[14] P. Shor, Algorithms for quantum computation: discrete logarithms and factoring, Proc. 35th Ann. Symp. on Found. of Comp. Sci., 124-134 (IEEE Comp. Soc. Press, Los Alamitos, CA, 1994).
[15] P. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM J. Comput. 26, 1484 (1997), preliminary version in FOCS 1994.
[16] M. Grohe and P. Schweitzer, The graph isomorphism, Communications of the ACM, 63, 128 (2020).
[17] L. Babai, Graph isomorphism in quasipolynomial time. In Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC’16, 2016), 684–697.
[18] R. Jozsa, Quantum factoring, discrete logarithms, and the hidden subgroup problem, Computing in Science and Engineering, 3:34-43 (2001).
[19] K. Johannes, S. Uwe, and T. Jacobo, The graph isomorphism problem: Its structural complexity, Birkhäuser Boston, Boston, 1993.
[20] D. Micciancio, The shortest vector in a lattice is hard to approximate to within some constant, SIAM J. Comput., 30:2008 (2001).
[21] A. K. Lenstra, H. W. Lenstra, and L. Lovász, Factoring polynomials with rational coefficients, Mathematische Annalen, 261:515-534 (1982).
[22] D. Aharonov and O. Regev, Lattice problems in NP ∩ coNP. J. ACM, 52:749 (2005).
[23] V. Lyubashevsky and D. Micciancio, On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology, pages 577-594, Berlin, Heidelberg, 2009. Springer-Verlag.
[24] O. Regev, Quantum computation and lattice problems, SIAM J. Comput., 33, 738-760 (2004).