mathematics

MDPI

Article

Using a Simplified Quantum Counter to Implement Quantum Circuits Based on Grover's Algorithm to Tackle the Exact Cover Problem

Jehn-Ruey Jiang * and Yu-Jie Wang

Department of Computer Science and Information Engineering, National Central University, Taoyuan 320317, Taiwan
* Correspondence: jrjiang@csie.ncu.edu.tw

Abstract: In this paper, we use a simplified quantum counter to implement Grover's algorithm-based quantum circuits to tackle the NP-hard exact cover problem (ECP). The ECP seeks a subcollection of sets such that every element is covered by exactly one set. Leveraging Grover's algorithm, our quantum circuits achieve a quadratic speedup, querying the oracle $O(\sqrt{N})$ times, compared to $O(N)$ for classical methods, where $N = 2^n$ is the total number of unstructured input instances and n is the number of input (quantum) bits. For the whole quantum circuit, the simplified quantum counter saves $(4mb - 4m)\lfloor \pi/4\sqrt{N/M} \rfloor$ quantum gates and reduces the quantum circuit depth by $(2mb)\lfloor \pi/4\sqrt{N/M} \rfloor$ compared to Heidari et al.'s design, where $b = \lfloor \log n \rfloor + 1$ is the number of counting qubits used in a counter. Experimental results obtained using IBM Qiskit packages confirm the effectiveness of our quantum circuits.

Keywords: exact cover problem; Grover's algorithm; oracle; quantum circuit; quantum counter; quantum search

MSC: 68Q12; 81P68

Academic Editor: João Nuno Prata

Received: 5 December 2024
Revised: 24 December 2024
Accepted: 28 December 2024
Published: 29 December 2024

Citation: Jiang, J.-R.; Wang, Y.-J. Using a Simplified Quantum Counter to Implement Quantum Circuits Based on Grover's Algorithm to Tackle the Exact Cover Problem. *Mathematics* **2025**, *13*, 90. https://doi.org/10.3390/math13010090

1. Introduction

Quantum computers operate on quantum bits (or qubits), harnessing phenomena such as quantum superposition, quantum entanglement, and quantum tunneling [1,2] to perform computations. Classical computers utilize bits for computing, which can only represent either 0 or 1 at any given time. Unlike a bit, a qubit can exist in a superposition of both 0 and 1; it collapses to either 0 or 1 only when measured. As a result, with the superposition of n qubits, it is possible to represent and operate on all 2^n states simultaneously. In contrast, n classical bits can represent and operate on only one of 2^n states at a time. The computational power of quantum computers thus scales exponentially with the number of qubits, surpassing the computational capabilities of classical computers and hence achieving quantum supremacy [3].

Some quantum computers, like IBM Q [4], Google Sycamore [5], and Rigetti Ankaa [6], are classified as universal, whereas others, such as D-Wave Advantage [7] and QuEra Aquila [8], are non-universal. Universal quantum computers utilize quantum gates to create quantum circuits to tackle a wide range of problems. In contrast, non-universal quantum computers employ alternative methods instead of quantum gates to address specific challenges. For instance, D-Wave Advantage uses quantum annealing, a mechanism

that exploits quantum tunneling to traverse the energy landscape and identify the lowest global energy state [9] to address various combinatorial optimization problems [2].

In this paper, we focus on universal quantum computers that use quantum circuits composed of quantum gates to address problems. Numerous quantum algorithms based on quantum circuits composed of quantum gates have been proposed in the literature, including the well-known Deutsch–Jozsa algorithm [10], Shor's algorithm [11], and Grover's algorithm [12], which are elaborated below.

In 1992, Deutsch and Jozsa proposed the Deutsch–Jozsa algorithm [10], which addresses the problem of determining whether a given oracle or black-box function $f(x)$ is a balanced function or a constant function, where x is the input consisting of n qubits. If $f(x)$ returns 0 for all inputs or 1 for all inputs, then $f(x)$ is a constant function. In contrast, if $f(x)$ returns 0 for half of the inputs and 1 for the other half, then $f(x)$ is a balanced function. When executed on a quantum computer, the Deutsch–Jozsa algorithm can solve the problem with just one invocation of the oracle or black-box function. In contrast, a classical algorithm requires $2^{n-1}+1$ invocations of the oracle in the worst case to address the same problem. This demonstrates that the Deutsch–Jozsa quantum algorithm offers an exponential speedup compared to classical algorithms designed to address the same problem.

In 1994, Shor proposed Shor's algorithm [11], which can solve the semiprime factorization problem (SFP) to factorize a large semiprime N with polynomial time complexity of $O((\log N)^2(\log\log N)\ (\log\log\log N))$. This time complexity represents an exponential speedup compared to the time complexity of the state-of-the-art classical algorithm, the General Number Field Sieve (GNFS), which has time complexity of $O(e^{1.9(\log N)^{1/3}(\log\log N)^{2/3}})$. Furthermore, Shor's algorithm can also solve the discrete logarithm problem (DLP) with polynomial time complexity. However, so far, no classical algorithm can solve the DLP with polynomial time complexity. It is known that the mainstream public-key cryptosystems, such as Rivest–Shamir–Adleman (RSA), Diffie–Hellman (DH), and ElGamal, rely on the computational difficulty in solving the SFP or the DLP. Thus, if Shor's algorithm is implemented and run on a quantum computer with a sufficient number of reliable qubits, it can rapidly solve the SFP and DLP to break mainstream public-key cryptosystems. Therefore, Shor's algorithm is considered a groundbreaking algorithm and has a profound impact on everyday life.

In 1996, Grover proposed Grover's algorithm [12], a quantum search algorithm used to find a specific input, called the "target input", from N unstructured inputs. Note that we also use "solution" to refer to the "target input" in the following. Grover's algorithm is based on the concept of amplitude amplification. It assumes an "oracle" to invert the phase of a state representing the target input. It also assumes a diffusion operator or "diffuser" that can amplify the amplitude of the target input. Consequently, with $O(\sqrt{N})$ calls to the oracle along with the diffuser, Grover's algorithm can identify the target input with a high probability. In contrast, a classical search algorithm requires $O(N)$ oracle calls to locate the target input from unstructured inputs in both the average case and the worst case. This implies that Grover's algorithm is quadratically faster than the classical search algorithm in terms of oracle calls. In 1997, Bennett et al. proved that any quantum algorithm needs $O(\sqrt{N})$ oracle calls to find a specific solution from N unstructured inputs [13], thus establishing Grover's algorithm as the optimal quantum algorithm for unstructured data searching. Furthermore, in 1998, Boyer et al. deduced that, for a situation with M ($1 \le M \ll N$) target inputs, $T = \lfloor \pi/4\sqrt{N/M} \rfloor$ oracle calls are required to find all M target inputs with a high probability [14].

Several research studies [15–24] have designed quantum circuits based on Grover's algorithm to address different problems. The problems addressed include the k-coloring

problem, the maximum clique problem, the list coloring problem, the pure Nash equilibrium finding problem in graphical games, the Hamiltonian cycle problem, the vertex cover problem, the dominating set problem, and the exact cover problem. These algorithms offer a quadratic speedup in terms of oracle calls when compared to the classical sequential search algorithm.

This paper is an extension of the research in [24], seeking to implement quantum circuits based on Grover's algorithm to address the exact cover problem (ECP) [25], as defined below. Given a universal set U of m elements and a collection C of n sets, where each set is a subset of U, the ECP is to determine if there exists a subcollection $C' \subseteq C$ such that each of m elements is covered by (or belongs to) exactly one set in C'. The ECP has been shown to be NP-hard and NP-complete in [25]. It is thus likely that there exists no classical algorithm to address the ECP with polynomial time complexity in the worst case. The ECP is useful because many complex problems, such as the 3-satisfiability problem and the Hamiltonian cycle problem, can be reduced to the ECP for solutions [25].

Our quantum circuit implementations use a simplified quantum counter, rather than the existing quantum counter proposed by Heidari et al. in [26], to reduce the number of quantum gates and shorten the circuit depth. As will be shown later, for a quantum counter with b counting qubits, the number of gates is reduced by $2b-2$, and the quantum circuit depth is reduced by b. This is particularly advantageous in the current noisy intermediate-scale quantum (NISQ) era [27], where quantum computers have a limited number of qubits, low quantum gate fidelity, and short decoherence times. Specifically, quantum counters are used to build the oracle to check if every element is covered by exactly a set exactly once. If so, a solution is found, and the phase of the qubit state representing the solution is flipped. Then, the probability amplitude of this state is amplified through the diffuser.

The ECP is solved with high probability by calling the oracle along with the diffuser $\lfloor \pi/4\sqrt{N/M} \rfloor = \mathrm{O}(\sqrt{N})$ times, where $N = 2^n$ is the total number of possible input instances, and $M, 1 \leq M \ll N$, is the number of solutions to the ECP. This provides a quadratic speedup compared to classical sequential search algorithms that need to call the oracle $\mathrm{O}(N)$ times. We will later show that the oracle quantum circuit uses m quantum counters twice, with each counter having $b = \lfloor \log n \rfloor + 1$ counting qubits, where m is the number of elements in the universal set given in the ECP. Thus, each single quantum circuit to address the ECP saves $(4mb - 4m)\lfloor \pi/4\sqrt{N/M} \rfloor$ quantum gates and its circuit depth is reduced by $(2mb)\lfloor \pi/4\sqrt{N/M} \rfloor$, which is a significant improvement. We conduct three experiments using the IBM Qiskit packages [28] to implement and execute quantum circuits, successfully solving the ECP for validation purposes.

The contribution of this paper is fourfold. First, we propose using a simplified quantum counter, rather than the existing quantum counter proposed by Heidari et al., to implement quantum circuits, which reduces the number of quantum gates and shorten the quantum circuit depth. Second, we design quantum circuits based on Grover's algorithm using the simplified quantum counter to address the ECP. Third, we analyze the number of quantum gates saved and the reduction in the quantum circuit depth for a single quantum circuit that is employed to address the ECP using the simplified quantum counter. Fourth, we implement and execute the designed quantum circuits with IBM Qiskit packages, successfully solving the ECP for correctness validation.

The remainder of this paper is organized as follows. Some background knowledge is introduced in Section 2, and related works are described in Section 3. Our quantum circuit design is elaborated in Section 4. The results of three experiments based on IBM Qiskit packages are shown in Section 5. Finally, some concluding remarks are given in Section 6.

2. Background Knowledge

2.1. Exact Cover Problem

The exact cover problem (ECP) is a decision problem, as defined below. Given a universal set $U = \{u_0, u_1, \ldots, u_{m-1}\}$ with m elements, and a collection $C = \{s_0, s_1, \ldots, s_{n-1}\}$ of n sets, where each set is a subset of U, the ECP is to determine whether or not there exists a subcollection $C' \subseteq C$ such that C' is an exact cover of U, i.e., every element in U is covered by (or belongs to) exactly one set in C'. The ECP is useful because many intricate problems, such as the 3-satisfiability problem and the Hamiltonian cycle problem, can be reduced to the ECP for solution [25]. However, the ECP has been shown to be NP-hard [25]. It is thus unlikely that there exists any classical algorithm to address the ECP with polynomial time complexity in the worst case.

The original ECP is a decision problem that involves deciding whether there exists an exact cover of U. Nonetheless, this paper aims at the extended version of the ECP to find all exact covers of U. Please note that when we refer to the ECP in this paper, we are specifically referring to the extended ECP.

For example, given a universal set $U = \{u_0, u_1, u_2\}$ of three elements and a collection $C = \{s_0 = \{u_0\}, s_1 = \{u_1\}, s_2 = \{u_2\}, s_3 = \{u_0, u_1, u_2\}\}$ of four sets, the ECP has two solutions, which are the two exact covers of U, namely $C' = \{s_0, s_1, s_2\}$ and $C'' = \{s_3\}$.

2.2. Grover's Algorithm

Lov Kumar Grover proposed Grover's algorithm [12] in 1996. This is a quantum search algorithm designed to find a target data item from N unstructured or unsorted data. Specifically, Grover's algorithm aims to find the "target input" or "solution" that corresponds to the target data item out of N input instances. Assuming the existence of an oracle that can check whether or not an input instance is the target input, Grover's algorithm can locate the target input from N unstructured data entries by calling the oracle $O(\sqrt{N})$ times. In contrast, classical algorithms typically require $O(N)$ oracle calls to find the target data item from N unstructured data in the average case and the worst case. This indicates that Grover's algorithm achieves a quadratic speedup in terms of oracle calls when compared to classical algorithms.

The quantum circuit of Grover's algorithm is shown in Figure 1. There are two major components in the circuit, namely the oracle and the diffusion operator (or diffuser), as described below. The oracle and diffuser form the so-called "Grover iterator" and typically iterate multiple times.

The oracle is represented as U_f in Figure 1 to identify the target input and invert its phase. It is defined as follows:

$$U_f |x\rangle = \begin{cases} -|x\rangle & \text{if } |x\rangle = |x^*\rangle \\ |x\rangle & \text{if } |x\rangle \neq |x^*\rangle \end{cases} \quad (1)$$

In Equation (1), $|x^*\rangle$ is the target input. The oracle flips the phase of an input $|x\rangle$ only when $|x\rangle = |x^*\rangle$ (that is, the input $|x\rangle$ is equal to the target input $|x^*\rangle$); otherwise, it does nothing.

The diffusion operator or diffuser was proposed by Grover in [12]. It is an inversion-about-amplitude-mean (IAAM) operation and was proven to be capable of achieving the probability amplitude inversion about the mean of the probability amplitudes of quantum states, as shown at the bottom of Figure 2. Please refer to [12] for the proof details.

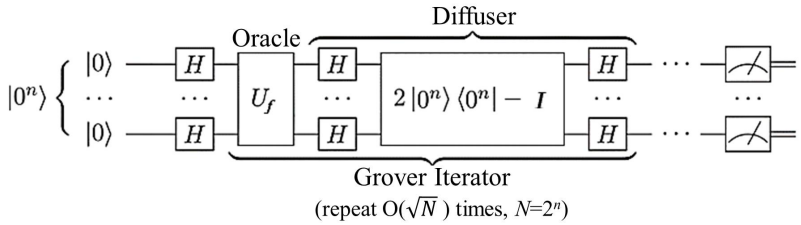

Figure 1. The quantum circuit of Grover's algorithm.

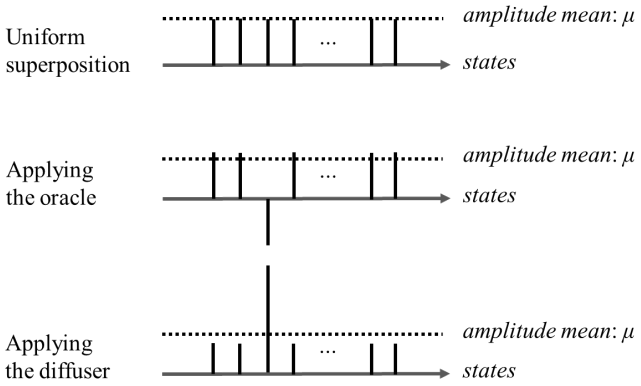

Figure 2. An illustration of the application of the diffusion operator (or diffuser) of Grover's algorithm.

In the following, we discuss the implementation of the diffuser, which is represented as U_s below.

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

In Equation (2), $|0^n\rangle$ represents the $n \times 1$ matrix $(1, 0, \ldots, 0)^{\mathrm{T}}$, $\langle 0^n|$ represents the $1 \times n$ matrix $(1, 0, \ldots, 0)$, $|0^n\rangle\langle 0^n|$ stands for the outer product of $(1, 0, \ldots, 0)^{\mathrm{T}}$ and $(1, 0, \ldots, 0)$, and I is the $n \times n$ identity matrix, with diagonal elements being 1 and other elements being 0. Note that the operation of $2|0^n\rangle\langle 0^n| - I$ can be achieved by applying the X gate (or NOT gate), the multi-controlled Z gate (or MCZ gate), and again the X gate to all input qubits, as will be employed in this paper to implement the quantum circuit of Grover's algorithm. Below, we explain why this implementation can realize $2|0^n\rangle\langle 0^n| - I$.

First, note that $|0^n\rangle\langle 0^n|$ represents the outer product of an $n \times 1$ matrix and a $1 \times n$ matrix, resulting in an $n \times n$ matrix, as shown below:

$$|0^n\rangle\langle 0^n| = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{3}$$

Thus, we can derive the following equation:

$$2|0^n\rangle\langle 0^n| - I = 2\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} - I$$

$$= \begin{pmatrix} 2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (4)$$

This transformation matrix applies a phase flip to all states except $|0^n\rangle$. If we multiply this transformation matrix by -1, the resulting matrix applies only a phase flip to the state $|0^n\rangle$, as shown in the matrix on the right-hand side below. Note that multiplying the transformation matrix by -1 results in a global phase flip, which is unobservable, does not affect the measurement outcomes for the final qubit states, and can therefore be ignored.

$$-1 \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

By applying X gates to each qubit to exchange the probability amplitudes of $|0\rangle$ and $|1\rangle$, the above matrix becomes one that applies a phase flip to the $|1^n\rangle$ state. This phase flip for $|1^n\rangle$ can be implemented using a multi-controlled Z gate [MCZ], where the target qubit is the last (or the most significant) qubit and all other qubits are control qubits. The transform matrix of the MCZ gate is shown below:

$$[MCZ] = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (6)$$

Finally, by applying X gates to each qubit again, we restore the original qubit states, except that a phase flip has been applied to the state $|0^n\rangle$.

Thus, we obtain the following transformation of $2|0^n\rangle\langle 0^n| - I$:

$$2|0^n\rangle\langle 0^n| - I = -1 \cdot X^{\otimes n}[MCZ]X^{\otimes n} \quad (7)$$

Note that multiplying the transformation matrix by -1 results in a global phase flip, which is unobservable. As it does not affect the measurement outcomes for the final qubits, it can be ignored. Therefore, we can construct the quantum circuit for the diffusion operator $U_s = H^{\otimes n}(2|0^n\rangle\langle 0^n| - I)H^{\otimes n}$ of n qubits as follows:

1. Apply an H gate to each of n qubits;
2. Apply an X gate to each of n qubits;
3. Apply a multi-controlled Z (MCZ) gate to n qubits, with the most significant qubit as the target qubit and all other qubits as control qubits;
4. Apply an X gate to each of n qubits;
5. Apply an H gate to each of n qubits.

Grover's algorithm has six major steps, each of which is elaborated below. Note that we assume that there are n working qubits to represent a total of $N = 2^n$ states

corresponding to $N = 2^n$ input instances. We also assume that there are a total of M target inputs or solutions, where $M \ll N$.

Step 1: Prepare n working qubits in the state $|0\rangle$ as input qubits, so that the input qubits are of the state $|0^n\rangle$ (or $|0\rangle^{\otimes n}$).
Step 2: All qubits undergo H gates to achieve a uniform superposition state, given by $H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle = |+\rangle^{\otimes n}$, where $N = 2^n$.
Step 3: All qubits in uniform superposition undergo the oracle to perform the phase inversion of the target inputs. Thus, the phase of every qubit state representing a target input is flipped and its amplitude becomes negative.
Step 4: All qubits undergo the diffuser to perform the inversion-about-amplitude-mean (IAAM) operation, so that the amplitude of every qubit state representing a target input becomes positive and larger than those of qubit states not representing target inputs.
Step 5: Repeat Step 3 and Step 4 a total of $\left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rfloor$ times, where $N = 2^n$ is the number of possible input instances, M is the number of target inputs or solutions, and $M \ll N$.
Step 6: Measure the qubits and take the states with significant occurrence probabilities as solutions.

In summary, Grover's algorithm iteratively modifies the states of the qubits, as shown in Figure 2. Initially, the qubit states exhibit uniform superposition. Then, the oracle is applied, which inverts the phase of the qubit states corresponding to solutions, creating negative probability amplitudes. This is followed by the diffuser, which performs the inversion-about-amplitude-mean operation. Thus, the amplitude of the qubit state corresponding to a solution becomes positive and significantly larger than the amplitudes of the qubit states that do not correspond to any solution. In fact, the amplitudes of the non-solution qubit states are decreased by the diffuser. After the qubits undergo a suitable number of iterations via the Grover iterator that contains the oracle and the diffuser, the amplitude of the qubit state corresponding to a solution becomes significantly large, while the amplitude of every non-solution qubit state becomes noticeably small or even approaches zero.

2.3. Quantum Counting Algorithm

According to [14], Grover's algorithm can find the target input with a high probability if the Grover iterator containing the oracle and the diffuser repeats the iteration $\left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rfloor$ times, where N is the total number of possible input instances and M is the number of target inputs (or solutions). However, we face the challenge in which the number of solutions is not known in advance. To address this challenge, this study utilizes the quantum counting algorithm [29,30] to estimate the number of solutions to the given problem.

The quantum counting algorithm (QCA) is based on the technique of quantum phase estimation (QPE) to derive the phase θ of the Grover iterator G to estimate the number M of solutions. The quantum circuit of the QCA is illustrated in Figure 3 [30] and explained below. In Figure 3, Register 1 is composed of t qubits, which are used as the counting qubits for QPE. Generally, increasing the number t of counting qubits allows for the more accurate estimation of the actual value of M. Moreover, Register 2 consists of $n + 1$ qubits, serving as the working qubits for the controlled Grover iterators, represented as $G^{2^0}, G^{2^1}, \ldots, G^{2^{t-1}}$ in Figure 3. The reason for using $n + 1$ working qubits is that the Grover iterator originally requires n qubits to form the space of 2^n input instances, along with one additional qubit serving as an auxiliary qubit for the phase kickback.

The inverse Fourier transform (denoted as $FT^{\dagger}$ in Figure 3) is finally applied to the t counting qubits to transform their states from the Fourier basis of $|+\rangle$ and $|-\rangle$ into the computation basis of $|0\rangle$ and $|1\rangle$. This is for the purpose of measuring the counting qubits.

After measuring the counting qubits, the measurement results can be used to derive the phase θ of the Grover iterator, where θ is represented as a radian within the range of $[-\pi, \pi]$. According to [30], the phase θ, the number M of solutions, and the total number N of input instances are related by the following Equation (8), which allows us to easily derive M from θ and N:

$$M = N \times \sin^2\left(\frac{\theta}{2}\right) \tag{8}$$

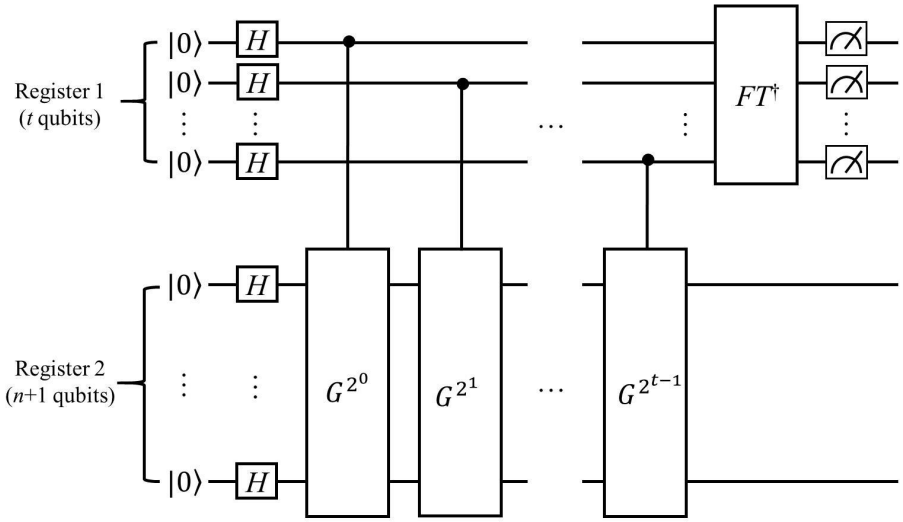

Figure 3. The quantum circuit of the quantum counting algorithm proposed in [29].

2.4. Quantum Counter

Heidari et al., in [26], proposed the quantum circuit of a quantum counter, as shown in Figure 4. The quantum counter consists of one control qubit labeled "control" and b counting qubits labeled $\text{count}_0,\ldots,\text{count}_{b-1}$, initialized with the counter value $|0\rangle^{\otimes b}$. The control qubit is to be fed with a series of qubit states. If the fed state is $|1\rangle$, the counter value is incremented by 1. In contrast, if the fed state is $|0\rangle$, the counter value remains unchanged. Notably, when the counter value reaches $|1\rangle^{\otimes b}$, further increments reset the counter value to $|0\rangle^{\otimes b}$.

It can be observed that, in the quantum counter proposed by Heidari et al., an X gate is applied after each controlled X (CX) gate or multi-controlled X (MCX) gate, and $b-1$ X gates are appended at the bottom. This not only increases the number of quantum gates but also adds to the depth of the quantum circuit. Later, we will demonstrate how to avoid the need to add all the X gates, while maintaining the proper function of the quantum counter.

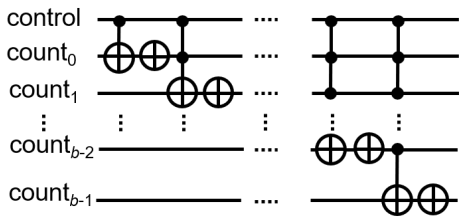

Figure 4. The quantum circuit of the controlled quantum counter proposed in [26].

3. Related Work

Numerous research papers in the literature [15–23] have proposed using quantum circuits based on Grover's algorithm to address various NP-hard problems. The solved problems include the k-coloring problem, the maximum clique problem, the list coloring problem, the pure Nash equilibrium finding problem in graphical games, the Hamiltonian

cycle problem, the vertex cover problem, and the dominating set problem. Table 1 presents the relevant information from these papers.

Table 1. Information from research papers using quantum circuits based on Grover's algorithm to address various NP-hard problems.

Paper	Year	Authors	Problem Solved
[15]	2020	Saha et al.	k-coloring
[16]	2021	Saha et al.	k-coloring in d-ary quantum systems
[17]	2021	Haverly and Lopez	maximum clique
[18]	2022	Mukherjee	list coloring
[19]	2022	Roch et al.	pure Nash equilibria in graphical games
[20]	2022	Jiang	Hamiltonian cycle
[21]	2023	Jiang and Kao	Hamiltonian cycle
[22]	2023	Jiang and Yan	vertex cover
[23]	2023	Jiang and Lin	dominating set

Below, we briefly describe the research results of the papers listed in Table 1. In [17], Haverly and Lopez implemented a quantum circuit based on Grover's algorithm using IBM Qiskit packages to address the maximum clique problem, i.e., to find the largest fully connected subgraph of a given graph. The implementation is based on the descriptions mentioned by Bojic in [31]. Since the maximum clique of a graph may not be unique, and the number of such cliques is unknown, Bojic employed the method proposed by Boyer et al. [14] to devise steps to identify at least one maximum clique in a given graph under the condition that the number of solutions is unknown. These steps are based on Grover's algorithm, but Bojic did not demonstrate their implementation. In contrast, Haverly and Lopez implemented their corresponding quantum circuit using IBM Qiskit packages and verified that this quantum circuit can indeed produce the maximum cliques of an example graph.

In [15], Saha et al. designed a quantum circuit based on Grover's algorithm to address the k-coloring problem. Given a graph $G = (V, E)$ with the vertex set V and the edge set E, the k-coloring problem is to assign one of k different colors to each vertex in the graph, such that no two adjacent vertices (i.e., vertices connected by an edge) share the same color. The authors proposed the quantum comparator concept to design the oracle of Grover's algorithm to address the k-coloring problem. They also proposed a generalized algorithm to synthesize oracle quantum circuits to address the k-coloring problem for any unweighted and undirected graph. The algorithm's output is a quantum circuit netlist in the form of the quantum assembly (QASM) language, which can be applied to general NISQ devices. For the purpose of validation, the authors also executed the synthesized oracle quantum circuit on the IBMQ cloud [32] to correctly solve the k-coloring problem for an example graph.

In [16], Saha et al. designed a quantum circuit based on Grover's algorithm to address the k-coloring problem in d-ary quantum systems with d possible states, where $d \geq 2$. Saha et al. showed that d-state quantum systems use fewer quantum d-state digits to build quantum comparators than binary quantum systems. They also proposed a generalized algorithm that can synthesize oracle quantum circuits in d-ary quantum systems to address the k-coloring problem for a given graph. Finally, the authors used a MATLAB (R2012b) simulator [33] to simulate a synthesized quantum circuit on a 3-ary quantum system using seven qutrits (i.e., quantum three-state digits), verifying that it can generate the correct k coloring for the given graph. Note that, like a qubit that represents a binary quantum system with two possible states denoted as $|0\rangle$ and $|1\rangle$, a qutrit represents a 3-ary quantum system with three possible states, typically denoted as $|0\rangle$, $|1\rangle$, and $|2\rangle$.

In [18], Mukherjee proposed a quantum circuit based on Grover's algorithm, utilizing a restricted search space to address the list coloring problem for a given graph, where each vertex is assigned a list of permissible colors. The objective of the list coloring problem is to find a valid or proper coloring such that adjacent vertices have different colors, while satisfying the constraint to choose colors from their respective lists. This additional constraint makes the problem more complex than the standard graph coloring problem, as it restricts the set of feasible solutions that need to be explored. Mukherjee used a restricted version of Grover's algorithm [34] to deal with the constraint to search only within the feasible region defined by the color lists. The proposed quantum circuit was constructed with Python using blueqat-sdk and was simulated on the Amazon Statevector Simulator for the purpose of validation.

In [19], Roch et al. designed a quantum circuit based on Grover's algorithm to find the pure Nash equilibrium (PNE) in graphical games. A graphical game is a specialized model in game theory where a player's interactions and payoffs depend only on their neighbors. In graphical games, the PNE is a situation where each player selects a single strategy that maximizes their payoff for given strategies of their neighbors, and no player can improve their payoff by unilaterally changing their strategy. The PNE involves pure strategies rather than mixed strategies. The former strategies are simpler and allow each player to explicitly decide on a particular action, whereas the latter allow each player to mix several pure strategies according to a probability distribution. However, determining the existence of the PNE is NP-complete (and thus also NP-hard) [35]. Roch et al. created quantum circuits of the Grover algorithm oracle from a given graphical game by first translating it into a Boolean satisfiability (SAT) problem, asking if there exists a Boolean assignment to variables to satisfy a given Boolean formula. Afterwards, the oracle quantum circuit was built using the methods proposed in [36,37]. Experiments based on IBM Qiskit built-in functions were performed to validate the proposed quantum circuits.

In [20], Jiang proposed the concept of an "explicit oracle" to implement the quantum circuit of Grover's algorithm to address the Hamiltonian cycle problem. Given a graph, the Hamiltonian cycle problem is to find a cycle that goes through every vertex once and exactly once. Jiang classified the oracles as implicit and explicit. He defined an oracle as implicit if it checks whether an input instance meets the solution conditions to determine whether the instance is a solution. In contrast, he defined an oracle as explicit if it directly compares an input instance against specific known patterns to determine whether it is a solution. That is, the explicit oracle is built with knowledge of explicit solution patterns in advance, so the quantum circuit can be built easily by inverting the phase of the qubit states associated with the known solution patterns. However, the quantum circuit is based on known solution patterns, so it cannot be applied to the solution of practical application problems whose solution patterns are not known in advance. The main goal of the quantum circuit is to demonstrate the execution process of Grover's algorithm for educational purposes. The paper presents two quantum circuits that are built and executed with IBM Qiskit packages to validate their correctness in generating Hamiltonian cycles for fully connected graphs of four and five vertices.

In [21], Jiang and Kao built the oracle quantum circuit of Grover's algorithm to generate all possible solutions to the Hamiltonian cycle problem for a given undirected graph. Their concept is to implement the quantum circuit of an implicit oracle to check if the edge set S associated with a quantum state meets the following conditions: (C1) if the number of edges included in S equals the number of vertices of the graph, (C2) if each graph vertex is connected to exactly two edges included in S, and (C3) if the edges included in S form a single cycle. A quantum circuit was built and executed using IBM

Qiskit packages to validate its correctness in generating the Hamiltonian cycle for a fully connected graph of three vertices.

In [22], Jiang and Yan proposed a novel quantum circuit design called "quantum semaphore" to construct the oracle quantum circuit of Grover's algorithm to address the vertex cover problem for a given graph and an integer k. The vertex cover problem asks whether, in a given undirected graph $G(V, E)$, a vertex cover of size $k' \leq k$ exists. That is, it checks if there exists a set $V' \subseteq V$ of vertices such that every edge in the graph has at least one of its vertices in V', and the size of V' is less than or equal to k. The original vertex cover problem is a decision problem. However, Jiang and Yan's quantum circuit can generate all vertex covers of size k for a given graph and an integer k. Quantum circuits were constructed and executed with IBM Qiskit packages to validate their correctness in generating all vertex covers for an example graph of five vertices with k = 2 and 3, respectively.

In [23], Jiang and Lin proposed a quantum circuit design that utilizes quantum counters to construct the oracle of Grover's algorithm to address the dominating set problem. The dominating set problem asks whether, in a given undirected graph $G(V, E)$, there exists a dominating set $D \subseteq V$ such that (C1) every vertex in V is either in D or adjacent to at least one vertex in D and (C2) the size of D is equal to a specified integer k. Jiang and Lin's approach focuses on generating all dominating sets of a given size for a specified graph, rather than merely deciding the existence of such dominating sets. Their quantum circuits use quantum counters to check if (C1) and (C2) are satisfied. They were developed and validated with IBM Qiskit packages. The correctness was demonstrated by finding all dominating sets of size k for an example graph.

4. Proposed Quantum Circuit Design

Figure 5 shows the proposed quantum circuit design based on Grover's algorithm to address the ECP. The details of the quantum circuit are described in the following subsections, including the qubit initialization, the simplified quantum counter, and the oracle construction based on the simplified quantum counter.

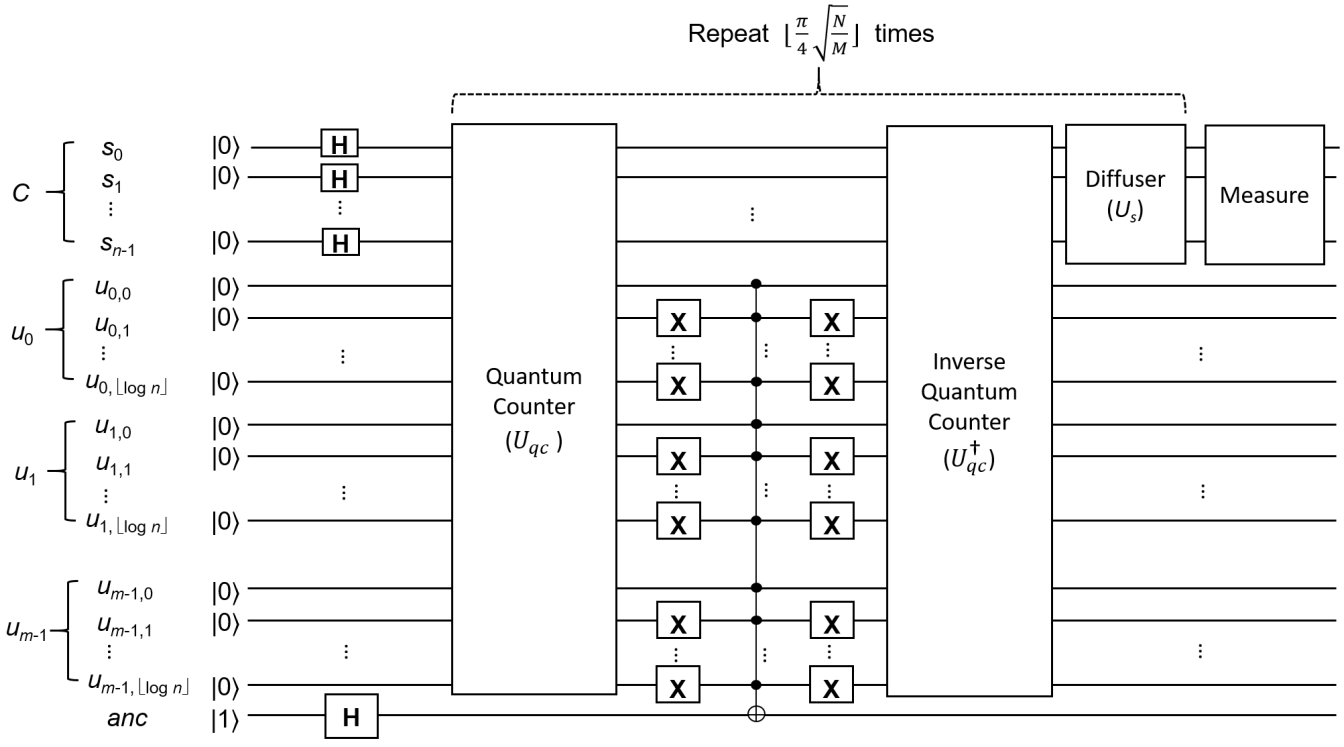

Figure 5. The proposed quantum circuit design to address the ECP.

4.1. Initialization

Figure 5 shows the quantum circuit design to address the ECP with a given universal set $U = \{u_0, \ldots, u_m\}$ of m elements and a given collection $C = \{s_0, s_1, \ldots, s_{n-1}\}$ of n sets, each of which is a subset of U. In Figure 5, qubits labeled with $s_0, s_1, \ldots, s_{n-1}$ are used to represent subsets in the collection C, and each of the qubits labeled with $u_{i,0}, u_{i,1}, \ldots, u_{i,b}$ is used to represent the quantum counter associated with the element u_i in the universal set U to store the number of sets covering u_i, where $0 \leq i \leq m-1$ and $b = \lfloor \log n \rfloor + 1$ is the number of qubits used in a quantum counter. Finally, a qubit that is initially $|1\rangle$ and then undergoes an H gate to be transformed into $|-\rangle$ is used as an ancilla qubit, which is the target qubit of a multi-controlled X (MCX) gate to trigger the phase kickback. In summary, the quantum circuit uses $n + m(\lfloor \log n \rfloor + 1)$ qubits.

4.2. Simplified Quantum Counter

We use a simplified quantum counter, as shown in Figure 6, to implement the quantum circuit of the oracle. In the quantum counter circuit, a qubit labeled with "control" serves as the control qubit, and qubits labeled with $\text{count}_0, \ldots, \text{count}_{b-1}$ are the b counting qubits. The quantum counter functions like an incrementer, where the counter's value increases by 1 whenever the control qubit is set to $|1\rangle$.

Compared to the existing quantum counter proposed by Heidari et al. in [26], the simplified quantum counter is noticeably simpler in design. Furthermore, it utilizes fewer quantum gates and has a smaller quantum circuit depth. Specifically, for a quantum counter with b counting qubits, the simplified quantum counter decreases the number of X gates by $2b - 2$ and reduces the quantum circuit depth by b.

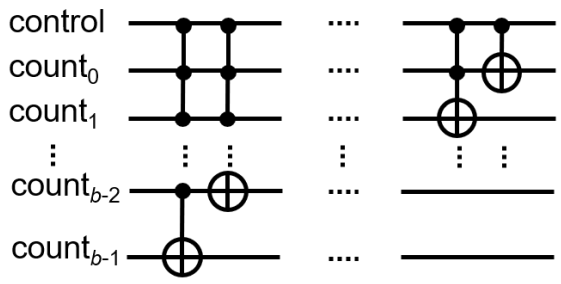

Figure 6. The quantum circuit of the controlled quantum counter of b counting qubits.

4.3. Oracle

As shown in Figure 5, the oracle uses a controlled quantum counter gate denoted as U_{qc}, an inverse of the controlled quantum counter gate denoted as $U_{qc}^{\dagger}$, and some X gates and an MCX gate between U_{qc} and $U_{qc}^{\dagger}$. In order to determine whether each element in U is covered by exactly one set in C, the oracle sets up a quantum counter for each element in U to count how many times it is covered by sets in C. Afterward, the oracle uses an MCX gate to invert the ancilla qubit if every counter corresponds to the value 1. Specifically, the MCX gate has to control qubits from every quantum counter, with X gates being added to all qubits of the quantum counter except the least significant qubit. This means that if every quantum counter is of the binary value $0 \ldots 1$ (i.e., value 1), then the ancilla qubit is intended to be flipped. Note that, since the ancilla qubit is in state $|-\rangle$, an eigenstate of the X gate, the phase kickback occurs to affect the qubits of the quantum counters. We will explain the phase kickback effect in detail below.

Since the circuit of the oracle and the diffuser is repeated multiple times or iterations, it is necessary to reset the counting qubits of every quantum counter before the next iteration starts. This can be achieved by employing the inverse of the controlled quantum counter gate, i.e., the $U_{qc}^{\dagger}$, as shown in Figure 5. The reader can check that, in U_{qc}, the qubits labeled

with $s_0, s_1, \ldots, s_{n-1}$ have no relation to the ancilla qubit. However, when all the X gates directly after the MCX gate and the $U_{qc}^{\dagger}$ gate are executed, the quantum counter qubits that were originally affected by the MCX gate's phase kickback will start affecting qubits labeled with $s_0, s_1, \ldots, s_{n-1}$.

The quantum circuits of both U_{qc} and $U_{qc}^{\dagger}$ use m quantum counters. Hence, the whole quantum circuit of the oracle uses $2m$ quantum counters, each of which has $b = \lfloor \log n \rfloor + 1$ counting qubits. Thus, using simplified quantum counters to replace the existing quantum counters proposed by Heidari et al. [26] can reduce the number of quantum gates by $2m(2b-2)$ and shorten the quantum circuit depth by $2mb$ for a single oracle. The overall quantum circuit invokes the oracle $\lfloor \pi/4\sqrt{N/M} \rfloor$ times to address the ECP, where $N = 2^n$ is the total number of possible input instances and M is the number of solutions to the ECP, $1 \leq M \ll N$. Consequently, the overall quantum circuit using the simplified quantum counter instead of existing quantum counters can save $(4mb - 4m)\lfloor \pi/4\sqrt{N/M} \rfloor$ quantum gates, and its circuit depth is reduced by $(2mb)\lfloor \pi/4\sqrt{N/M} \rfloor$.

4.4. Quantum Circuit Generation Algorithm

This subsection shows the algorithm, called Grover-ECP, used to generate a quantum circuit based on Grover's algorithm to tackle the ECP by finding all exact covers for a given universal set $U = \{u_0, \ldots, u_m\}$ of m elements and a given collection $C = \{s_0, \ldots, s_{n-1}\}$ of n sets, where each set in C is a subset of U.

As shown in Algorithm 1, the Grover-ECP algorithm first prepares a quantum circuit QC with n working qubits $s_0, \ldots, s_{n-1}$ in state $|0\rangle$, a total of $m \cdot b$ counting qubits $u_{0,0}, \ldots, u_{0,b-1}, \ldots, u_{m-1,0}, \ldots, u_{m-1,b-1}$ (or $cnt_0, \ldots, cnt_{mb-1}$) of counters in state $|0\rangle$, and an ancilla qubit labeled by *anc* in state $|1\rangle$ and n classical bits to store the measurement results of the n working qubits $s_0, \ldots, s_{n-1}$. Then, the n working qubits $s_0, \ldots, s_{n-1}$ undergo H gates to ensure the uniform superposition of 2^n states of the search space.

Afterwards, the quantum circuit repeats the Grover iterator $\lfloor (\pi/4)\sqrt{(N/M)} \rfloor$ times (iterations). At each iteration, a controlled quantum counter gate U_{qc} is added to qubits $s_0, \ldots, s_{n-1}$ and $u_{0,0}, \ldots, u_{0,b-1}, \ldots, u_{m-1,0}, \ldots, u_{m-1,b-1}$. Then, X gates are added to qubits $u_{0,1}, \ldots, u_{0,b-1}, \ldots, u_{m-1,1}, \ldots, u_{m-1,b-1}$. Note that no X gates are added to the qubits $u_{0,0}, \ldots, u_{m-1,0}$, each of which is the first counting qubit of the quantum counter associated with every element. Furthermore, the quantum circuit is appended by an MCX gate that takes qubits $u_{0,0}, \ldots, u_{0,b-1}, \ldots, u_{m-1,0}, \ldots, u_{m-1,b-1}$ as control qubits and the *anc* qubit as the target qubit. Then, X gates are added to qubits $u_{0,1}, \ldots, u_{0,b-1}, \ldots, u_{m-1,1}, \ldots, u_{m-1,b-1}$. The inverse quantum counter gate $U_{qc}^{\dagger}$ is added to qubits $s_0, \ldots, s_{n-1}$ and $u_{0,0}, \ldots, u_{0,b-1}, \ldots, u_{m-1,0}, \ldots, u_{m-1,b-1}$. The diffuser gate U_s is added to qubits $s_0, \ldots, s_{n-1}$. After $\lfloor (\pi/4)\sqrt{(N/M)} \rfloor$ repetitions of the Grover iterator, the n working qubits $s_0, \ldots, s_{n-1}$ are measured to derive the solution to the given ECP.

The Grover-ECP algorithm invokes other algorithms to generate the quantum circuits of the quantum counter (Algorithm 2), the inverse quantum counter (Algorithm 3), and the diffuser (Algorithm 4). These algorithms are also shown below to reveal the details of the quantum circuits. It is noted that the inverse quantum counter executes all the gates used in the quantum counter in reverse order. This has the effect of resetting all counters to the initial value of 0, so that the counters can count from 0 in the next iteration.

Algorithm 1 Grover-ECP(U, C)

Input: An exact cover problem (ECP) instance:
$U = \{u_0, \ldots, u_{m-1}\}$, a universal set of m elements
$C = \{s_0, \ldots, s_{n-1}\}$, a collection of n sets, each of which is a subset of U
Output: QC: a quantum circuit using QQSA to address the ECP

$b \leftarrow \lfloor \log n \rfloor + 1$
$QC \leftarrow$ a quantum circuit with
n working qubits $s_0, \ldots, s_{n-1}$ in $|0\rangle$,
mb counter qubits $u_{0,0}, \ldots, u_{0,b-1}, \ldots, u_{m-1,0}, \ldots, u_{m-1,b-1}$ (or $cnt_0, \ldots, cnt_{mb-1}$) in $|0\rangle$, and
1 ancilla qubit anc in $|1\rangle$
Add H gates on n qubits $s_0, \ldots, s_{n-1}$
for $i \leftarrow 1$ to $\lfloor (\pi/4)\sqrt{(N/M)} \rfloor$ **do**
  $U_{qc} \leftarrow$ Quantum_Counter(n, m)
  Add U_{qc} gate on qubits $s_0, \ldots, s_{n-1}$ and $u_{0,0}, \ldots, u_{0,b-1}, \ldots, u_{m-1,0}, \ldots, u_{m-1,b-1}$
  Add X gates on qubits $u_{0,1}, \ldots, u_{0,b-1}, \ldots, u_{m-1,1}, \ldots, u_{m-1,b-1}$
  Add an MCX gate taking qubits $u_{0,0}, \ldots, u_{0,b-1}, \ldots, u_{m-1,0}, \ldots, u_{m-1,b-1}$ as control qubits and anc qubit as the target qubit
  Add X gates on qubits $u_{0,1}, \ldots, u_{0,b-1}, \ldots, u_{m-1,1}, \ldots, u_{m-1,b-1}$
  $U_{qc}^{\dagger} \leftarrow$ Inverse_Quantum_Counter(n, m)
  Add $U_{qc}^{\dagger}$ gate on qubits $s_0, \ldots, s_{n-1}$ and $u_{0,0}, \ldots, u_{0,b-1}, \ldots, u_{m-1,0}, \ldots, u_{m-1,b-1}$
  $U_s \leftarrow$ Diffuser(n)
  Add U_s gate on qubits $s_0, \ldots, s_{n-1}$
end for
Measure qubits $s_0, \ldots, s_{n-1}$
return QC

Algorithm 2 Quantum_Counter(n, m)

Input: Integers n and m
Output: QC: a quantum circuit of the quantum counter

$b \leftarrow \lfloor \log n \rfloor + 1$
$QC \leftarrow$ a quantum circuit with
n working qubits $s_0, \ldots, s_{n-1}$ and
mb counter qubits $u_{0,0}, \ldots, u_{0,b-1}, \ldots, u_{m-1,0}, \ldots, u_{m-1,b-1}$ (or $cnt_0, \ldots, cnt_{mb-1}$)
for $i \leftarrow 0$ to $n-1$ **do**
  for $u_j \in s_i$ with ascending index order of j **do**
    for $k \leftarrow b-1$ downto 1 **do**
      Add an MCX gate taking s_i and $u_{j,0}, \ldots, u_{j,k-1}$ as control qubits and $u_{j,k}$ as the target qubit
    end for
    Add a CX gate taking s_i as the control qubit and $u_{j,0}$ as the target qubit
  end for
end for
return QC

Algorithm 3 Inverse_Quantum_Counter(n, m)

Input: Integers n and m
Output: QC: a quantum circuit of the inverse quantum counter
1: $b \leftarrow \lfloor \log n \rfloor + 1$
2: $QC \leftarrow$ a quantum circuit with
n working qubits $s_0, \ldots, s_{n-1}$ and
mb counter qubits $u_{0,0}, \ldots, u_{0,b-1}, \ldots, u_{m-1,0}, \ldots, u_{m-1,b-1}$ (or $cnt_0, \ldots, cnt_{mb-1}$)
3: **for** $i \leftarrow n-1$ downto 0 **do**
4: **for** $u_j \in s_i$ with descending index order of j **do**
5: **Add** a CX gate taking s_i as the control qubit and $u_{j,0}$ as the target qubit
6: **for** $k \leftarrow 1$ to b -1 **do**
7: **Add** an MCX gate taking s_i and $u_{j,0}, \ldots, u_{j,k-1}$ as control qubits and $u_{j,k}$ as the target qubit
8: **end for**
9: **end for**
10: **end for**
11: **return** QC

Algorithm 4 Diffuser(n)

Input: An integer n
Output: QC: a quantum circuit of the diffuser
1: $QC \leftarrow$ a quantum circuit with n qubits $q_0, \ldots, q_{n-1}$
2: **Add** H gates on qubits $q_0, \ldots, q_{n-1}$
3: **Add** X gates on qubits $q_0, \ldots, q_{n-1}$
4: **Add** an MCZ gate with $q_0, \ldots, q_{n-2}$ as control qubits and q_{n-1} as the target qubit
5: **Add** X gates on qubits $q_0, \ldots, q_{n-1}$
6: **Add** H gates on qubits $q_0, \ldots, q_{n-1}$
7: **return** QC

5. Experimental Results

We conduct four experiments based on the IBM Qiskit packages to implement the proposed quantum circuit design to address the ECP. The details of the experiments are described in the following subsections.

5.1. The First Experiment

The first experiment is to construct and run a quantum circuit to address the ECP defined below. The universal set U is $\{u_0, u_1, u_2\}$ of three elements u_0, u_1 and u_0, and the collection C of sets is $\{s_0, s_1\}$, where s_0 and s_1 are subsets of U, $s_0 = \{u_0\}$, and $s_1 = \{u_1\}$.

Figure 7 depicts the quantum circuit of the Grover iterator that contains the oracle and the diffuser to address the given ECP. The oracle has two qubits s_0 and s_1 for sets s_0 and s_1 in C, and it has three quantum counters for elements u_0, u_1 and u_2, respectively. Specifically, each quantum counter uses two qubits, where cnt_0 and cnt_1 are with the counter for element u_0; cnt_2 and cnt_3 are with the counter for element u_1; and cnt_4 and cnt_5 are with the counter for element u_2. Furthermore, the oracle also has an ancilla qubit that is denoted as *acn* and is used for phase kickback, which will be explained later. It is noted that the Grover iterator uses nine qubits.

Figure 8 shows the quantum circuit of the quantum counting algorithm (QCA) using the Grover iterator to derive the number of solutions to the ECP in the first experiment. The QCA quantum circuit uses three counting qubits cq_0, cq_1, and cq_2, each serving as the control qubit of the controlled Grover iterator (CG, or G for short). Specifically, qubit cq_0 is the control qubit of a G (or G^1 or G^{2^0}), cq_1 is the control qubit of two Gs (or G^2 or G^{2^1}), and cq_2 is the

control qubit of four Gs (or G^4 or G^{2^2}). The counting qubits initially undergo H gates to be in superposition and in the Fourier basis and then undergo an inverse Fourier transform ($QFT^\dagger$) gate to return to the computational basis for measurement to derive the phase of the Grover iterator. Note that the quantum circuit has nine working qubits, $wq_0, \ldots, wq_8$, which correspond to the nine qubits used by the Grover iterator. The working qubits wq_0 and wq_1 correspond to the two qubits s_0 and s_1 in the Grover iterator; they undergo H gates to be in superposition initially. The last working qubit wq_8 corresponds to the ancilla qubit of the Grover iterator; it undergoes an X gate and an H gate to be in the state $|-\rangle$ initially.

Figure 9 shows the histogram of the measurement results of the QCA to derive the number of solutions to the ECP in the first experiment. Outcome 100 has the measurement probability of 1. It corresponds to the value of 4. By dividing 2^3, where 3 is the number of counting qubits, 4 becomes $\frac{4}{2^3}$, which in turn is mapped into a phase angle in radians within the range $[-\pi, \pi]$. Specifically, it is mapped into the phase angle $2\pi \times \frac{4}{2^3} - \pi = \frac{8\pi}{8} - \pi = 0$. With Equation (8), we can derive $M = N \times \sin^2\left(\frac{\theta}{2}\right) = 4 \times \sin^2 0 = 0$. Therefore, using the quantum counting algorithm, we can determine that the ECP in the first experiment has no solution. We thus do not need to address the ECP in the first experiment further, as it has no solution.

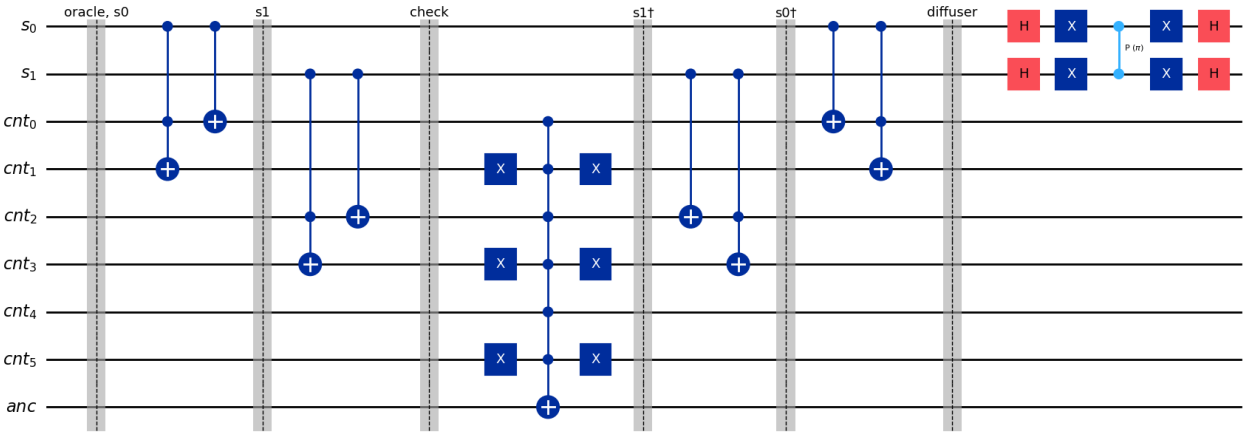

Figure 7. The quantum circuit of the Grover iterator (i.e., the oracle and the diffuser) to address the ECP in the first experiment.

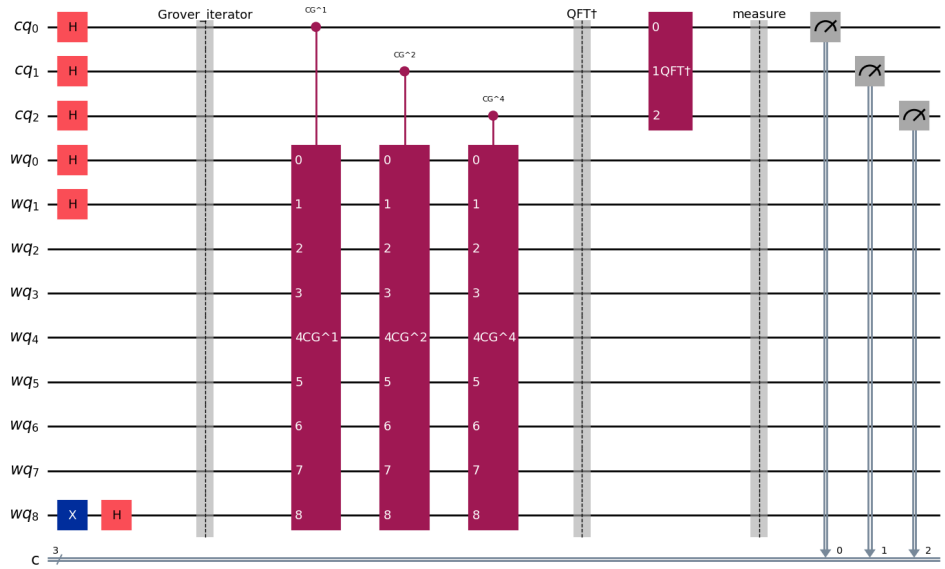

Figure 8. The quantum circuit of the quantum counting algorithm using the Grover iterator to derive the number of solutions to the ECP in the first experiment.

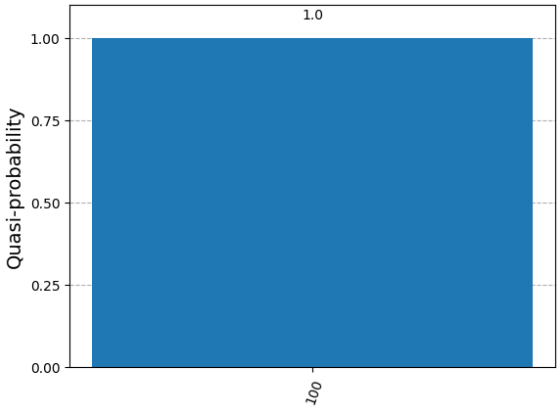

Figure 9. The histogram of the measurement results of the quantum counting algorithm to derive the number of solutions to the ECP in the first experiment.

5.2. The Second Experiment

The second experiment is to construct and run a quantum circuit to address the ECP defined below. The universal set U is $\{u_0, u_1\}$ of two elements u_0 and u_1, and the collection C of sets is $\{s_0, s_1\}$, where s_0 and s_1 are subsets of U, $s_0 = \{u_0\}$, and $s_1 = \{u_1\}$.

Figure 10 shows the quantum circuit of the Grover iterator that contains the oracle and the diffuser to address the given ECP. The oracle has two qubits s_0 and s_1 for sets s_0 and s_1 in C, and it has two counters for elements u_0 and u_1, respectively. Specifically, each counter uses two qubits, where cnt_0 and cnt_1 are with the counter for element u_0, and cnt_2 and cnt_3 are with the counter for element u_1. Furthermore, the oracle also has an ancilla qubit that is denoted as acn and is used for phase kickback, which will be explained later. It is noted that the Grover iterator uses seven qubits.

Figure 11 shows the quantum circuit of the quantum counting algorithm (QCA) using the Grover iterator to derive the number of solutions to the ECP in the second experiment. The QCA quantum circuit uses three counting qubits $cq_0, \ldots, cq_2$, each serving as the control qubit of the controlled Grover iterator (CG, or G for short). Specifically, qubit cq_0 is the control qubit of a G (or G^1 or G^{2^0}), cq_1 is the control qubit of two Gs (or G^2 or G^{2^1}), and cq_2 is the control qubit of four Gs (or G^4 or G^{2^2}). The counting qubits initially undergo H gates to be in superposition and in the Fourier basis and then undergo an inverse Fourier transform ($QFT^\dagger$) gate to return to the computation basis for measurement to derive the phase of the Grover iterator. Note that the quantum circuit has seven working qubits, $wq_0, \ldots, wq_6$, which correspond to the seven qubits used by the Grover iterator. The working qubits wq_0 and wq_1 correspond to the two qubits s_0 and s_1 in the Grover iterator; they undergo H gates and are in superposition initially. The last working qubit wq_6 corresponds to the ancilla qubit of the Grover iterator; it undergoes an X gate and an H gate to be in the state $|-\rangle$ initially.

Figure 12 shows the histogram of the measurement results of the QCA to derive the number of solutions to the ECP in the second experiment. Two outcomes, 011 and 101, have significant probabilities. They correspond to the values of 3 and 5, respectively. By dividing 2^3, where 3 is the number of counting qubits, 3 and 5 become $\frac{3}{2^3}$ and $\frac{5}{2^3}$, which in turn are mapped into phase angles in radians within the range $[-\pi, \pi]$. Specifically, they are mapped into the phase angle $2\pi \times \frac{3}{2^3} - \pi = \frac{6\pi}{8} - \pi = -\frac{2\pi}{8}$ and the phase angle $2\pi \times \frac{5}{2^3} - \pi = \frac{10\pi}{8} - \pi = \frac{2\pi}{8}$. With Equation (8), we can derive $M = N \times \sin^2\left(\frac{\theta}{2}\right) = 4 \times \sin^2\left(\frac{\pm 2\pi}{8}\right) = 0.58578$, which is rounded as 1. That is, using the quantum counting algorithm, we can determine that the ECP in the second experiment has $M = 1$ solution.

Figure 13 shows the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the second experiment. Note that the Grover iterator is repeated $T = \lfloor \pi/4\sqrt{N/M} \rfloor = \lfloor \pi/4\sqrt{4/1} \rfloor = \lfloor 1.57079 \rfloor = 1$ time.

Figure 14 shows the histogram of the measurement results of the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the second experiment. There is only one outcome of pattern 11 with a significant probability of occurrence. Outcome 11 corresponds to the sets $s_0 = \{u_0\}$ and $s_1 = \{u_1\}$, and the subcollection $\{s_0, s_1\}$ of C is exactly the only solution to the ECP in the second experiment.

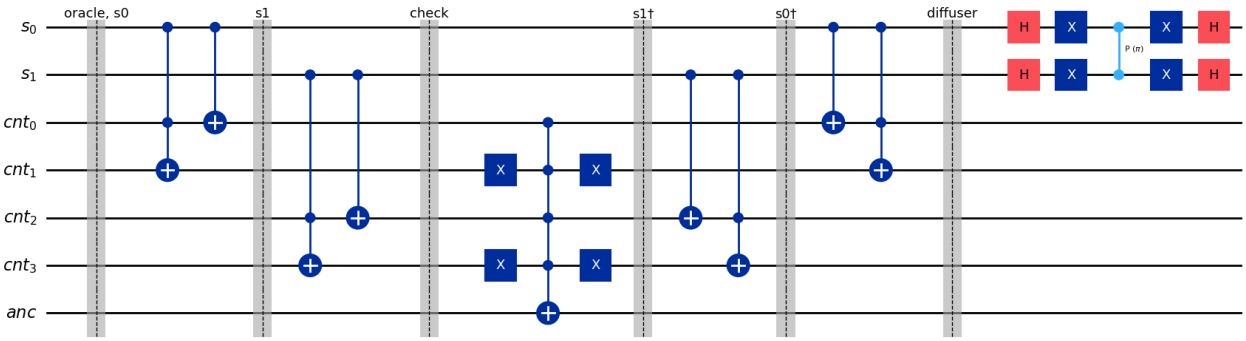

Figure 10. The quantum circuit of the Grover iterator (i.e., the oracle and the diffuser) to address the ECP in the second experiment.

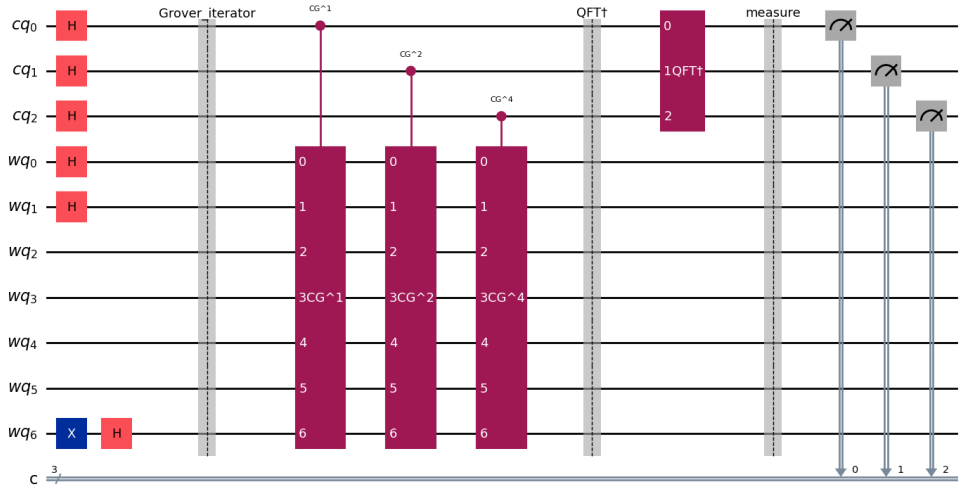

Figure 11. The quantum circuit of the quantum counting algorithm using the Grover iterator to derive the number of solutions to the ECP in the second experiment.

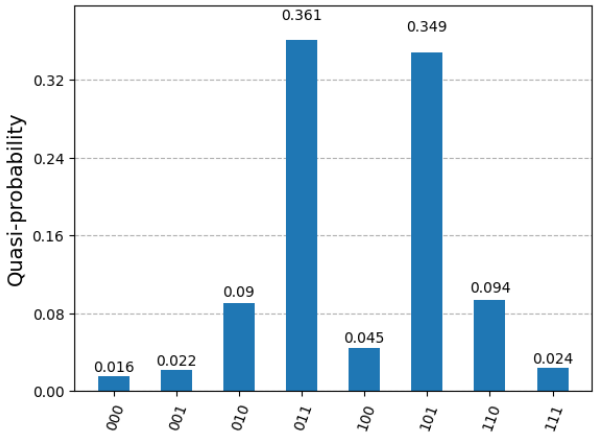

Figure 12. The histogram of the measurement results of the quantum counting algorithm to derive the number of solutions to the ECP in the second experiment.

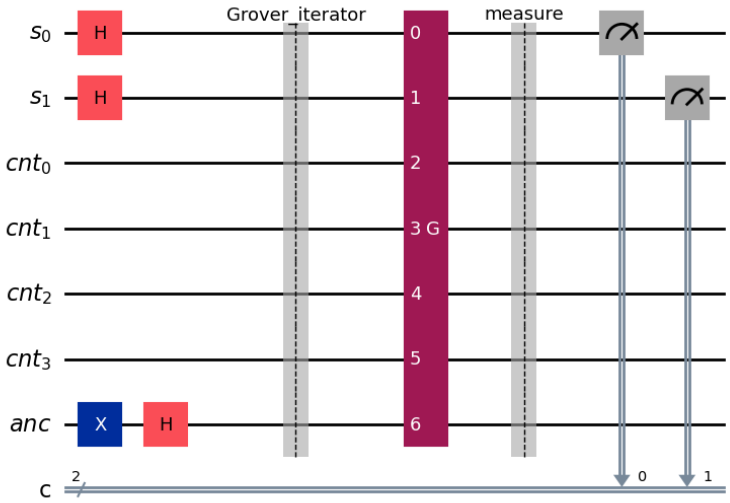

Figure 13. The quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the second experiment.

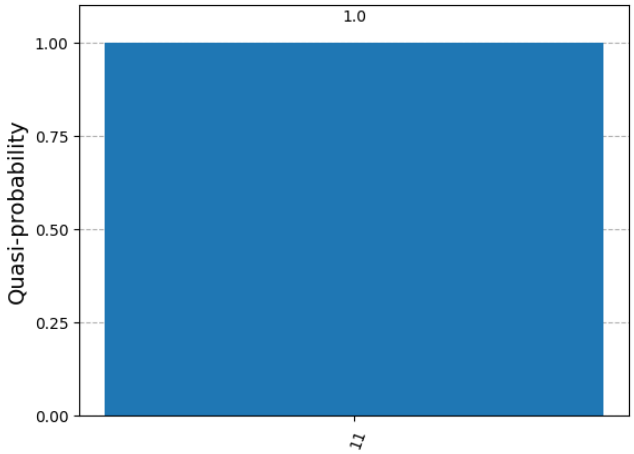

Figure 14. The histogram of the measurement results of the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the second experiment.

5.3. The Third Experiment

The third experiment is to construct and run a quantum circuit to address the ECP defined below. The universal set U is $\{u_0, u_1, u_2\}$ of three elements u_0, u_1 and u_2, and the collection C of sets is $\{s_0, s_1, s_2, s_3\}$, where s_0, s_1 and s_2 are subsets of U, $s_0 = \{u_0\}$, $s_1 = \{u_1\}$, $s_2 = \{u_2\}$ and $s_3 = \{u_0, u_1, u_2\}$.

Figure 15 depicts the quantum circuit of the Grover iterator that contains the oracle and the diffuser to address the given ECP. The oracle has four qubits $s_0, \ldots, s_3$ for sets $s_0, \ldots, s_3$ in C, and it has three counters for elements $u_0, \ldots, u_2$. Specifically, each counter uses three qubits, where $cnt_0, \ldots, cnt_2$ are with the counter for element u_0; $cnt_3, \ldots, cnt_5$ are with the counter for element u_1; and $cnt_6, \ldots, cnt_8$ are with the counter for element u_2. Furthermore, the oracle also has an ancilla qubit that is denoted as acn and is used for phase kickback. It is noted that the Grover iterator uses a total of 14 qubits.

Figure 16 shows the quantum circuit of the quantum counting algorithm using the Grover iterator to derive the number of solutions of the ECP in the third experiment. The quantum circuit uses four counting qubits $cq_0, \ldots, cq_3$, each serving as the control qubit of the controlled Grover iterator (CG, or G for short). Specifically, qubit cq_0 is the control qubit of a G, G^1 or G^{2^0}; cq_1 is the control qubit of two Gs, G^2 or G^{2^1}; cq_2 is the control qubit

of four Gs, G^4 or G^{2^2}; and cq_3 is the control qubit of eight Gs, G^8 or G^{2^3}. The counting qubits initially undergo H gates to be in superposition and in the Fourier basis and then undergo an inverse Fourier transform ($QFT^\dagger$) gate to return to the computational basis for measurement to derive the phase of the Grover iterator. Note that the quantum circuit has 14 working qubits, $wq_0, \ldots, wq_{13}$, which correspond to the 14 qubits used by the Grover iterator. The working qubits $wq_0, \ldots, wq_3$ correspond to the four qubits $s_0, \ldots, s_3$ in the Grover iterator; they undergo H gates and are in superposition initially. The last working qubit wq_{13} corresponds to the ancilla qubit of the Grover iterator; it undergoes an X gate and an H gate to be in the state $|-\rangle$ initially.

Table 2 shows the detailed measurement results of the QCA to derive the number of solutions to the ECP in the third experiment. Figure 17 shows the histogram of the measurement results of the QCA to derive the number of solutions to the ECP in the third experiment. Two combinations or outcomes, 0110 and 1010, have significant measurement probabilities. They correspond to the values of 6 and 10, respectively. By dividing 2^4, where 4 is the number of counting qubits, 6 and 10 become $\frac{6}{2^4}$ and $\frac{10}{2^4}$, which in turn are mapped into phase angles in radians within the range $[-\pi, \pi]$. Specifically, they are mapped into the phase angle $2\pi \times \frac{6}{2^4} - \pi = \frac{12\pi}{16} - \pi = -\frac{4\pi}{16}$ and the phase angle $2\pi \times \frac{10}{2^4} - \pi = \frac{20\pi}{16} - \pi = \frac{4\pi}{16}$. With Equation (8), we can derive $M = N \times \sin^2\left(\frac{\theta}{2}\right) = 16 \times \sin^2\left(\frac{\pm 4\pi}{16}\right) = 2.34314$, which is rounded as 2. That is, using the quantum counting algorithm, we can determine that the ECP in the third experiment has $M = 2$ solutions.

Figure 18 shows the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the third experiment. Note that the Grover iterator is repeated $T = \lfloor \pi/4\sqrt{N/M} \rfloor = \lfloor \pi/4\sqrt{16/2} \rfloor = \lfloor 2.22144 \rfloor = 2$ times.

Table 3 shows the detailed measurement results (combination and probability) of the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the fourth experiment. Furthermore, Figure 19 shows the histogram of the measurement results of the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the third experiment. There are two combinations, 0111 and 1000, with significant measurement probabilities of occurrence. Outcome 0111 corresponds to the sets $s_0 = \{u_0\}$, $s_1 = \{u_1\}$, and $s_2 = \{u_2\}$, and the subcollection $\{s_0, s_1, s_2\}$ of C is a solution to the ECP in the third experiment. Similarly, outcome 1000 corresponds to the set $s_4 = \{u_0, u_1, u_2\}$, and the subcollection $\{s_4\}$ of C is the other solution to the ECP in the third experiment.

Table 2. The detailed measurement results (combination and probability) of the quantum counting algorithm to derive the number of solutions to the ECP in the third experiment.

Comb.	Prob.	Comb.	Prob.	Comb.	Prob.	Comb.	Prob.
00000	0.0008	00001	0.001	00010	0.0006	00011	0.0016
00100	0.0014	00101	0.0006	00110	0.001	00111	0.0022
01000	0.0022	01001	0.005	01010	0.0072	01011	0.02
01100	0.3574	01101	0.0768	01110	0.0162	01111	0.0064
10000	0.0038	10001	0.006	10010	0.0116	10011	0.081
10100	0.356	10101	0.0226	10110	0.0072	10111	0.0042
11000	0.0024	11001	0.001	11010	0.001	11011	0.0006
11100	0.0014	11101	0.0000	11110	0.0006	11111	0.0002

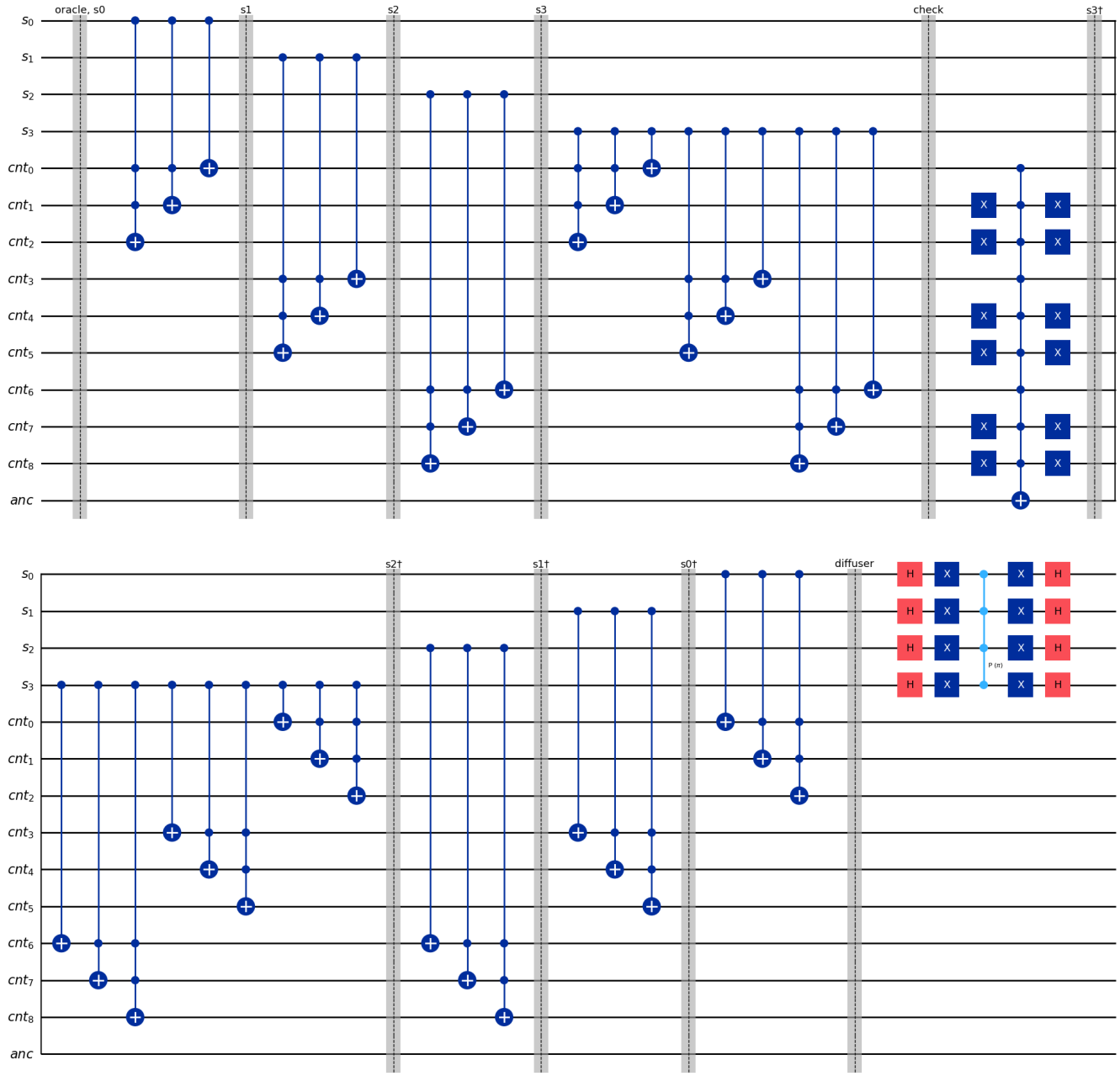

Figure 15. The quantum circuit of the Grover iterator (i.e., the oracle and the diffuser) to address the ECP in the third experiment.

Table 3. The detailed measurement results (combination and probability) of the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the third experiment.

Comb.	Prob.	Comb.	Prob.	Comb.	Prob.	Comb.	Prob.
0000	0.0026	0001	0.003	0010	0.0054	0011	0.004
0100	0.0032	0101	0.0042	0110	0.003	0111	0.4768
1000	0.4698	1001	0.003	1010	0.0058	1011	0.004
1100	0.0044	1101	0.0034	1110	0.0036	1111	0.0038

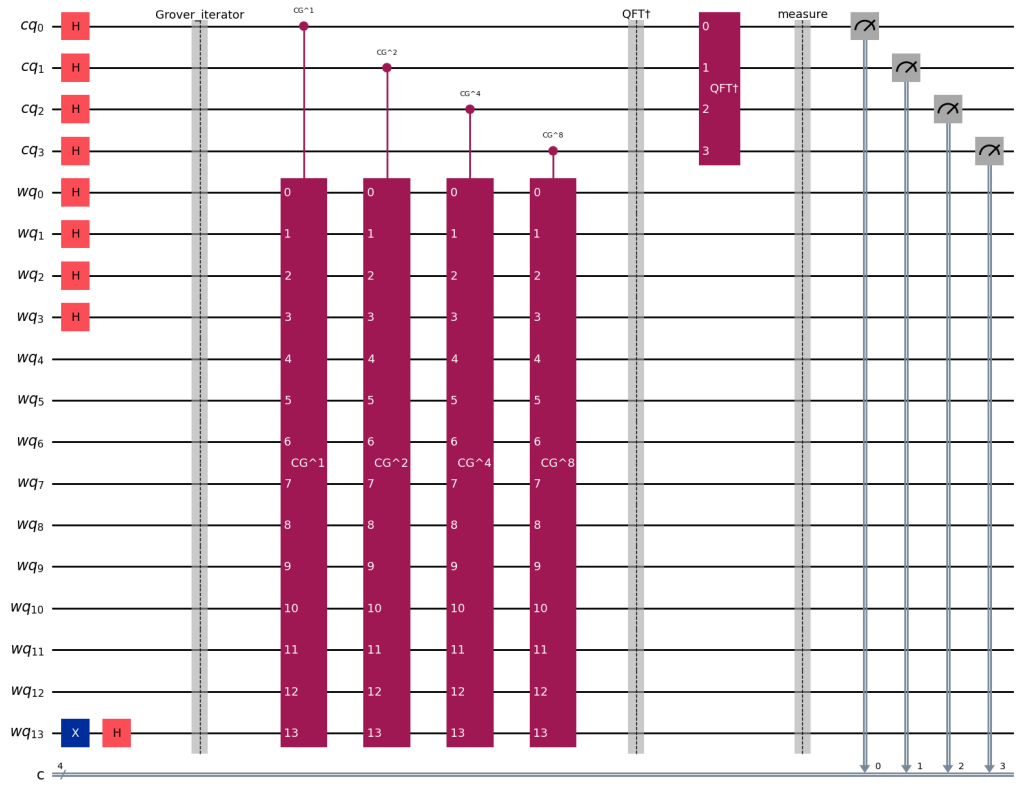

Figure 16. The quantum circuit of the quantum counting algorithm using the Grover iterator to derive the number of solutions of the ECP in the third experiment.

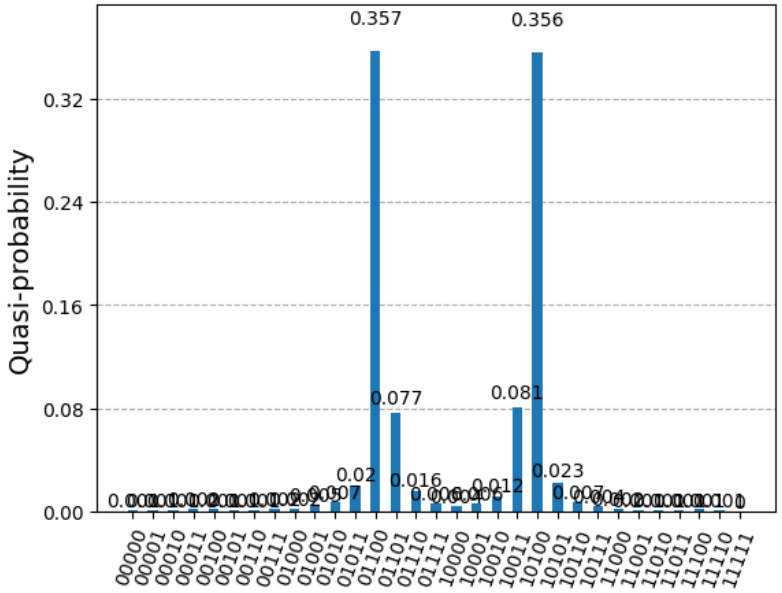

Figure 17. The histogram of the measurement results of the quantum counting algorithm to derive the number of solutions to the ECP in the third experiment.

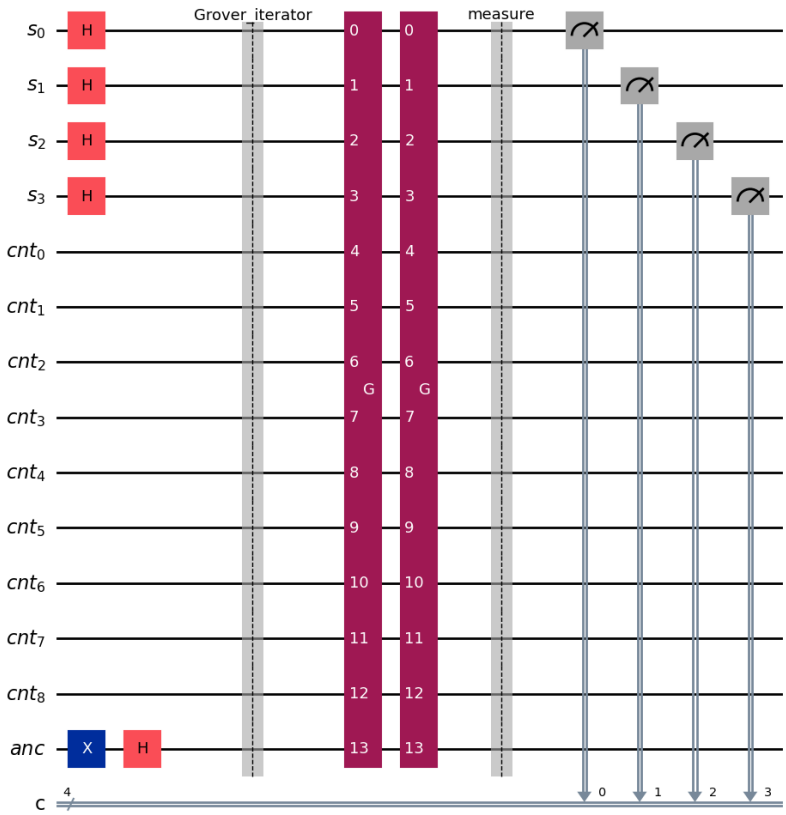

Figure 18. The quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the third experiment.

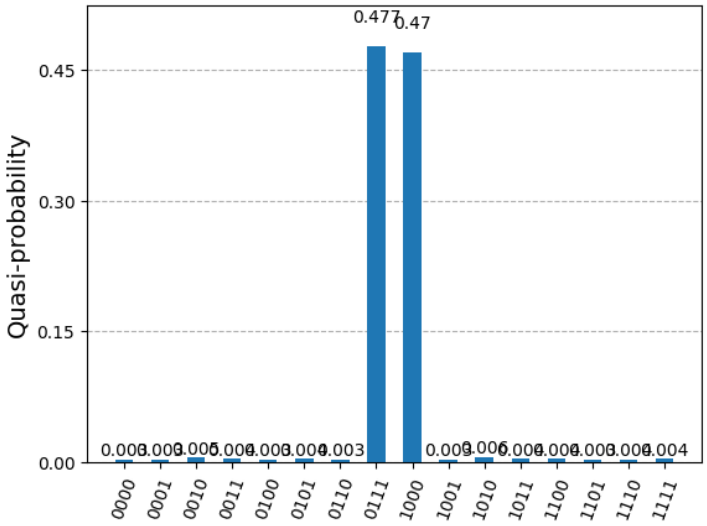

Figure 19. The histogram of the measurement results of the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the third experiment.

5.4. The Fourth Experiment

The fourth experiment is to construct and run a quantum circuit to address the ECP defined below. The universal set U is $\{u_0, u_1, u_2\}$ of three elements u_0, u_1 and u_2, and the collection C of sets is $\{s_0, s_1, s_2, s_3, s_4\}$, where $s_0, \ldots, s_4$ are subsets of U, $s_0 = \{u_0\}$, $s_1 = \{u_1\}, s_2 = \{u_2\}, s_3 = \{u_0, u_1\}$, and $s_4 = \{u_0, u_2\}$.

Figure 20 shows the quantum circuit of the Grover iterator that contains the oracle and the diffuser to address the given ECP. The oracle has five qubits $s_0, \ldots, s_4$ for sets $s_0, \ldots, s_4$ in C, and it has three counters for elements $u_0, \ldots, u_2$. Specifically, each counter uses three qubits, where $cnt_0, \ldots, cnt_2$ are with the counter for element u_0; $cnt_3, \ldots, cnt_5$ are with the counter for element u_1; and $cnt_6, \ldots, cnt_8$ are with the counter for element u_2. Furthermore, the oracle also has a an ancilla qubit that is denoted as acn and is used for phase kickback. Note that the Grover iterator uses a total of 15 qubits.

Figure 21 shows the quantum circuit of the quantum counting algorithm using the Grover iterator to derive the number of solutions to the ECP in the fourth experiment. The quantum circuit uses five counting qubits $cq_0, \ldots, cq_4$, each serving as the control qubit of the controlled Grover iterator (CG, or G for short). Specifically, qubit cq_0 is the control qubit of a G (or G^1 or G^{2^0}), cq_1 is the control qubit of two Gs (or G^2 or G^{2^1}), cq_2 is the control qubit of four Gs (or G^4 or G^{2^2}), and cq_4 is the control qubit of 16 Gs (or G^{16} or G^{2^4}). The counting qubits initially undergo H gates to be in superposition and in the Fourier basis and then undergo an inverse Fourier transform ($QFT^\dagger$) gate to return to the computational basis for measurement to derive the phase of the Grover iterator. Note that the quantum circuit has 15 working qubits, $wq_0, \ldots, wq_{14}$, which correspond to the 15 qubits used by the Grover iterator. The working qubits $wq_0, \ldots, wq_4$ correspond to the five qubits $s_0, \ldots, s_4$ in the Grover iterator; they undergo H gates and are in superposition initially. The last working qubit wq_{14} corresponds to the ancilla qubit of the Grover iterator; it undergoes an X gate and an H gate to be in state $|-\rangle$ initially.

Table 4 shows the detailed measurement results of the QCA to derive the number of solutions to the ECP in the fourth experiment. Moreover, Figure 22 shows the histogram of the measurement results of the QCA to derive the number of solutions to the ECP in the fourth experiment. Two outcomes, 01101 and 10011, have significant probabilities. They correspond to the values of 13 and 19, respectively. By dividing 2^5, where 5 is the number of counting qubits, 13 and 19 become $\frac{13}{2^5}$ and $\frac{19}{2^5}$, which in turn are mapped into phase angles in radians within the range $[-\pi, \pi]$. Specifically, they are mapped into the phase angle $2\pi \times \frac{13}{2^5} - \pi = \frac{26\pi}{32} - \pi = -\frac{6\pi}{32}$ and the phase angle $2\pi \times \frac{19}{2^5} - \pi = \frac{38\pi}{32} - \pi = \frac{6\pi}{32}$. With Equation (8), we can derive $M = N \times \sin^2\left(\frac{\theta}{2}\right) = 32 \times \sin^2\left(\frac{\pm 6\pi}{32}\right) = 2.69648$, which is rounded as 3. That is, using the quantum counting algorithm, we can determine that the ECP in the fourth experiment has $M = 3$ solutions.

Figure 23 shows the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the fourth experiment. Note that the Grover iterator is repeated $T = \lfloor \pi/4\sqrt{N/M} \rfloor = \lfloor \pi/4\sqrt{32/3} \rfloor = \lfloor 2.56509 \rfloor = 2$ times.

Table 5 shows the detailed measurement results (combination and probability) of the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the fourth experiment. Furthermore, Figure 24 shows the histogram of the measurement results of the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the fourth experiment. There are three outcomes of combinations (or patterns), namely 00111, 01100, and 10010, with significant measurement probabilities. Outcome 00111 corresponds to the sets $s_0 = \{u_0\}$, $s_1 = \{u_1\}$ and $s_2 = \{u_2\}$. Outcome 01100 corresponds to the sets $s_2 = \{u_2\}$ and $s_3 = \{u_0, u_1\}$. Outcome 10010 corresponds to the sets $s_1 = \{u_1\}$ and $s_4 = \{u_0, u_2\}$. The three outcomes correspond to the subcollections of C, each of which is a solution to the ECP in the fourth experiment.

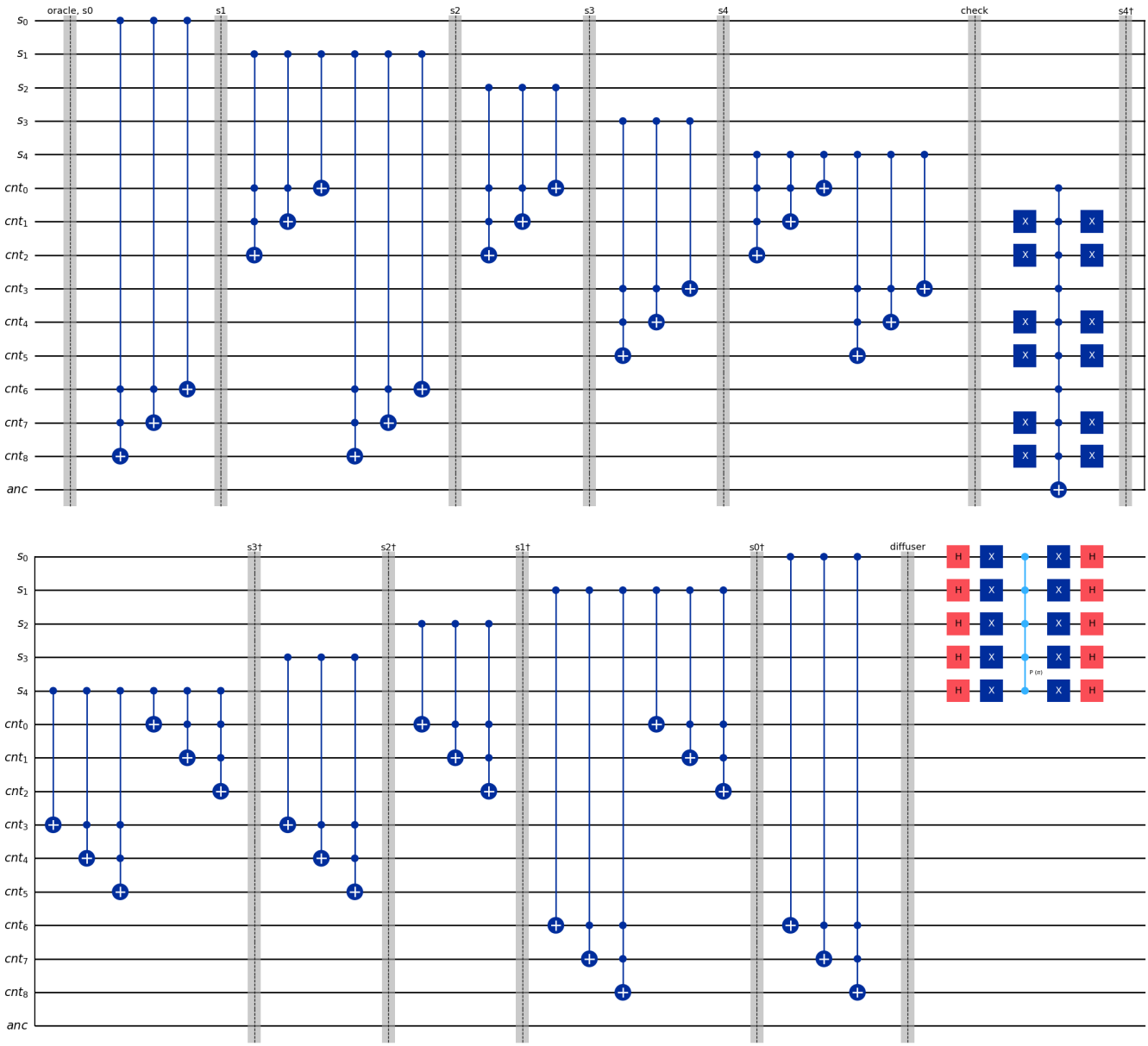

Figure 20. The quantum circuit of the Grover iterator (i.e., the oracle and the diffuser) to address the ECP in the fourth experiment.

Table 4. The detailed measurement results (combination and probability) of the quantum counting algorithm to derive the number of solutions to the ECP in the fourth experiment.

Comb.	Prob.	Comb.	Prob.	Comb.	Prob.	Comb.	Prob.
00000	0.0002	00001	0.0004	00010	0.0006	00011	0.0006
00100	0.0004	00101	0.0002	00101	0.0000	00111	0.0008
01000	0.0006	01001	0.0014	01010	0.0018	01011	0.0036
01100	0.0184	01101	0.458	01110	0.0096	01111	0.0038
10000	0.003	10001	0.0048	10010	0.0116	10011	0.4528
10100	0.019	10101	0.0028	10110	0.0012	10111	0.0012
11000	0.0006	11001	0.0008	11010	0.0002	11011	0.0004
11100	0.0002	11101	0.0002	11110	0.0002	11111	0.0006

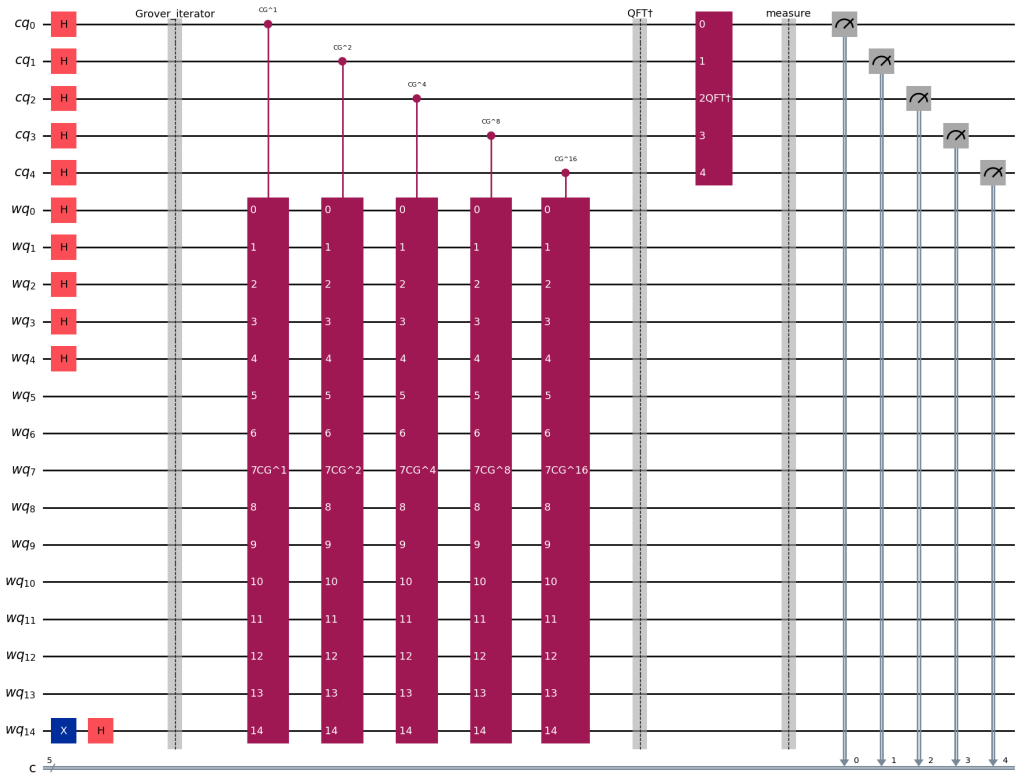

Figure 21. The quantum circuit of the quantum counting algorithm using the Grover iterator to derive the number of solutions to the ECP in the fourth experiment.

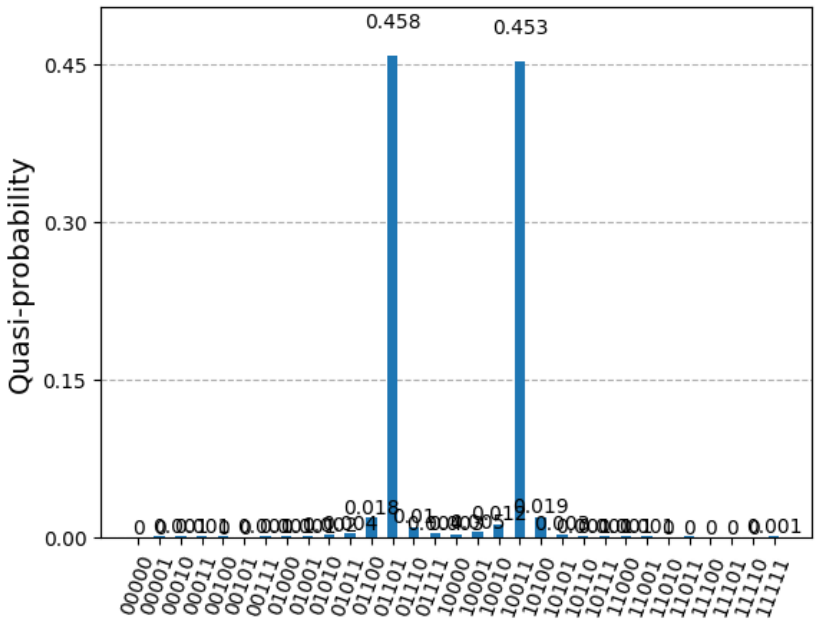

Figure 22. The histogram of the measurement results of the quantum counting algorithm to derive the number of solutions to the ECP in the fourth experiment.

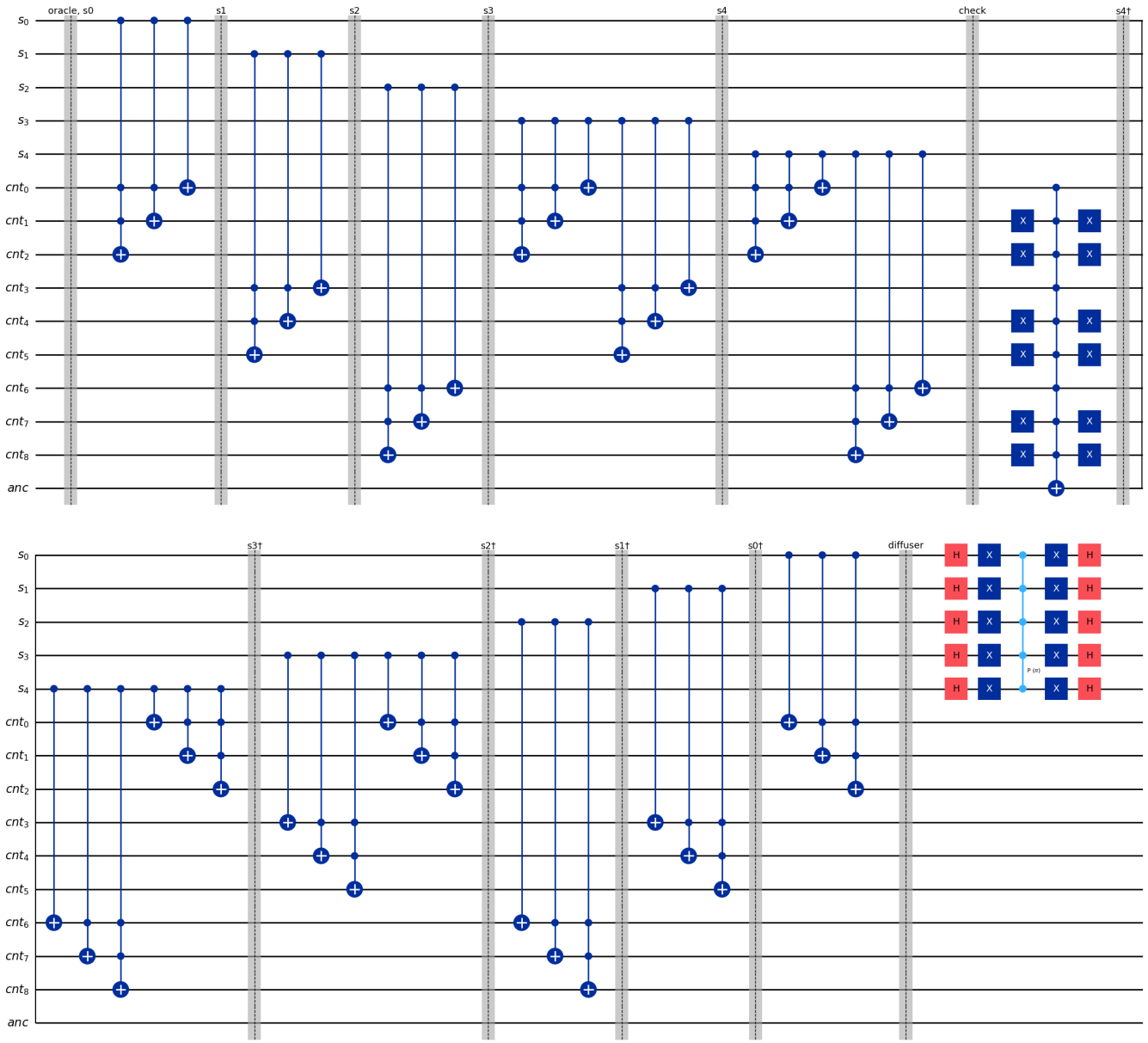

Figure 23. The quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the fourth experiment.

Table 5. The detailed measurement results (combination and probability) of the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the fourth experiment.

Comb.	Prob.	Comb.	Prob.	Comb.	Prob.	Comb.	Prob.
00000	0.0114	00001	0.0116	00010	0.015	00011	0.0112
00100	0.0098	00101	0.0124	00110	0.014	00111	0.2342
01000	0.0104	01001	0.0138	01010	0.0092	01011	0.0118
01100	0.2082	01101	0.0126	01110	0.0122	01111	0.0122
10000	0.0132	10001	0.0112	10010	0.2238	10011	0.0112
10100	0.0118	10101	0.0112	10110	0.0104	10111	0.0106
11000	0.0104	11001	0.0116	11010	0.011	11011	0.011
11100	0.0112	11101	0.011	11110	0.0114	11111	0.009

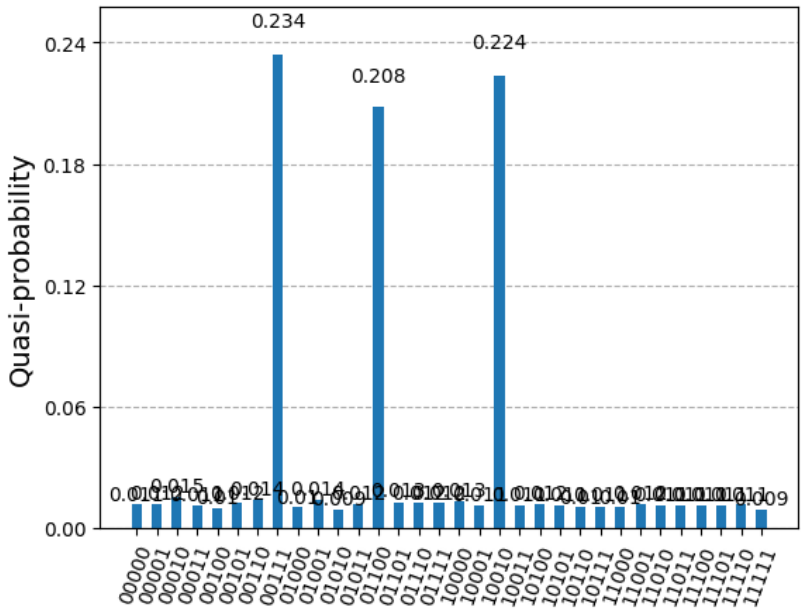

Figure 24. The histogram of the measurement results of the quantum circuit based on Grover's algorithm using the Grover iterator to address the ECP in the fourth experiment.

6. Conclusions

In this paper, we have designed and implemented quantum circuits based on Grover's algorithm to address the NP-hard ECP with a given collection of n sets, each of which is a subset of a given universal set containing m elements. The quantum circuits utilize a simplified quantum counter to construct the oracle of Grover's algorithm. Thanks to Grover's algorithm, these quantum circuits invoke the oracle along with the diffuser $\lfloor \pi/4\sqrt{N/M} \rfloor = O(\sqrt{N})$ times to address the ECP, where $N = 2^n$ is the total number of possible input instances and M is the number of solutions to the ECP. This achieves quadratic acceleration compared to the classical sequential algorithms invoking the oracle $O(N)$ times.

Compared to the existing quantum counter proposed by Heidari et al. in [26], the simplified quantum counter decreases the number of X gates by $2b - 2$ and reduces the depth of the quantum circuit by b for a quantum counter with b counting qubits. An oracle quantum circuit uses m quantum counters with $b = \lfloor \log n \rfloor + 1$ counting qubits twice. Thus, a whole quantum circuit to address the ECP saves $(4mb - 4m)\lfloor \pi/4\sqrt{N/M} \rfloor$ quantum gates, and its circuit depth is reduced by $(2mb)\lfloor \pi/4\sqrt{N/M} \rfloor$. This is particularly beneficial in the current NISQ era, where quantum computers have a limited number of qubits, low quantum gate fidelity, and short decoherence times.

We relied on the QCA to derive the number of solutions to the ECP that is not known in advance. However, the QCA needs to repeat the Grover iterator with t counting qubits a total of $2^0 + \ldots + 2^{t-1} = 2^t$ times. This corresponds to not only a large number of quantum gates but also a very large quantum circuit depth, which is a problem in the current NISQ era. We are now investigating how to address this problem. Furthermore, we plan to utilize the simplified quantum counter to implement quantum circuits based on Grover's algorithm to address more intricate problems, besides the Hamiltonian cycle, vertex cover, and dominating set problems solved by us in [20–23].

We have conducted four experiments using the IBM Qiskit packages to implement and execute our designed quantum circuits using the IBM Aer Simulator, successfully solving the ECP for correctness validation. We have also conducted experiments to execute the designed quantum circuits on real IBM 127-qubit quantum computers, e.g., *ibm_brisbane*

and *ibm_strasbourg*. However, despite our efforts to reduce the number of qubits used and shorten the quantum circuit depth, the results returned by the real quantum computer are not entirely consistent with those returned by the simulator. We do not observe significant probabilities of outcomes that lead to correct solutions, so these results are not included in the paper. This is likely because quantum simulators usually assume perfect gate fidelity with no noise, infinite qubit coherence times, and accurate qubit measurement without errors. However, we are still in the NISQ era, where the gate fidelity is not yet high enough, the decoherence time is not long enough, and qubit measurement still has errors. In the future, we plan to investigate error mitigation techniques and to re-execute these quantum circuits on available real quantum computers, hoping to achieve experimental results that are more consistent with those returned by the simulator. Another possibility is to resort to the latest fault-tolerant quantum computers with error correction capabilities, such as Google's Willow [38], which realizes a 101-qubit distance-7 surface code. Willow maintains below-threshold performance, meaning that its error rates stay below the critical threshold needed for effective quantum error correction. Implementing and executing our quantum circuit designs on large-scale, fault-tolerant quantum computers is also a future research goal that we hope to realize.

Author Contributions: Conceptualization, J.-R.J. and Y.-J.W.; methodology, J.-R.J. and Y.-J.W.; software, J.-R.J. and Y.-J.W.; validation, J.-R.J. and Y.-J.W.; formal analysis, J.-R.J. and Y.-J.W.; investigation, J.-R.J. and Y.-J.W.; resources, J.-R.J.; writing—original draft preparation, J.-R.J. and Y.-J.W.; writing—review and editing, J.-R.J.; visualization, J.-R.J. and Y.-J.W.; supervision, J.-R.J.; project administration, J.-R.J.; funding acquisition, J.-R.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Central University (NCU) and in part by the National Science and Technology Council (NSTC) under Grant 113-2622-E-008-019.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We express our gratitude to the National Taiwan University—IBM Quantum Computing Center (IBM Q Hub at NTU) for providing us with access to the IBM Q system.

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

References

1. Jiang, J.R.; Chu, C.W. Classifying and benchmarking quantum annealing algorithms based on quadratic unconstrained binary optimization for solving NP-hard problems. *IEEE Access* **2023**, *11*, 104165–104178. [CrossRef]
2. Jiang, J.R.; Shu, Y.C.; Lin, Q.Y. Benchmarks and Recommendations for Quantum, Digital, and GPU Annealers in Combinatorial Optimization. *IEEE Access* **2024**, *12*, 25014–125031. [CrossRef]
3. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef]
4. Bozzo-Rey, M.; Loredo, R. Introduction to the IBM Q experience and quantum computing. In Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering, Markham, ON, Canada, 29–31 October 2018; pp. 410–412.
5. AbuGhanem, M. A quantum state tomography study of Google's Sycamore gate on an IBM's quantum computer. *EPJ Quantum Technol.* **2023**, *11*, 10–1140. [CrossRef]
6. Computing, R. Scalable Quantum Systems Built from the Chip Up to Power Practical Applications. Available online: https://www.rigetti.com/what-we-build (accessed on 2 September 2024).
7. McGeoch, C.; Farré, P. *The D-Wave Advantage System: An Overview*; D-Wave Systems Inc.: Burnaby, BC, Canada, 2020.
8. Jiang, J.R.; Lin, Q.Y. Solving Maximum Independent Set Problem Using Analog Neutral Atom Quantum Computers. In Proceedings of the 2023 IEEE 5th Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 27–29 October 2023; pp. 802–807.
9. Abel, S.; Chancellor, N.; Spannowsky, M. Quantum computing for quantum tunneling. *Phys. Rev. D* **2021**, *103*, 016008. [CrossRef]

10. Deutsch, D.; Jozsa, R. Rapid solution of problems by quantum computation. *Proc. R. Soc. Lond. Ser. A Math. Phys. Sci.* **1992**, *439*, 553–558.
11. Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
12. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia PA, USA, 22–24 May 1996; pp. 212–219.
13. Bennett, C.H.; Bernstein, E.; Brassard, G.; Vazirani, U. Strengths and weaknesses of quantum computing. *SIAM J. Comput.* **1997**, *26*, 1510–1523. [CrossRef]
14. Boyer, M.; Brassard, G.; Høyer, P.; Tapp, A. Tight bounds on quantum searching. *Fortschritte Der Phys. Prog. Phys.* **1998**, *46*, 493–505. [CrossRef]
15. Saha, A.; Saha, D.; Chakrabarti, A. Circuit design for k-coloring problem and its implementation on near-term quantum devices. In Proceedings of the 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), Chennai, India, 14–16 December 2020; pp. 17–22.
16. Saha, A.; Saha, D.; Chakrabarti, A. Circuit design for k-coloring problem and its implementation in any dimensional quantum system. *SN Comput. Sci.* **2021**, *2*, 427. [CrossRef]
17. Haverly, A.; López, S. Implementation of Grover's algorithm to solve the maximum clique problem. In Proceedings of the 2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Tampa, FL, USA, 7–9 July 2021; pp. 441–446.
18. Mukherjee, S. A Grover search-based algorithm for the list coloring problem. *IEEE Trans. Quantum Eng.* **2022**, *3*, 3101008. [CrossRef]
19. Roch, C.; Castillo, S.L.; Linnhoff-Popien, C. A Grover based quantum algorithm for finding pure Nash equilibria in graphical games. In Proceedings of the 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C), Honolulu, HI, USA, 12–15 March 2022; pp. 147–151.
20. Jiang, J.R. Quantum circuit based on Grover algorithm to solve Hamiltonian cycle problem. In Proceedings of the 2022 IEEE 4th Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 28–30 October 2022; pp. 364–367.
21. Jiang, J.R.; Kao, T.H. Solving Hamiltonian Cycle Problem with Grover's Algorithm Using Novel Quantum Circuit Designs. In Proceedings of the 2023 IEEE 5th Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 27–29 October 2023; pp. 796–801.
22. Jiang, J.R.; Yan, W.H. Novel Quantum Circuit Designs for the Oracle of Grover's Algorithm to Solve the Vertex Cover Problem. In Proceedings of the 2023 IEEE 5th Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 27–29 October 2023; pp. 652–657.
23. Jiang, J.R.; Lin, Q.Y. Utilizing Novel Quantum Counters for Grover's Algorithm to Solve the Dominating Set Problem. *arXiv* **2023**, arXiv:2312.09388.
24. Jiang, J.R.; Wang, Y.J. Quantum circuit based on Grover's algorithm to solve exact cover problem. In Proceedings of the 2023 VTS Asia Pacific Wireless Communications Symposium (APWCS), Tainan City, Taiwan, 23–25 August 2023; pp. 1–5.
25. Karp, R.M. Reducibility among Combinatorial Problems. In Proceedings of the Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, New York, NY, USA, 20–22 March 1972; pp. 85–103.
26. Heidari, S.; Farzadnia, E. A novel quantum LSB-based steganography method using the Gray code for colored quantum images. *Quantum Inf. Process.* **2017**, *16*, 242. [CrossRef]
27. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
28. Quantum, I. Qiskit: An Open-Source Quantum Computing Framework. 2024. Available online: https://www.ibm.com/quantum/qiskit (accessed on 1 October 2024).
29. Brassard, G.; Høyer, P.; Tapp, A. Quantum counting. In Proceedings of the Automata, Languages and Programming: 25th International Colloquium, ICALP'98, Aalborg, Denmark, 13–17 July 1998; Proceedings 25; Springer: Berlin/Heidelberg, Germany, 1998; pp. 820–831.
30. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2010.
31. Bojić, A. Quantum algorithm for finding a maximum clique in an undirected graph. *J. Inf. Organ. Sci.* **2012**, *36*, 91–98.
32. IBM. *IBM Builds Its Most Powerful Universal Quantum Computing Processors*; IBM: Hampshire, UK, 2017.
33. The MathWorks. *MATLAB, R2012b*; The MathWorks: Natick, MA, USA, 2012.
34. Gilliam, A.; Woerner, S.; Gonciulea, C. Grover adaptive search for constrained polynomial binary optimization. *Quantum* **2021**, *5*, 428. [CrossRef]
35. Gottlob, G.; Greco, G.; Scarcello, F. Pure Nash equilibria: Hard and easy games. In Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge, Bloomington, IN, USA, 20–22 June 2003; pp. 215–230.
36. Fernandes, D.; Dutra, I. Using Grover's search quantum algorithm to solve Boolean satisfiability problems: Part I. *XRDS Crossroads ACM Mag. Stud.* **2019**, *26*, 64–66. [CrossRef]

37. Campbell, E.; Khurana, A.; Montanaro, A. Applying quantum algorithms to constraint satisfaction problems. *Quantum* **2019**, *3*, 167. [CrossRef]
38. Acharya, R.; Abanin, D.A.; Aghababaie-Beni, L.; Aleiner, I.; Andersen, T.I.; Ansmann, M.; Arute, F.; Arya, K.; Asfaw, A.; Astrakhantsev, N.; et al. Quantum error correction below the surface code threshold. *Nature* **2024**, *618*, 84–90. [CrossRef]