

Quantum algorithm for minimum dominating set problemwith circuit design

Haoying Zhang(张皓颖), Shaoxuan Wang(王绍轩), Xinjian Liu(刘新建), Yingtong Shen(沈颖童), and Yukun Wang(王玉坤)

Citation: Chin. Phys. B, 2024, 33 (**2**): 020310. DOI: 10.1088/1674-1056/ad02e5
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Threshold-independent method for single-shot readout of spin qubits in semiconductor quantum dots

Rui-Zi Hu(胡睿梓), Sheng-Kai Zhu(祝圣凯), Xin Zhang(张鑫), Yuan Zhou(周圆), Ming Ni(倪铭), Rong-Long Ma(马荣龙), Gang Luo(罗刚), Zhen-Zhen Kong(孔真真), Gui-Lei Wang(王桂磊), Gang Cao(曹刚), Hai-Ou Li(李海欧), and Guo-Ping Guo(郭国平)
Chin. Phys. B, 2024, 33 (**1**): 010304. DOI: 10.1088/1674-1056/ace3a9

A quantum algorithm for Toeplitz matrix-vector multiplication

Shang Gao(高尚) and Yu-Guang Yang(杨宇光)
Chin. Phys. B, 2023, 32 (**10**): 100309. DOI: 10.1088/1674-1056/acb914

Variational quantum simulation of the quantum critical regime

Zhi-Quan Shi(石志全), Xu-Dan Xie(谢旭丹), and Dan-Bo Zhang(张旦波)
Chin. Phys. B, 2023, 32 (**8**): 080305. DOI: 10.1088/1674-1056/accb43

Variational quantum semi-supervised classifier based on label propagation

Yan-Yan Hou(侯艳艳), Jian Li(李剑), Xiu-Bo Chen(陈秀波), and Chong-Qiang Ye(叶崇强)
Chin. Phys. B, 2023, 32 (**7**): 070309. DOI: 10.1088/1674-1056/acb9fb

Variational quantum simulation of thermal statistical states on a superconducting quantum processer

Xue-Yi Guo(郭学仪), Shang-Shu Li(李尚书), Xiao Xiao(效骁), Zhong-Cheng Xiang(相忠诚), Zi-Yong Ge(葛自勇), He-Kang Li(李贺康), Peng-Tao Song(宋鹏涛), Yi Peng(彭益), Zhan Wang(王战), Kai Xu(许凯), Pan Zhang(张潘), Lei Wang(王磊), Dong-Ning Zheng(郑东宁), and Heng Fan(范桁)
Chin. Phys. B, 2023, 32 (**1**): 010307. DOI: 10.1088/1674-1056/aca7f3

Quantum algorithm for minimum dominating set problem with circuit design

Haoying Zhang(张皓颖)[1], Shaoxuan Wang(王绍轩)[1], Xinjian Liu(刘新建)[1], Yingtong Shen(沈颖童)[1], and Yukun Wang(王玉坤)[1,2,†]

[1] *Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum, Beijing* 102249, *China*
[2] *State Key Laboratory of Cryptology, P.O. Box* 5159, *Beijing* 100878, *China*

(Received 14 June 2023; revised manuscript received 3 October 2023; accepted manuscript online 13 October 2023)

Using quantum algorithms to solve various problems has attracted widespread attention with the development of quantum computing. Researchers are particularly interested in using the acceleration properties of quantum algorithms to solve NP-complete problems. This paper focuses on the well-known NP-complete problem of finding the minimum dominating set in undirected graphs. To expedite the search process, a quantum algorithm employing Grover's search is proposed. However, a challenge arises from the unknown number of solutions for the minimum dominating set, rendering direct usage of original Grover's search impossible. Thus, a swap test method is introduced to ascertain the number of iterations required. The oracle, diffusion operators, and swap test are designed with achievable quantum gates. The query complexity is $O(1.414^n)$ and the space complexity is $O(n)$. To validate the proposed approach, qiskit software package is employed to simulate the quantum circuit, yielding the anticipated results.

Keywords: quantum algorithm, circuit design, minimum dominating set

PACS: 03.67.Ac **DOI:** 10.1088/1674-1056/ad02e5

1. Introduction

Quantum computing is an interdisciplinary field that combines principles in modern physics and computer science. In recent years, quantum algorithms have received considerable attention from researchers,[1,2] attracted by their unique properties such as superposition,[3] interference,[4] and entanglement,[5] which enable exponential or square order speedups compared to classical algorithms. Numerous studies have explored quantum algorithms,[1,2,6–15] highlighting their potential for solving various computational problems.

The minimum dominating set problem, particularly on an undirected graph, is a well-known problem that belongs to the class of problems known to be NP-complete,[16] where NP stands for non-deterministic polynomial. The concept of a dominating set has applications in various real-life scenarios, offering a versatile mapping to a range of problems. Examples include the optimization of school bus routing, wireless sensor network management, urban traffic route planning, and several other domains.[17] The problem seeks to find a subset of vertices in a graph that dominates all other vertices. In other words, each vertex in the graph must either belong to the dominating set or be adjacent to at least one vertex in the set. Various classical algorithms, such as greedy algorithm[18] and genetic algorithm,[19] have been proposed to solve this problem. However, these approaches often yield approximate solutions and are highly dependent on the chosen strategy. Exact solution algorithms also exist,[20–26] but they suffer from high computational complexity.

Grover's search algorithm has emerged as one of the most widely studied quantum algorithms, with applications in diverse problem domains.[27–30] This algorithm employs Grover iterations, which comprise two essential components: the oracle and the diffusion operator. The oracle identifies the target states, while the diffusion operator amplifies the amplitudes of these states. By performing measurements, the algorithm can identify the target states representing solutions to the problem with a high probability. Compared to classical algorithms, Grover's algorithm offers a computational advantage by requiring only $O(\sqrt{N})$ iterations to find a solution in an unsorted database of size N, significantly outperforming the lower bound of $O(N)$ evaluations required by classical methods.

In the field of NP-complete problems, Grover's algorithm is a popular solution that can solve many problems, including the graph coloring problem,[10–13] the shortest path problem,[14] and the 0–1 knapsack problem.[15] Therefore, leveraging the accelerated properties of quantum algorithms, applying Grover's search algorithm to the dominating set problem is an attractive proposition. However, a challenge arises in the minimum dominating set problem due to the unknown number of solutions, making direct use of original Grover's search algorithm infeasible. In recent years, there has been substantial research focusing on enhancing Grover's algorithm for various applications.[31–37] These proposed methods have limited efficacy when it comes to solving the minimum dominating set problem. For instance, the approaches

†Corresponding author. E-mail: wykun06@gmail.com
© 2024 Chinese Physical Society and IOP Publishing Ltd

described in Ref. [31] can only identify the dominating set and do not provide a guarantee of finding the minor dominating set. Moreover, the design complexity associated with the phase gate in Ref. [32] presents challenges. As an alternative, Aaronson and Rall presented an approximate counting algorithm in Ref. [33], which estimates the number of target solutions within a set without relying on the quantum fourier transform, resulting in a relatively complex implementation. Therefore, a universal method is needed to estimate the number of iterations for Grover's algorithm.

To address this challenge, we propose a swap test method to estimate the number of iterations required by calculating the fidelity between the initial state and the state of the algorithm after one iteration. By exploiting fidelity information, we can determine the number of iterations for Grover's algorithm. Using the swap test and Grover's algorithm, we design a quantum algorithm for the minor dominated set and subsequently obtain the minimum dominated set through classical post processing. Our proposed method ensures the attainment of an approximate exact solution and facilitates a simple circuit design. We outline the algorithm framework and systematically design the oracle circuit, diffusion operators, and swap test circuit, employing achievable quantum gates. The query complexity and qubit cost of this method are $O(1.414^n)$ and $O(n)$ when the number of vertices in a graph is n.

The structure of this paper is as follows. Section 2 provides an overview of quantum circuits, Grover's algorithm, and dominating sets. In Section 3, we present a detailed description and performance analysis of our algorithm circuit. We implement the method using qiskit and demonstrate its effectiveness with two examples in Section 4. Finally, Section 5 concludes the paper.

2. Preliminaries

In this section, we provide readers with some background knowledge to better understand our method. We describe qubits and quantum gates[38] in the context of quantum circuits, then provide a concise overview of Grover's algorithm, and the dominating set problem in graph theory.

2.1. Quantum circuit

In quantum computing, all algorithms can be realized by quantum circuits which are composed of qubits and quantum gates.

Qubits The fundamental unit of a quantum circuit is the quantum bit, known as a qubit, which is the quantum counterpart of classical bits used in classical computing. In the realm of physics, a qubit can be realized as a superposition of any two distinguishable quantum states, typically denoted as $|0\rangle$ and $|1\rangle$. These quantum states can be physically represented based on the properties of certain physical systems, such as the spin of an electron or the energy level state of an atom. In a quantum circuit, a qubit can exist in a superposition of multiple states, exemplified as $\alpha|0\rangle+\beta|1\rangle$, where α and β are complex numbers, representing the probability amplitudes of the respective states. This property enables quantum circuits to simultaneously process multiple possibilities, potentially conferring a computational advantage. There are two types of qubits, the first type is data qubits which are used to encode the input or output of a quantum algorithm. The other is the ancilla qubits that are used to store interim or temporary results during the execution of the algorithm.

Quantum gates Quantum gates are employed to enact all operations within a quantum circuit, serving as operations or evolutions within physical systems capable of modifying the state of qubits. Distinct quantum gates exert diverse influences on qubits. In our algorithm, we utilize several specific quantum gates, including the Hadamard (H) gate, the controlled-NOT (CNOT) gate, the Toffoli gate, and the multiple control Toffoli (MCT) gate.

- **H gate** The Hadamard gate is employed to generate superposition states of qubits, transforming the $|0\rangle$ state into $|+\rangle$ ($H|0\rangle=|+\rangle$) and the $|1\rangle$ state into $|-\rangle$ ($H|1\rangle=|-\rangle$), where $|+\rangle=\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|-\rangle=\frac{|0\rangle-|1\rangle}{\sqrt{2}}$.
- **CNOT gate** It is a two-qubit gate where the upper qubit acts as the control qubit and the lower qubit functions as the target qubit. When the gate is applied, it transforms the tensor product state of the two qubits, denoted as $|a\rangle\otimes|b\rangle$ into $|a\rangle\otimes|a\oplus b\rangle$, where the $\otimes$ is used to represent the tensor product in quantum mechanics, indicating the combined state of two or more qubits and the $\oplus$ represents the bitwise XOR (exclusive OR) operation between bit values, indicating the result of applying XOR to the binary values of the bits.
- **Toffoli gate** This is a three-qubit quantum gate. There are two control qubits and a single target qubit. When it is applied, $|a_1a_2\rangle\otimes|b\rangle$ can be transformed to $|a_1a_2\rangle\otimes|a_1a_2\oplus b\rangle$.
- **MCT gate** This gate is similar to the Toffoli gate, which is an n-qubit quantum gate. By applying it, $|a_1\ldots a_n\rangle\otimes|b\rangle$ can be transformed to $|a_1\ldots a_n\rangle\otimes|a_1\ldots a_n\oplus b\rangle$.

Here, $a,b,a_i\in\{0,1\}$ represent binary values. We collectively refer to the CNOT gate, Toffoli gate, and MCT gate as controlled gates. In this context, a visual notation is employed, utilizing a black dot to signify that when the control qubit is in the state $|1\rangle$, it induces a change in the target qubit. Conversely, for ease of representation, a white dot is used to signify that when the control qubit is in the state $|0\rangle$, it likewise influences the state of the target qubit. In current quantum devices, basic quantum logic gates, such as single-qubit gates and CNOT

gates, are implemented using control techniques to manipulate individual qubits and establish interactions between them. These gates are realized on various quantum platforms, including superconducting qubits, trapped ions, and hybrid systems combining different types of qubits.

2.2. Grover's algorithm

Currently, Grover's algorithm has been implemented on various quantum computing platforms including IBM Quantum Experience, Rigetti Forest, and Microsoft Quantum Development Kit. These platforms provide programming languages and tools for users to utilize Grover's algorithm for tasks such as unstructured database search. Except for the simulation platforms, there have been several experimental demonstrations of Grover's algorithm on different quantum hardware platforms, such as superconducting qubits and trapped ions. Although these implementations are still limited in terms of the number of qubits and the level of noise, they represent significant progress towards achieving practical applications of Grover's algorithm on quantum hardware.

The quantum circuit framework employed in Grover's algorithm is visually depicted in Fig. 1. It comprises two fundamental components: the oracle and the diffusion operator. The oracle plays a pivotal role in this quantum search algorithm as it is responsible for altering the phase of the target states. It is worth noting that different problem instances necessitate distinct oracle configurations, tailoring the algorithm to specific problem domains. The diffusion operator, alternatively known as the inversion about the average operator, serves as a critical amplification step within Grover's algorithm. Its primary function is to enhance the amplitude of the target states, thereby elevating the likelihood of measuring the desired solution. This enhancement is achieved through a process that reflects the average amplitude across all states, accentuating those that deviate from this average. The initial setup involves the utilization of the first n qubits as data qubits, responsible for encoding the problem's solution space. Meanwhile, the final qubit, known as the ancilla qubit, is dedicated to effectuating phase alterations within the target states. Grover's algorithm encompasses four distinct stages, namely, initialization, Hadamard transformation, oracle, and diffusion operator.

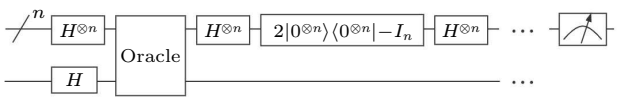


Fig. 1. Generalized circuit of Grover's algorithm.

Initialization In the initialization stage, the data qubits are initialized in the state $|0\rangle$. The ancilla qubit, on the other hand, is initialized in the state $|1\rangle$, preparing the system for computation.

Hadamard transformation Hadamard operations are applied to both the data and ancilla qubits, the operation facilitates the creation of superposition states, endowing all possible solutions with an equal probability amplitude.

$$|X\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle, \tag{1}$$

and the ancilla qubit is transformed from $|1\rangle$ to $|-\rangle$.

Oracle The oracle, a problem-specific gate, is pivotal in marking the correct solution with a negative phase. It perturbs the phases of the target states in a manner tailored to the problem's nature. Let the solution set of the problem be denoted as S. The oracle can be represented as a function f defined as

$$f(x) = \begin{cases} 1, & \text{if} \quad x \in S, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Thus, the effect of the oracle can be defined as

$$|x\rangle_n \otimes |q\rangle \rightarrow |x\rangle_n \otimes |q \oplus f(x)\rangle, \tag{3}$$

where $|x\rangle_n$ represents the data qubits, and $|q\rangle$ represents the ancilla qubit.

Diffusion operator The diffusion operator increases the probability amplitude of the correct solution by adjusting the phase of the target state, which effectiveness arises from its underlying quantum mechanical principles. It is fundamentally based on the concept of quantum interference. In quantum systems, probabilities are determined by the complex amplitudes associated with each possible outcome. These amplitudes can interfere constructively or destructively. If a quantum state is viewed as a vector in a high-dimensional space, the diffusion operator will rotate this vector closer to the target state.

As we can see in Fig. 1, the operation described as

$$H^{\otimes n}(2|0^{\otimes n}\rangle\langle 0^{\otimes n}| - I_n)H^{\otimes n} \tag{4}$$

is the diffusion operator. During this phase, the amplitude of the marked states is enhanced, while the probability amplitudes of the unmarked states are reduced. As a result, when the data qubits are measured, the marked states are obtained with a significantly higher probability. This enables an efficient identification and extraction of the desired marked states from the superposition of all possible states.

Grover's algorithm is comprised of repetitively executing the oracle and diffusion operators, collectively referred to as iterations. In practical problem-solving scenarios, it becomes necessary to determine the optimal number of iterations for Grover's algorithm. Assuming a search space of size N with M possible solutions, it has been shown that after an order of $O(\sqrt{N/M})$ iterations, there exists a high probability of obtaining one of the solution states. In essence, the query complexity of Grover's algorithm is estimated to be $O(\sqrt{N/M})$, providing a quadratic improvement over the classical approach that requires $O(N/M)$ oracle calls.

2.3. Dominating set

Giving an undirected graph $G = (V,E)$, there is a vertex subset $S \subseteq V$, with $S \neq \varnothing$. For $\forall\, v \in V$, either v belongs to S, or at least one adjacent vertex of v belongs to S. Such an S is called a dominating set of G. The vertices which belong to S are called the dominant vertex, and other vertices are called the non-dominant vertex. The number of dominant vertices for the dominating set S is defined as $\gamma(S)$. If any true subsets of the dominating set S are non-dominating, S is called a minor dominating set. The sets with the smallest $\gamma(S)$ in all minor dominating sets are called minimum dominating sets. The minimum dominating and minor dominating sets of a graph are not unique, but the $\gamma(S_i)$ of all minimum dominating sets is equal. For a more comprehensive description of the dominating set problem, we refer the reader to the books.[39,40] Figure 2 illustrates an example with three vertices, where the dominating sets can be identified as $\{A\},\{A,C\},\{A,B\},\{B,C\},\{A,B,C\}$, the minor dominating sets are $\{A\}$ and $\{B,C\}$, and the final minimum dominating set is $\{A\}$.

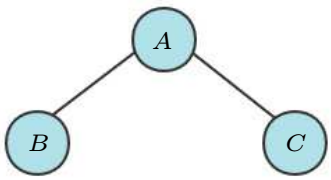


Fig. 2. An undirected graph $G = (V,E)$ with three vertices.

3. The proposed approach for the minimum dominating set problem

This section presents the algorithmic implementation for identifying the minimum dominating set in undirected graphs. Figure 3 illustrates the step-by-step process of the algorithm, which aims to leverage quantum algorithms to accelerate the search for minor dominating sets. Through the measurement of the quantum circuit, a set of quantum states describing the states of the vertices (whether they are dominant or non-dominant) can be obtained. Within this set, the minimum dominating set corresponds to the smallest number of dominant vertices. However, the measured result represents a combination of minor dominating sets, necessitating classical post-processing. The overall idea of the quantum algorithm is shown in Algorithm 1, which shows how the search algorithm finds the minor dominating set. In the subsequent sections, we introduce Algorithm 1 in detail.

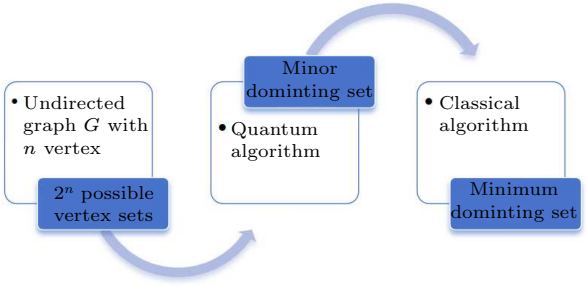


Fig. 3. The process of finding minimum dominating set. The overall algorithmic process entails employing a quantum algorithm to obtain the minor dominating set of a graph, subsequently employing this minor dominating set as input for a classical algorithm, and finally outputting the minimum dominating set of the graph.

Algorithm 1 Quantum algorithm of minor dominating set

Input: An undirected graph $G = (V,E)$.
Output: the minor dominating set of graph G.

```
Quantum Circuit(2n + 2m + 3)
for v in V do
  Define a controlled gate.
  |φ_v⟩ as control qubit.
  for u in N(v) do
    |φ_u⟩ as control qubit.
  end for
  |a_v⟩ as target qubit.
end for
Use an MCT gate to record the result of the first phase.
for v in V do
  for u in N(v) do
    Use a Toffoli gate which the control qubit are |φ_v⟩ and |φ_u⟩,
    and the target qubit is |c_u⟩.
end for
Use a controlled gate to record the result on |e⟩.
Define a controlled gate.
  for u in N(v) do
    |φ_u⟩ as control qubit.
  end for
  |d_u⟩ as target qubit.
  Use an MCT gate to record the result.
end for
Use an MCT gate whose target qubit is |ϕ⟩ to flip the phase
of the marked state.
```

3.1. The algorithm idea

We present a novel approach for identifying the minor dominating set in an undirected graph $G = (V,E)$ composed of n vertices. Each vertex in the graph can be classified into one of two categories: dominant or non-dominant. Consequently, the search space for the minor dominating set problem, denoted as S, encompasses 2^n possible vertex sets, where $\mathbb{S} = \{S_i\}_{i=1,\ldots,2^n}$.

The quantum algorithm module is divided into two phases: the first phase focuses on searching all dominating sets, and the second phase focuses on searching all minor dominating sets within the obtained dominating sets.

3.1.1. Circuit design for the minor dominating set problem using Grover's algorithm

The primary task of this research endeavor entails the design of a quantum circuit capable of implementing the aforementioned ideas. Figure 4 shows the cases of the quantum circuit employed for addressing the minor dominating set problem. The algorithm proposed in this study builds upon Grover's algorithm as its foundation. Subsequently, we delve into an elaborate examination of the five sequential steps constituting the algorithm.

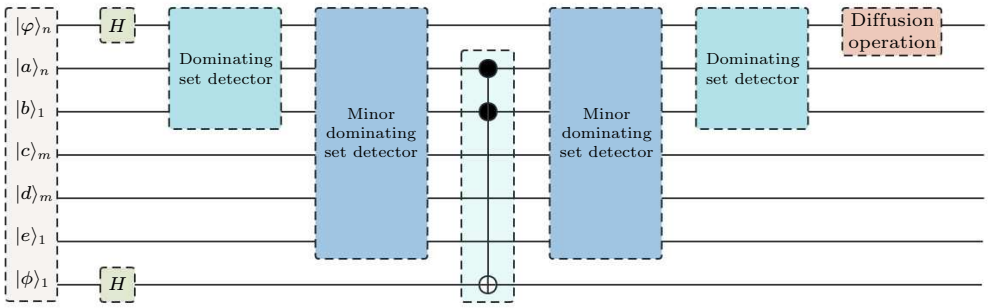


Fig. 4. The general framework of applying Grover's algorithm to the minor dominating set problem, the most important parts are the dominating set detector and minor dominating set detector.

3.1.2. Circuit setup

It is possible to classify the quantum bits into seven distinct categories based on their specific functions within the circuit, as shown in Fig. 4. The effect of different qubits is introduced as follows.

- $|\varphi\rangle_n$: The minimum dominating set problem necessitates the utilization of n qubits, denoted by $|\varphi_x\rangle$, where $x \in \{1,2,\ldots,n\}$, in order to represent the search space associated with this problem.

- $|a\rangle_n$ and $|b\rangle_1$: The state $|a\rangle$ comprises n qubits and serves as a repository for storing the outcome of judgments obtained during both the first and second phases outlined in Subsection 3.1. On the other hand, the state $|b\rangle$ encompasses a single qubit solely designated for recording the outcomes associated with the first phase. For the purposes of representation, let us define the dominant set of the problem as Z. Consequently, the state $|b\rangle$ can be mathematically defined as follows:

$$|b\rangle = \begin{cases} |1\rangle, & \text{if} \quad |\varphi\rangle_x \in Z, \\ |0\rangle, & \text{otherwise.} \end{cases} \tag{5}$$

- $|c\rangle_m$ and $|e\rangle_1$: For an undirected graph $G = (V,E)$, we define

$$m = \max_{v \in V} \text{degree}(v). \tag{6}$$

The state $|c\rangle$ comprises m qubits and serves as storage for information related to the fulfillment of the first judgment condition in the second phase of the method proposed in Subsection 3.1. Specifically, if $|c\rangle_x = |1\rangle$ for $x \in \{1,2,\ldots,m\}$, it indicates that the condition has been satisfied. Additionally, a single qubit denoted as $|e\rangle$ is utilized to record the results corresponding to the state $|c_x\rangle, x \in \{1,2,\ldots,m\}$.

- $|d\rangle_m$: The state $|d\rangle$ is employed to assess whether one of the neighboring vertices of the current vertex can be dominated by other vertices in the dominating set when the current vertex is changed to a non-dominating vertex. If $|d_x\rangle = |1\rangle$ for $x \in \{1,2,\ldots,m\}$, it signifies that this condition holds true for at least one adjacent vertex. In the event that $|d\rangle_m = |1\rangle_m$, it indicates that all adjacent vertices of the current vertex can be dominated by other vertices in the event of the current vertex being changed to a non-dominating vertex. In other words, the current vertex satisfies the second judgment condition in the second phase of the method proposed in Subsection 3.1.

- $|\phi\rangle$: It is utilized for the purpose of flipping the phase of marked states and requires only a single qubit.

3.1.3. Initialization

The initialization process is as follows:

$$\begin{aligned} &|\varphi\rangle \otimes |a\rangle \otimes |b\rangle \otimes |c\rangle \otimes |d\rangle \otimes |e\rangle \otimes |\phi\rangle \\ =&|0\rangle^{\otimes n} \otimes |1\rangle^{\otimes n} \otimes |0\rangle \otimes |0\rangle^{\otimes m} \otimes |1\rangle^{\otimes m} \otimes |1\rangle \otimes |1\rangle. \end{aligned} \tag{7}$$

3.1.4. Hadamard transformation

The Hadamard transformation is applied to both the data qubits and the phase reversal ancilla qubits. The transformation can be defined as follows:

$$\begin{aligned} |\varphi\rangle = |0\rangle^{\otimes n} \to |\varphi\rangle &= \sum_{i=0}^{2^n-1} |i\rangle, \\ |\phi\rangle = |1\rangle \to |\phi\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \end{aligned} \tag{8}$$

In our representation, the state $|1\rangle$ signifies that the current vertex is classified as a dominant vertex, while the state $|0\rangle$ represents that the vertex is categorized as a non-dominant vertex. We take $|101\rangle$ as an example, where the two $|1\rangle$ means that vertices A and C are dominant vertices, and the middle $|0\rangle$ means that vertex B is a non-dominant vertex. After Hadamard transformation on the data qubits, the search space will be encoded into the circuit.

3.1.5. Oracle

Before introducing oracle, in order to better understand the algorithm, it is necessary to introduce two related concepts, namely garbage qubit and mirror. When a qubit undergoes an operation and is not reset to its initial state, it becomes unusable and is referred to as a garbage qubit. To prevent this,

a mirror is employed to restore such qubits to their original states for subsequent use, thereby eliminating a large number of circuit costs. Consequently, the oracle circuit presented in this research exhibits symmetry.

The proposed quantum circuit is partitioned into two distinct phases, aligning with each respective stage elucidated in the proposition of the algorithm described in Subsection 3.1. Quantum logic gates are commonly used to express quantum operations. In graph $G = (V,E)$ with n vertices, we require the Toffoli gate, MCT gate, and CNOT gate to complete the circuit. Referring to Fig. 4, the details are given as follows:

Dominating set detector For every $v \in V$, we use $N(v)$ to represent the adjacent vertices of v, and $|\varphi_w\rangle$ to represent one vertex set of $w = v \cup N(v)$. When $|\varphi_w\rangle = |0,\ldots,0\rangle$ is satisfied, the set comprising $w = 0,\ldots,0$ is identified as non-dominating set. We mark these dominating sets by introducing an ancilla qubit $|a_v\rangle$. Specifically, we introduce a controlled gate, where $|\varphi_w\rangle$ serves as the control gate, and $|a_v\rangle$ as the target bit. When $|\varphi_w\rangle = |0,\ldots,0\rangle$ is satisfied, the target qubit $|a_v\rangle$ undergoes a bit flip operation.

After all the vertices have been done, we employ a controlled gate to map the marked results onto the ancilla qubit $|b\rangle$ and restore all $|a_v\rangle_{v\in V}$ for subsequent procedures. The controlled gate takes $|a_v\rangle_{v\in V}$ as the control qubits and $|b\rangle$ as the target qubit. In the case of combining non-dominated sets, no operations are applied to $|b\rangle$. Conversely, when $|b\rangle$ undergoes a transformation from $|0\rangle$ to $|1\rangle$, it indicates that the current set is a dominating set. Thus, if $|b\rangle$ is in the state $|0\rangle$, it signifies that the current set is a non-dominating set. Otherwise, the current set is identified as a dominating set.

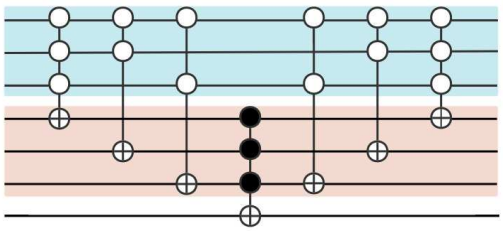


Fig. 5. Dominating set detector. The circuit is to mark the set that can dominate vertex A, B and C in the given undirected graph in Fig. 2.

Taking the undirected graph in Fig. 2 as an example, its corresponding circuit diagram is shown in Fig. 5. Following the application of the $H^{\otimes 3}$, the state $|\varphi\rangle_3$ represents the superposition of the search space spanning from $|000\rangle$ to $|111\rangle$. The initial controlled gate, referred to as an MCT gate, is utilized to mark the sets in which the vertex A is a non-dominating vertex.

Notably, when all the control qubits are in the state $|0\rangle$, signifying that vertex A cannot be dominated in the current set, the target qubit $|a\rangle$ is transformed from $|1\rangle$ to $|0\rangle$. Similar operations are performed on the remaining vertices. Only when all the marked qubits remain un-flipped, namely $|a_1,a_2,a_3\rangle = |1,1,1\rangle$, $|b\rangle$ is transformed from $|0\rangle$ to $|1\rangle$. The outcome $|b\rangle = |1\rangle$ serves as a marker for the dominating set. Consequently, the state $|\varphi_1,\varphi_2,\varphi_3,b\rangle$ assumes the superstition of $|0000\rangle$, $|0010\rangle$, $|0100\rangle$, $|0111\rangle$, $|1001\rangle$, $|1011\rangle$, $|1101\rangle$, and $|1111\rangle$. The resulting dominating sets are $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$, and $|111\rangle$, where $|b\rangle = |1\rangle$. These states correspond to $\{B,C\}$, $\{A\}$, $\{A,C\}$, $\{A,B\}$, and $\{A,B,C\}$. The three controlled gates on the right side of the circuit serve to restore the qubits of the ancillary circuit $|a_v\rangle$.

Minor dominating set detector This process is divided into two phases.

For $v \in V$, set $k = \text{degree}(v)$ to denote its degree. We can employ k controlled gates to determine whether the vertex v can be dominated after it becomes a non-dominating vertex. For each $u \in N(v)$, the control qubit of each controlled gate is $|\varphi_w\rangle$, where $w = v \cup u$ and the target qubit is $|c_u\rangle$. Then, introduce another ancilla $|e\rangle$, which serves as a marker to record the sets in which v can be dominated even after it becomes a non-dominating vertex. This is accomplished by employing a controlled gate, where $|c_u\rangle$ serves as a control qubit. The resulting state $|e\rangle = |1\rangle$ signifies that the corresponding sets are non-minor dominating sets, as vertex v can still be dominated even after it becomes a non-dominating vertex.

For each adjacent vertex u of the current vertex, we can employ a controlled gate to determine whether u can be dominated after it becomes a non-dominating vertex. The control qubit of the controlled gate is $|\varphi_w\rangle$, where $w = u \cup N(u)/v$, and the target qubit is $|d_u\rangle$. If the target qubit $|d_u\rangle$ is in $|1,\ldots,1\rangle$, it indicates that the adjacent vertices of the current vertex are already in the dominating set. Therefore, each adjacent vertex of the current vertex can be dominated even after the current vertex becomes a non-dominating vertex.

We use an MCT gate to record the result of the states after performing the above operations, where $|d\rangle$ and $|e\rangle$ are control qubits, and $|a_v\rangle$ is the target qubit. If $|a_v\rangle$ is in $|1\rangle$, it signifies that this set is a minor dominating set. After all vertices in the set are judged, we employ another MCT gate to mark the minor dominating sets and flip their corresponding phases. The control qubits of the MCT gate are $|a\rangle$ and $|b\rangle$, while the target qubit is denoted as $|\phi\rangle$.

Taking Fig. 2 as an example, its judgment circuit diagram for vertex A is shown in Fig. 6. The initial three controlled gates are utilized to assess the dominant ability of the vertex A subsequent to its removal from the dominating set. The third controlled gate is used to map the result onto the ancilla $|e\rangle$. Specifically, it turns the dominating set as well as ancilla into

$$(|011\rangle+|100\rangle+|101\rangle+|110\rangle+|111\rangle)|00\rangle_c|1\rangle_e \xrightarrow{\text{with three controlled gates}}$$
$$(|011\rangle+|100\rangle)|00\rangle_c|0\rangle_e+(|101\rangle|01\rangle_c+|110\rangle|10\rangle_c+|111\rangle|11\rangle_c)|1\rangle_e := \phi_1, \tag{9}$$

where $e=1$ serves as an indicator denoting the set in which vertex A can still be dominated even after its removal from said set.

The fourth and fifth controlled gates are used to judge whether the vertices B and C (all adjacent vertices of vertex A) can still be dominated when the vertex A becomes a non-dominating vertex. The sixth controlled gate maps the result to $|a\rangle$. The five controlled gates on the right side are used to restore the initial qubit of $|c\rangle$, $|d\rangle$, and $|e\rangle$. Specifically,

$$\phi_1 \xrightarrow{\text{controlled gates by } B,\ C}$$
$$(|011\rangle|11\rangle_d+|100\rangle|00\rangle_d)|00\rangle_c|0\rangle_e+(|101\rangle|01\rangle_d|01\rangle_c+|110\rangle|10\rangle_d|10\rangle_c+|111\rangle|11\rangle_d|11\rangle_c)|1\rangle_e \xrightarrow[\text{restore operations}]{\text{map to ancilla } |a\rangle}$$
$$(|011\rangle|1\rangle_a+|100\rangle|1\rangle_a+|101\rangle|1\rangle_a+|110\rangle|1\rangle_a+|111\rangle|0\rangle_a)|c\rangle|d\rangle|e\rangle. \tag{10}$$

Therefore, after the judgment operations based on A and its adjacent vertices, the set $|111\rangle$ which is a non-minor dominating set is marked.

The aforementioned judgment operations are sequentially applied to vertices A, B, and C within the overall circuit. The circuit diagram is depicted in Fig. 7. We use an MCT gate whose target qubit is $|\phi\rangle$ to flip the phase of the minor dominating sets, and the controlled gates on the right side symmetrical about the MCT gate are used to restore the ancilla circuit. The controlled gates situated to the right of the MCT gate exhibit symmetry and serve the purpose of restoring the ancilla circuit. This restoration ensures that the qubit value of the ancilla circuit remains consistent with its state prior to executing the algorithm in subsequent Grover iterations. The final marked minor dominating sets comprise $|011\rangle$ and $|100\rangle$, corresponding to $\{B,C\}$ and $\{A\}$, respectively.

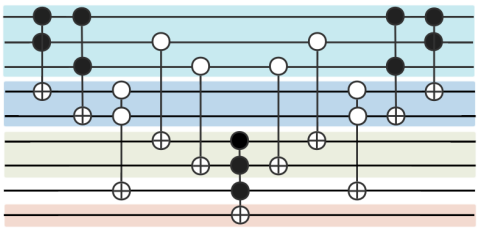


Fig. 6. Minor dominating set detector. The circuit is to mark that a dominant set containing vertex A (the first line) is still a dominant set when vertex A is eliminated from the dominant vertex. If so, A can be eliminated in the dominant set. Similar circuits will be executed in sequence for the remaining vertices to check whether this vertex can be eliminated. Finally, turn the dominant set into a minor dominating set.

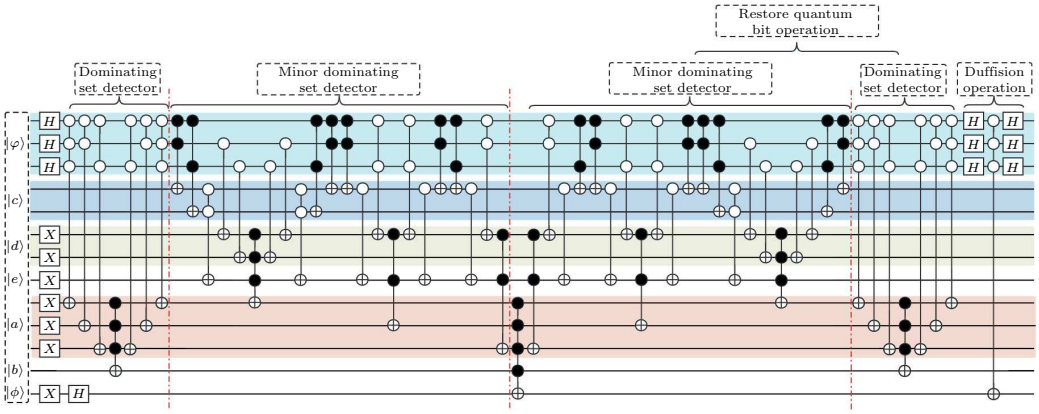


Fig. 7. The circuit for dominating set problem of the given three-vertex undirected graph. It includes two modules, i.e., the dominating set detector module and the minor dominating set detector module. The dominating set detector module is employed to identify all dominating sets within the graph, while the minor dominating set detector module is utilized to identify the minor dominating sets within the graph. Lastly, the diffusion operation module is applied to amplify the identified collection of minor dominating sets.

3.1.6. Diffusion operation

The diffusion operation, which forms an integral part of the Grover iteration, can be conceptualized as a reflection of the state [as described in Eq. (1)]. The operation of this part is

$$H^{\otimes n}(2|0\rangle\langle 0|-I)H^{\otimes n}, \tag{11}$$

which equals $2|X\rangle\langle X|-I$.

The circuit diagram depicting the complete process of finding the minor dominating set for Fig. 2 is illustrated in Fig. 7, considering the incorporation of multiple measurements. By performing multiple measurements on the data qubits, we are able to identify the set that contains the fewest dominating vertices, known as the minor dominating set. However, before conducting measurements, an important is-

sue that needs to be addressed is to determine the optimal iteration numbers. In the case of minor dominating set problems, the number of potential solutions remains unknown, rendering the direct application of original Grover's search algorithm impractical.

From the practical implementation perspective, the H gate, CNOT gate, and Pauli X gate in the quantum circuit designed above are basic gates and can be directly implemented on quantum computers. However, the circuit also involves multi-controlled gates. For these controlled multi-qubit gates, we can typically decompose them into combinations of basic gates and entangling operations, with implementation methods varying based on the specific quantum platform being used. There have been numerous studies on the decomposition of multi-controlled gates.[38,41–45] The seminal work by Barenco *et al.*[38] introduced a technique to decompose an $(n+1)$-qubit controlled single-qubit unitary gate into a combination of single-qubit gates and controlled-X gates. However, the resulting circuit depth and the number of quantum gates employed exhibit quadratic scaling with the number of qubits. Saeedi *et al.*[41] proposed a linear-depth, quadratic-size quantum circuit for realizing an n-qubit Toffoli gate using controlled-rotation gates. Notably, this method does not require an ancilla qubit. Da *et al.*[42] shows that an n-qubit controlled single-qubit unitary gate can be decomposed into a circuit with a linear depth of single-qubit and CNOT gates. Vartiainen *et al.*[43] decomposes multiple controlled gates through the use of Gray code. He *et al.*[44] proposes a work that only requires $O(n)$ number of general resources for an n-qubit Toffoli gate. Vale *et al.*[45] decomposes multiple controlled gates into single-qubit unitary gates and controlled-X gates, similarly, no auxiliary qubits are required. This approach enables achieving linear scaling in both circuit depth and the number of quantum gates.

3.2. Estimation of the number of iterations of Grover's algorithm

For problems with known target solutions, the number of iterations can be determined with certainty. Specifically, if the number of the target states is known and denoted as M, the required number of iterations can be calculated as $T \leq \lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \rceil$. After T times of iterations, the target states are observed with a high probability. However, if the number of iterations is inappropriate, the probability of getting the correct result will decrease. Therefore, if M is unknown, it is difficult to find the right times to stop iterations.

We propose a more effective method that uses a swap test to estimate the number of iterations. In the subsequent sections, we provide a comprehensive and detailed description of the estimation process. The visualization of one iteration of Grover's algorithm is shown in Fig. 8, where $|\alpha\rangle$ is the superposition state of the non-target solution and $|\beta\rangle$ is the superposition state of the target solution. The rotated angle of the one iteration is denoted as θ, which is the value related to the number of iterations. Figure 9 shows the swap test circuit. Taking $|\varphi\rangle$ and $G|\varphi\rangle$ as the input of the swap test, the output state is expressed as follows:

$$\begin{aligned}|0,\phi,\psi\rangle &\xrightarrow{H_1} \frac{1}{\sqrt{2}}|0,\phi,\psi\rangle + \frac{1}{\sqrt{2}}|1,\phi,\psi\rangle \\ &\xrightarrow{\text{Controlled-SWAP}} \frac{1}{\sqrt{2}}|0,\phi,\psi\rangle + \frac{1}{\sqrt{2}}|1,\psi,\phi\rangle \\ &\xrightarrow{H_1} \frac{|0,\phi,\psi\rangle + |1,\phi,\psi\rangle}{2} + \frac{|0,\psi,\phi\rangle - |1,\psi,\phi\rangle}{2} \\ &= \frac{|0,\phi,\psi\rangle + |0,\psi,\phi\rangle}{2} + \frac{|1,\phi,\psi\rangle - |1,\psi,\phi\rangle}{2}, \end{aligned} \tag{12}$$

where $|\phi\rangle := |\varphi\rangle$ and $|\psi\rangle := G|\varphi\rangle$. Finally, we measure the first qubit in the Z basis and obtain the probability of $|0\rangle$ and $|1\rangle$:

$$\begin{aligned}P(|0\rangle) =& \frac{1}{4}(\langle 0,\phi,\psi| + \langle 0,\psi,\phi| + \langle 1,\phi,\psi| \\ &- \langle 1,\psi,\phi|)(|0\rangle\langle 0| \otimes I \otimes I)(|0,\phi,\psi\rangle + |0,\psi,\phi\rangle \\ &+ |1,\phi,\psi\rangle - |1,\psi,\phi\rangle) \\ =& \frac{1+|\langle\phi|\psi\rangle|^2}{2}, \\ P(|1\rangle) =& 1 - P(|0\rangle) = \frac{1-|\langle\phi|\psi\rangle|^2}{2}. \end{aligned} \tag{13}$$

Through $|\langle\phi|\psi\rangle|^2$, we obtain the angle θ of rotation for one iteration of Grover's algorithm between $|\phi\rangle$ and $|\psi\rangle$. The accuracy of the swap test is closely related to the number of measurements. In general, the accuracy can be increased as the number of measurements increases, considering that repeated measurements can reduce errors. The error probability of the swap test can be reduced to any $\varepsilon > 0$ by setting the associated value using the number of measurements $k \in (O(\log_2 \frac{1}{\varepsilon}))$.[46] The current method demonstrates an acceptable trade-off between precision and the number of measurements required. As a result, its feasibility is affirmed at present.

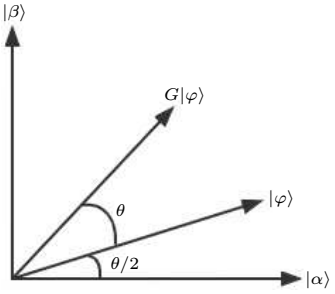


Fig. 8. One iteration visualization of Grover's algorithm.

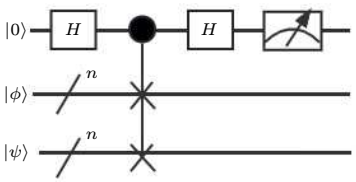


Fig. 9. General quantum circuit of swap test.

3.3. Circuit cost estimation

We describe the overall design process of our quantum algorithm, and conduct a detailed analysis of the circuit cost associated with the algorithm, as presented in Table 1. For an undirected graph $G = (V, E)$ with n vertices, we analyze the cost from space complexity and time complexity.

Table 1. Cost analysis of algorithm.

Number of vertex	Qubit cost	Gate cost
3	20	83
n	$O(n)$	$O(Tn^2)$

The cost of the qubit in the algorithm is called space complexity. The oracle we design requires n data qubits and $2m+n+3$ ancilla qubits when the maximum degree is m. In addition, for $|\varphi\rangle$ of n qubits, the swap test requires $2n+1$ qubits. Therefore, we require a total of $2m+4n+4$ qubits.

The time complexity of the algorithm is determined by the cost of the quantum gates utilized. However, before delving into time complexity, it is important to first introduce the query complexity of Grover's algorithm. As depicted in Fig. 8, it can be observed that $|\varphi\rangle$ approaches $|\beta\rangle$ after $T = CI(\frac{\pi/2-\theta/2}{\theta})$ iterations. For the scenarios of $M \ll N(N = 2^n)$, it can be deduced that $T \leq \lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \rceil$. In other words, the query complexity of this algorithm can be expressed as $O(\sqrt{N/M}) \approx O(1.414^n)$. The time complexity mentioned in Refs. [20,47] corresponds to the query complexity here. It is worth noting that the time complexity of those algorithms is higher than the aforementioned algorithm, denoted as $O(1.93782^n)$ and $O(1.8021^n)$, respectively. Next, we analyze the complexity of the algorithm which is related to the gate cost. In order to execute the algorithm in its entirety, it necessitates the utilization of $n+2$ quantum gates in the swap test and $2n+1$ quantum gates in the first phase of oracle.

The number of quantum gates required by the second phase of the oracle is related to the degree of each vertex in the undirected graph, rendering it infeasible to provide a precise numerical value. In order to assess the worst-case scenario, we consider a fully connected graph, where the gate cost amounts to $4n^2-3n+1$. Additionally, the restoration of ancilla qubits necessitates an equivalent number of gates as the first and second phases. Within the diffusion operator, the incorporation of $2n+1$ quantum gates is essential. Consequently, after T iterations, a total of quantum gates amount to $(T+1)(8n^2+6)+(n+2)$, which is expected to yield superior results in the majority of cases.

After getting the minor dominating set through the algorithm, we can use the classical computer to get the minimum dominating set with $O(M)$ where M is the number of minor dominating sets obtained by the quantum algorithm. However, this cost has little impact on the overall cost.

The complexity of this algorithm is more optimistic than that of classical algorithms. In the classical algorithms, Fomin *et al.* proposed a theory in Ref. [25] that focuses on reducing the search space and accelerating the algorithm's efficiency. However, this approach is limited to graphs with a minimum vertex degree of 3. The query complexity associated with it is denoted as $O(1.93782^n)$. Grandoni, on the other hand, established a correlation between the time complexity of the minimum set cover problem and the minimum dominating set problem in Ref. [47], this method by leveraging dynamic programming techniques reduces the complexity of the minimum set cover problem, the time complexity of the minimum dominating set problem was also improved to $O(1.8021^n)$. Although these methods yield accurate results, their elevated time complexity poses a challenge for current computing capabilities. Consequently, reducing time costs through the utilization of quantum algorithms holds significant promise.

3.4. The success rate of algorithm

The success rate is one of the important indicators to measure whether the algorithm is excellent. As far as this method is concerned, one of the root causes of the error is the process of calculating θ. According to the cost analysis in the above section, the cost of calculating θ for the swap test circuit is negligible in terms of the overall cost. Consequently, by conducting multiple measurements, the cumulative impact on accuracy can be disregarded.

We consider the impact on the success rate from the different ratios of M and N. Figure 8 reveals a correlation between the success rate and the proportion of the final state that is projected onto the $|\beta\rangle$ state. The rotation angle after T iterations is given by $T\theta + \frac{\theta}{2}$, where $T = CI(\frac{\pi/2-\theta/2}{\theta})$. The success rate, denoted by P, corresponds to $P = \sin^2(T\theta + \frac{\theta}{2})$. As known in Ref. [27], θ is related to M and N, and the relationship is $\theta = 2\arcsin\sqrt{M/N}$. Therefore, by deduction that P is related to λ where $\lambda = \sqrt{M/N}$. Subsequently, we analyze the impact of $1/\lambda$ on probability P, and the results are shown in Table 2.

Table 2 shows that the algorithm we presented has a high tolerance to the uncertainty of the high ratio $\lambda = \sqrt{M/N}$. However, in the case of $M \geq N/2$, the algorithm directly degenerates into a classical brute force search algorithm. For the low-ratio λ searching problem, one can use the Grover-long algorithm proposed by Long in Ref. [48], which has improved Grover's search algorithm to have zero theoretical failure rate by using rotation instead of phase inversion. However, executing this algorithm becomes more challenging when the value of M is unknown. In such cases, we can utilize the swap test method mentioned in the preceding section to address this issue.

Table 2. Relationship between λ and P.

$1/\lambda$	$\sqrt{4/3}$	$\sqrt{2}$	$\sqrt{3}$	2	5	10	50	100
T	0	0	1	1	3	7	45	75
P	0.75	0.5	0.933	1	0.990	0.994	0.998	0.998

4. Simulation of minimum dominating set algorithm

We simulate the quantum algorithm using a Python software package called Quantum Information Science Kit (Qiskit). The software package uses Open QASM or Open Quantum Assembly Language developed by IBM and is used in all IBM quantum hardware. As examples for our simulation, we consider two undirected graphs with three vertices and eight vertices, respectively.

4.1. Simulation of three-vertex graph

Taking Fig. 2 as an example of three-vertex graph, we can get $\cos\theta = 1/2$ through swap test circuit, then obtain the iteration $T = CI(\frac{\pi/2-\theta/2}{\theta}) = 1$. The simulation results are shown in Fig. 10, where the abscissa represents the set and the ordinate represents the probability. For the three-vertex graph, there are 8 combinations from $|000\rangle$ to $|111\rangle$. After one iteration of the algorithm, only $|100\rangle$ and $|011\rangle$ can be obtained through measurement and their respective probabilities are close to 0.5. Through the algorithm, the minor dominating sets $\{A\}$ and $\{B,C\}$ are obtained with a high probability. Among these sets, we identify $\{A\}$ as the minimum dominating set, which exhibits the smallest cardinality among all dominating vertex sets.

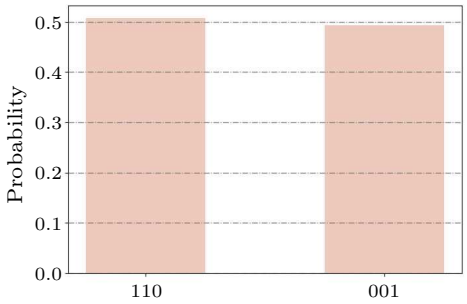


Fig. 10. Using qiskit to simulate the graph shown in Fig. 2, two minor dominating sets are obtained through measurement, with a probability of 0.5.

4.2. Simulation of eight-vertex graph

We randomly generate a graph with eight vertices as shown in Fig. 11. From the swap test circuit, the angle is calculated with $\cos\theta = 228/256$, then we get the iteration of $T = 3$ according to θ. Figure 12 shows the simulation results after 2000 times measurements. As we can see, the dominating sets are obtained with a high probability. The dominating sets corresponding to the measurement result are $\{A,C,D\}$, $\{A,B,C,E\}$, $\{B,F\}$, $\{D,F\}$, $\{A,C,G\}$, $\{B,C,G\}$, $\{C,E,G\}$, $\{A,F,G\}$, $\{E,F,G\}$, $\{B,C,D,H\}$, $\{C,E,H\}$, $\{A,F,H\}$, $\{E,F,H\}$, and $\{C,D,G,H\}$. Among these sets, the minimum dominating sets are identified as $\{B,F\}$ and $\{D,F\}$.

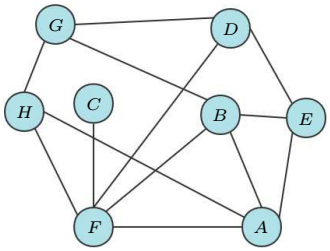


Fig. 11. An undirected graph with eight vertices generated randomly.

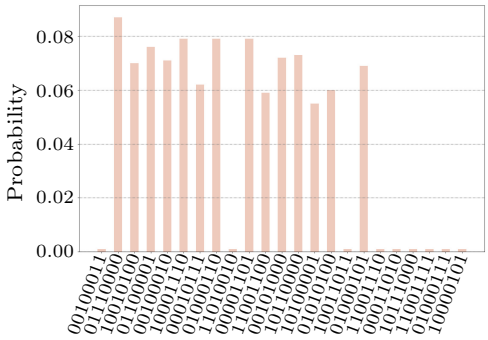


Fig. 12. The simulation result of the graph shown in (a). After measurement, the probability of each target state is about 0.07.

To verify the correctness and effectiveness of the algorithm for any graph, except the simulations of the above three- and eight-vertex graphs, we also performed experimental verification on graphs with different numbers of vertices, particularly 2, 4, 5, 6, and 8 vertices. The results of the experimental verification are presented in Table 3. It can be observed that the accuracy of the algorithm is close to 100 percent, indicating that it performs well in correctly identifying graph properties. Additionally, the running time of the algorithm is reasonable, considering the complexity of the task. However, due to limitations in computer memory capacity, we were only able to verify graphs up to 8 vertices. Based on the results presented in Table 3, it can be concluded that the circuit running time and accuracy of the algorithm are acceptable for the tested graphs. Furthermore, the current experimental results show that the proposed algorithm outperforms the method described in Ref. [47]. We are confident that the algorithm will yield even more promising results as computer memory capacity expands.

Table 3. Execution efficiency of undirected graphs with different vertex numbers.

Vertex number	2	4	5	6	8
Circuit running time (s)	0.53	0.76	1.05	1.38	2.99
Precision	1.000	1.000	1.000	0.998	0.990

5. Conclusion

In this paper, we present an automated quantum algorithm utilizing Grover's search algorithm to solve the problem of finding the minimum dominating set in undirected graphs. Comparative analysis with classical brute force search demonstrates a quadratic improvement in query complexity, rendering our algorithm more efficient and accurate than existing classical approximation algorithms. Our designed oracle exhibits a gate cost and circuit cost of $O(n^2)$ and $O(n)$, respectively. Furthermore, we propose a novel method for estimating the number of iterations by leveraging fidelity obtained from the swap test. This method features a simple circuit design, easy implementation, and broad applicability to any problem addressed by Grover's algorithm. Experimental validation supports the superior performance of our method in accurately determining the optimal number of iterations.

To demonstrate the effectiveness and efficiency of our algorithm, we conduct simulations using Python and the qiskit package developed by IBM. We showcase its performance on undirected graphs with three and eight vertices, showcasing the algorithm's accuracy in obtaining minor dominating sets with high probability. Subsequently, classical post-processings are employed to transform these minor dominating sets into minimum dominating sets, significantly reducing the search space and enabling accurate solutions within a short time. All the quantum circuits in this study are manually designed. However, future research should explore the automatic design of quantum circuits for a given graph. In this work, we employ a simulation platform to verify the accuracy of our proposed methods. In the future, we plan to optimize the quantum circuit implementation and conduct experiments on real quantum computers. Additionally, we will investigate the impact of hardware noise on the performance of the circuit implementation.

Acknowledgments

Project supported by the National Natural Science Foundation of China (Grant No. 62101600), the Science Foundation of China University of Petroleum, Beijing (Grant No. 2462021YJRC008), and the State Key Laboratory of Cryptology (Grant No. MMKFKT202109).

References

[1] Khadiev K and Ilikaev A 2019 *International Conference on Theory and Practice of Natural Computing*, December 9–11, 2019, Kingston, Canada, p. 234
[2] Lee T, Magniez F and Santha M 2017 *Algorithmica* **77** 459
[3] Friedman J R, Patel V, Chen W, Tolpygo S K and Lukens J E 2000 *Nature* **406** 43
[4] Ficek Z and Swain S 2005 *Quantum interference and coherence: theory and experiments* (Heidelberg: Springer Science & Business Media) Vol. 100 p. 3
[5] Horodecki R, Horodecki P, Horodecki M and Horodecki K 2009 *Rev. Mod. Phys.* **81** 865
[6] Jeffery S, Kothari R, Le Gall F and Magniez F 2016 *Algorithmica* **76** 1
[7] Kravchenko D, Khadiev K and Serov D 2019 *International Computer Science Symposium*, July 1–5, 2019, Novosibirsk, Russia, p. 228
[8] Shukla A and Vedula P 2019 *J. Global Optim.* **75** 199
[9] Mukherjee S 2022 *IEEE Trans. Quantum Eng.* **3** 3101008
[10] Saha A, Saha D and Chakrabarti A 2020 *IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, December 20–22, 2020, India, Jaipur, p. 17
[11] Shimizu K and Mori R 2022 *Algorithmica* **84** 3603
[12] Wang Y and Perkowski M 2011 *IEEE International Symposium on Multiple-Valued Logic*, May 23–25, 2011, Tuusula, Finland, p. 294
[13] Saha A, Chongder A, Mandal S B and Chakrabarti A 2015 *IEEE International Symposium on Nanoelectronic and Information Systems*, December 19–21, 2015, Bhubaneswar, India, p. 101
[14] Aghae M R S, Zukarnain Z A, Mamat A and Zainuddin H 2008 *International Conference on Advanced Computer Theory and Engineering*, July 29–30, 2008, Phuket, Thailand, p. 251
[15] Fujimura T 2024 *Global J. Pure Appl. Math.* **6** 263
[16] Hartmanis J 1982 *SIAM Rev.* **24** 90
[17] Karbasi A H and Atani R E 2013 *Int. J. Secur. Its Appl* **7** 185
[18] Sanchis L A 2002 *Algorithmica* **33** 3
[19] Hedar A R and Ismail R 2010 *International conference on computational science and its applications*, March 23–26, 2010, Fukuoka, Japan, p. 457
[20] Fomin F V, Kratsch D and Woeginger G J 2004 *International Workshop on Graph-Theoretic Concepts in Computer Science*, June 25–27, 2004, Berlin, Heidelberg, p. 245
[21] Van Rooij J M and Bodlaender H L 2011 *Discrete Appl. Math* **159** 2147
[22] Fomin F V, Grandoni F and Kratsch D 2005 *International Colloquium on Automata, Languages, and Programming,* July 4–8, 2005, Lisbon, Portugal, p. 191
[23] Schiermeyer I 2008 *Discrete Appl. Math.* **156** 3291
[24] Van Rooij J M and Bodlaender H L 2008 arXiv:0802.2827 [cs.DS]
[25] Fomin F V, Grandoni F and Kratsch D J 2009 *J. ACM* **56** 25
[26] Iwata Y 2011 *International Symposium on Parameterized and Exact Computation*, September 8–10, 2011, Berlin, p. 41
[27] Grover L K 1997 *Phys. Rev. Lett.* **79** 325
[28] Grover L K 1998 *Phys. Rev. Lett.* **80** 4329
[29] Long G L, Li Y S, Zhang W L and Tu C C 2000 *Phys. Rev. A* **61** 042305
[30] Chuang I L, Gershenfeld N and Kubinec M 1998 *Phys. Rev. Lett.* **80** 3408
[31] Boyer M, Brassard G, Høyer P and Tapp A 1998 *Fortschritte der Phys.* **46** 493
[32] Zhu W, Chen H, Liu Z and Xue X 2014 *International Conference in Swarm Intelligence*, June 12–15, 2014, Hefei, China, p. 357
[33] Aaronson S and Rall P 2020 *Symposium on Simplicity in Algorithms*, January 5–8, 2020, Salt Lake City, Utah, USA, p. 24
[34] Guo Y, Shi W, Wang Y and Hu J 2017 *J. Phys. Soc. Jpn.* **86** 024006
[35] Ma B W, Bao W S, Li T, Li F G, Zhang S and Fu X Q 2017 *Chin. Phys. Lett.* **34** 070305
[36] Korepin V E and Grover L K 2006 *Quantum Inf. Process.* **5** 5
[37] Younes A, Rowe J and Miller J 2008 *Physica D* **237** 1074
[38] Barenco A, Bennett C H, Cleve R, DiVincenzo D P, Margolus N, Shor P and Weinfurter H 1995 *Phys. Rev. A* **52** 3457
[39] Haynes T W, Hedetniemi S and Slater P 2013 *Fundamentals of domination in graphs* (New York: CRC Press) p. 5
[40] Haynes T W, Hedetniemi S T and Henning M A (eds) 2017 *Topics in Domination in Graphs* (Cham: Springer) Vol. 2 pp. 13–30
[41] Saeedi M and Pedram M 2013 *Phys. Rev. A* **87** 062318
[42] Da Silva A J and Park D K 2022 *Phys. Rev. A* **106** 042602
[43] Vartiainen J J, Möttönen M and Salomaa M M 2004 *Phys. Rev. Lett* **92** 177902
[44] He Y, Luo M X, Zhang E, Wang H K and Wang X F 2017 *Int. J. Theor. Phys* **56** 2350
[45] Vale R, Azevedo T M D, Araújo I, Araujo I F and da Silva A J 2023 arXiv:2302.06377 [quant-ph]
[46] Buhrman H, Cleve R, Watrous J and De Wolf R 2001 *Phys. Rev. Lett.* **87** 167902
[47] Grandoni F 2006 *J. Discrete Algorithms* **4** 209
[48] Long G L 2001 *Phys. Rev. A* **64** 022307