FRONTIERS OF PHYSICS

RESEARCH ARTICLE
Volume 18 / Issue 5 / 51305 / 2023

Distributed exact Grover's algorithm

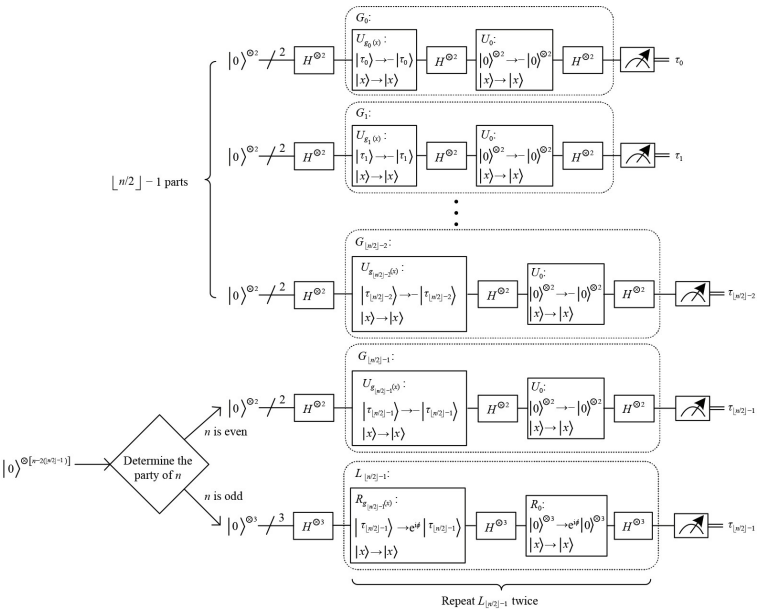
Xu Zhou[1,2,3,4,†], Daowen Qiu[1,2,4,‡], Le Luo[3,4,§]

1 Institute of Quantum Computing and Computer Theory, School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China
2 The Guangdong Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou 510006, China
3 School of Physics and Astronomy, Sun Yat-sen University, Zhuhai 519082, China
4 QUDOOR Co, Ltd., Beijing 100089, China
Corresponding authors. E-mail: †zhoux229@mail2.sysu.edu.cn, ‡issqdw@mail.sysu.edu.cn, §luole5@mail.sysu.edu.cn
Received March 16, 2023; accepted June 7, 2023

ABSTRACT

Distributed quantum computation has gained extensive attention. In this paper, we consider a search problem that includes only one target item in the unordered database. After that, we propose a distributed exact Grover's algorithm (DEGA), which decomposes the original search problem into $\lfloor n/2 \rfloor$ parts. Specifically, (i) our algorithm is as exact as the modified version of Grover's algorithm by Long, which means the theoretical probability of finding the objective state is 100%; (ii) the actual depth of our circuit is $8(n \bmod 2)+9$, which is less than the circuit depths of the original and modified Grover's algorithms, $1+8\left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor$ and $9+8\left\lfloor \frac{\pi}{4}\sqrt{2^n}-\frac{1}{2} \right\rfloor$, respectively. It only depends on the parity of n, and it is not deepened as n increases; (iii) we provide particular situations of the DEGA on MindQuantum (a quantum software) to demonstrate the practicality and validity of our method. Since our circuit is shallower, it will be more resistant to the depolarization channel noise.

Keywords distributed quantum computation, search problem, distributed exact Grover's algorithm (DEGA), MindQuantum, the depolarization channel noise



1 Introduction

Quantum computation is a novel computing model based on quantum mechanics. It has gained considerable attention over the past few decades since Benioff [1, 2] proposed the concept of the quantum Turing machine in 1980. Subsequently, rapid progress has been made in quantum computation, so that numerous impressive quantum algorithms have been presented, such as Deutsch's algorithm [3], Deutsch–Jozsa (DJ) algorithm [4], Bernstein-Vazirani (BV) algorithm [5], Simon's algorithm [6], Shor's algorithm [7], Grover's algorithm [8], HHL algorithm [9] and so on. For some specific problems, due to the unique properties of quantum superposition

Higher Education Press

arXiv: 2303.10670
https://doi.org/10.1007/s11467-023-1327-x

and quantum entanglement, quantum algorithms have shown incomparable advantages over classical algorithms.

Grover's algorithm was proposed by Grover [8] in 1996, which is a quantum search algorithm that solves the search problem in an unordered database with 2^n elements. Specifically, let Boolean function $f:\{0,1\}^n \to \{0,1\}$. For the m targets search problem, it aims to find out one target string $x_i \in \{0,1\}^n$ satisfying $f(x_i)=1$, where $i \in \{0,1,\cdots,m-1\}$. Suppose we have a black box (Oracle) that can recognize the inputs, which means it will return $f(x)=1$ when x is a target string, and output $f(x)=0$ otherwise. The classical algorithms have to query Oracle $\mathcal{O}(2^n/m)$ times, while Grover's algorithm needs to query $\mathcal{O}\left(\sqrt{2^n/m}\right)$ times to figure out the target string with great probability, which has square acceleration compared with the classical algorithms.

Grover's algorithm is widely regarded as one of the most prominent quantum algorithms. Furthermore, it is widely used in the minimum search problem [10], string matching problem [11], quantum dynamic programming [12] and computational geometry problem [13], etc. Subsequently, a generalization of Grover's algorithm known as the quantum amplitude amplification and estimation algorithms [14] were presented.

However, Grover's algorithm is unable to search for the target state accurately, except in the case of 1/4 of the data in the database meet the search conditions, which has been strictly proved [15]. In 2001, Long [16] improved Grover's algorithm and proposed the modified version, which will acquire the goal state with a probability of 100%. Its main idea is to extend Grover operator G to operator L. The adjustment is carried out by three steps: (i) substitute phase rotation for phase inversion; (ii) let the phase rotation angles of two reflections in L equal, which is called phase-matching condition; (iii) iterate $J+1$ times for operator L, where $J=\lfloor(\pi/2-\theta)/(2\theta)\rfloor$ and $\theta=\arcsin\left(\sqrt{m/2^n}\right)$. Actually, J can be any integer equal to or greater than $\lfloor(\pi/2-\theta)/(2\theta)\rfloor$.

In 2017, Preskill [17] declared that quantum computation is entering the noisy intermediate-scale quantum (NISQ) era. To deploy quantum computers, various physical systems are taken into consideration, such as ions [18], photons [19], superconduction [20], and other quantum systems [21, 22]. Currently, small-qubit quantum computers are much easier to implement than large-scale universal quantum computers. Hence, researchers are considering how several small-scale devices might collaborate to accomplish a task on a large scale.

Distributed quantum computation, combining distributed computing and quantum computation, has gained extensive attention. In order to process a huge issue efficiently, it attempts to break it up into many subproblems that are distributed across several quantum computers. In this way, each component needs fewer qubits and has a shallower quantum circuit.

This research field has seen a significant amount of theoretical and experimental work [23–26]. Avron *et al.* [27] proposed a distributed method for constructing Oracle and then illustrated the advantage in the case of a Grover research. In their Grover's algorithm experiment, however, they only acquired $n-1$ qubits of the target state rather than the target state itself directly. The total number of qubits used in their scheme is $2(n-1)$.

Qiu *et al.* [28] proposed two distributed Grover's algorithms (parallel and serial) and these algorithms require fewer qubits and have shallower depth of circuits. Notably, the distributed algorithms by Qiu *et al.* [28] can solve the search problem with multiple targets. They divided a Boolean function f into 2^k subfunctions, where 2^k represents the number of computing nodes. By employing the quantum counting algorithm, the number of solutions in each subfunction can be determined, and then the solution for each subfunction can be obtained. Particularly, Qiu *et al.* [28] proposed an efficient algorithm of constructing quantum circuits for realizing the Oracle corresponding to any Boolean function with conjunctive normal form (CNF). The total number of qubits used in their parallel scheme is $2^k(n-k)$.

Tan *et al.* [29] presented an interesting and novel distributed Simon's algorithm, which improves essentially the number of qubits for inputs and the depth of circuits compared to the famous Simon's algorithm. Recently, Xiao *et al.* [30] proposed a distributed Shor's algorithm, which can separately estimate patrial bits of s/r for some $s \in \{0,1,\cdots,r-1\}$ by two quantum computers during solving order-finding. Later, they extended this method to multiple nodes [31]. Zhou *et al.* [32] designed a distributed BV algorithm (DBVA) with $2 \le t \le n$ nodes. Comparing with the original BV algorithms, their scheme is easier to verify in experiments. Li *et al.* [33] proposed a multi-node distributed exact quantum algorithm for solving the DJ problem with super-exponential speedup compared to distributed classical deterministic algorithms.

In this paper, we consider a search problem that includes only one target item in the unordered database (we still do not know how to solve it for the case of multiple targets). After that, we propose a distributed exact Grover's algorithm (DEGA), which decomposes the original search problem into $\lfloor n/2 \rfloor$ parts. Specifically, (i) our algorithm is as exact as the modified version of Grover's algorithm by Long, which means the theoretical probability of finding the objective state is 100%; (ii) the actual depth of our circuit is $8(n \bmod 2)+9$, which is less than the circuit depths of the original and modified Grover's algorithms, $1+8\left\lfloor\frac{\pi}{4}\sqrt{2^n}\right\rfloor$ and $9+8\left\lfloor\frac{\pi}{4}\sqrt{2^n}-\frac{1}{2}\right\rfloor$, respectively. It only depends on the parity of n, and it is not deepened as n increases.

MindQuantum (a quantum software) [34] is a general software library supporting the development of applications for quantum computation. Eventually, we provide

particular situations of the DEGA on MindQuantum. These experiments will further demonstrate the correctness of the DEGA and explicate the specific steps of implement n-qubit DEGA, where $n \in \{2,3,4,5\}$. After that, we will simulate a 5-qubit DEGA running in the depolarization channel and compare with the original and modified Grover's algorithms. Since our circuit is shallower, it will be more resistant to the depolarization channel noise.

The rest of the paper is organized as follows. We will briefly review the original and modified Grover's algorithms in Section 2 and primarily describe the DEGA in Section 3. Next, the analysis of our the algorithm will be presented in Section 4. After that, we will provide particular situations of the DEGA on MindQuantum in Section 5. Finally, we will give a brief conclusion in Section 6.

2 Preliminaries

In this section, we briefly review the original [8] and modified [16] Grover's algorithms.

2.1 Grover's algorithm [8]

Suppose that the search task is carried out in an unordered set (or database) with 2^n elements. We establish a one-to-one correspondence between the elements in the database and the indexes (integers from 0 to 2^n-1). We focus on searching the indexes of these elements.

Let Boolean function $f:\{0,1\}^n \to \{0,1\}$. Define a function for the search problem as follows:

$$f(x) = \begin{cases} 1, & x = \tau, \\ 0, & x \neq \tau, \end{cases} \tag{1}$$

where $x \in \{0,1\}^n$ and τ is the target index. In a quantum system, indexes are represented by quantum states $|0\rangle, |1\rangle, \cdots, |2^n-1\rangle$ (or $|00\cdots0\rangle, |00\cdots1\rangle, \cdots, |11\cdots1\rangle$).

Without loss of generality, we consider a search problem that includes only one target item in the unordered database throughout this paper. Suppose we have a black box (Oracle $U_{f(x)}:|x\rangle \to (-1)^{f(x)}|x\rangle$, where $x \in \{0,1\}^n$) that can recognize the inputs. The purpose of Grover's algorithm is to find out $|\tau\rangle$ with high probability.

Here, we briefly review Grover's algorithm in Algorithm 1. The whole quantum circuit is shown in Fig. 1.

Algorithm 1 Grover's algorithm (includes only one target).

Input: (1) The number of qubits n; (2) A function $f(x)$, where $f(x)=0$ for all $x \in \{0,1\}^n$ except τ, for which $f(\tau)=1$.
Initialization: An Oracle $U_{f(x)}:|x\rangle \to (-1)^{f(x)}|x\rangle$, where $x \in \{0,1\}^n$.
Output: The string $\tau \in \{0,1\}^n$ such that $f(\tau)=1$ with great probability.
Procedure:
Step 1. Initialize n qubits $|0\rangle^{\otimes n}$.
Step 2. Apply n Hadamard gates $H^{\otimes n}$.
Step 3. Grover operator $G = -H^{\otimes n}U_0H^{\otimes n}U_{f(x)}$ is executed with $\lfloor \frac{\pi}{4}\sqrt{2^n} \rfloor$ times, where $U_0 = I^{\otimes n} - 2(|0\rangle\langle 0|)^{\otimes n}$ and $U_{f(x)} = I^{\otimes n} - 2|\tau\rangle\langle\tau|$.
Step 4. Measure each qubit in the basis $\{|0\rangle, |1\rangle\}$.

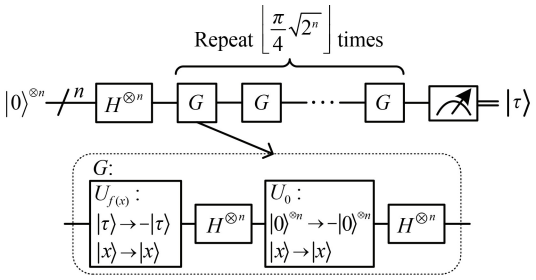


Fig. 1 Quantum circuit of Grover's algorithm.

Next, we analyse Grover's algorithm.
Denote

$$|\tilde{x}\rangle \triangleq \sqrt{\frac{1}{2^n-1}} \sum_{x \neq \tau} |x\rangle, \tag{2}$$

$$|h\rangle \triangleq H^{\otimes n}(|0\rangle^{\otimes n}) = \sqrt{\frac{1}{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle. \tag{3}$$

Then we can write

$$\begin{aligned} |h\rangle =& \sqrt{\frac{2^n-1}{2^n}}|\tilde{x}\rangle + \sqrt{\frac{1}{2^n}}|\tau\rangle \\ \triangleq& \cos\theta|\tilde{x}\rangle + \sin\theta|\tau\rangle, \end{aligned} \tag{4}$$

where

$$\theta = \arcsin\left(\sqrt{\frac{1}{2^n}}\right) \in \left(0, \frac{\pi}{2}\right). \tag{5}$$

Since $U_0 = I^{\otimes n} - 2(|0\rangle\langle 0|)^{\otimes n}$ and $U_{f(x)} = I^{\otimes n} - 2|\tau\rangle\langle\tau|$, we have

$$\begin{aligned} G =& -H^{\otimes n}U_0H^{\otimes n}U_{f(x)} \\ =& -H^{\otimes n}\left(I^{\otimes n} - 2(|0\rangle\langle 0|)^{\otimes n}\right)H^{\otimes n}\left(I^{\otimes n} - 2|\tau\rangle\langle\tau|\right) \\ =& -\left(I^{\otimes n} - 2|h\rangle\langle h|\right)\left(I^{\otimes n} - 2|\tau\rangle\langle\tau|\right). \end{aligned} \tag{6}$$

Therefore, the effect of one iteration of Grover operator G is two successive reflections in the space spanned by $\{|\tilde{x}\rangle, |\tau\rangle\}$, around $|\tilde{x}\rangle$ and $|h\rangle$, respectively. In other words, one iteration of G causes a rotation by an angle 2θ from the initial state $|h\rangle$ to the target state $|\tau\rangle$ (see Fig. 2).

After k iterations of G, the state will be

$$G^k|h\rangle = \cos((2k+1)\theta)|\tilde{x}\rangle + \sin((2k+1)\theta)|\tau\rangle. \tag{7}$$

The goal is to make

$$\sin((2k+1)\theta) \approx 1, \tag{8}$$

then

$$(2k+1)\theta \approx \frac{\pi}{2}, \tag{9}$$

will suffice, so we should choose

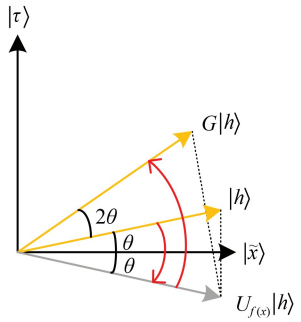


Fig. 2 Visualization of Grover's algorithm. In the two-dimensional plane space spanned by $\{|\tilde{x}\rangle, |\tau\rangle\}$, one iteration of G causes a rotation by an angle 2θ from the initial state $|h\rangle$ to the target state $|\tau\rangle$.

$$k \approx \frac{\pi/2-\theta}{2\theta} = \frac{\pi}{4\theta} - \frac{1}{2}. \tag{10}$$

Note that k must be a positive integer.

Since we have assumed there is only one target item, then

$$\theta = \arcsin\left(\sqrt{\frac{1}{2^n}}\right) \approx \sqrt{\frac{1}{2^n}}, \tag{11}$$

so

$$k = \left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor \tag{12}$$

is a suitable choice for Grover's algorithm. The success probability is

$$\sin^2\left(\left(2\left\lfloor\frac{\pi}{4}\sqrt{2^n}\right\rfloor+1\right)\arcsin\left(\sqrt{\frac{1}{2^n}}\right)\right). \tag{13}$$

2.2 The modified Grover's algorithm by Long [16]

In 2001, Long improved Grover's algorithm and proposed a modified version of Grover's algorithm, which will acquire the goal state with a probability of 100%. Its main idea is to extend Grover operator G to operator L. The adjustment is carried out by three steps: (1) substitute phase rotation for phase inversion; (2) let the phase rotation angles of two reflections in L equal, which is called phase-matching condition; (3) iterate $J+1$ times for operator L, where $J = \lfloor(\pi/2-\theta)/(2\theta)\rfloor$ and $\theta = \arcsin\left(\sqrt{1/2^n}\right)$. Actually, J can be any integer equal to or greater than $\lfloor(\pi/2-\theta)/(2\theta)\rfloor$.

Here, we briefly review the modified Grover's algorithm by Long in Algorithm 2. The whole quantum circuit is shown in Fig. 3.

Obviously, the modified Grover's algorithm is a Grover's algorithm when $\phi = \pi$.

Next, we analyse the modified Grover's algorithm.

In Grover's algorithm, iterate

Algorithm 2 The modified Grover's algorithm (includes only one target).

Input: (1) The number of qubits n; (2) A function $f(x)$, where $f(x) = 0$ for all $x \in \{0,1\}^n$ except τ, for which $f(\tau) = 1$.

Initialization: An Oracle $R_{f(x)} : |x\rangle \to e^{i\phi\cdot f(x)}|x\rangle$, where $x \in \{0,1\}^n$, $\phi = 2\arcsin\left(\sin\left(\frac{\pi}{4J+6}\right)/\sin\theta\right)$, $J = \lfloor(\pi/2-\theta)/(2\theta)\rfloor$, and $\theta = \arcsin\left(\sqrt{1/2^n}\right)$.

Output: The string $\tau \in \{0,1\}^n$ such that $f(\tau) = 1$ with a probability of 100%.

Procedure:

Step 1. Initialize n qubits $|0\rangle^{\otimes n}$.

Step 2. Apply n Hadamard gates $H^{\otimes n}$.

Step 3. Operator $L = -H^{\otimes n}R_0H^{\otimes n}R_{f(x)}$ is executed with $J+1$ times, where $R_0 = I^{\otimes n} + (e^{i\phi}-1)(|0\rangle\langle 0|)^{\otimes n}$ and $R_{f(x)} = I^{\otimes n} + (e^{i\phi}-1)|\tau\rangle\langle\tau|$.

Step 4. Measure each qubit in the basis $\{|0\rangle, |1\rangle\}$.

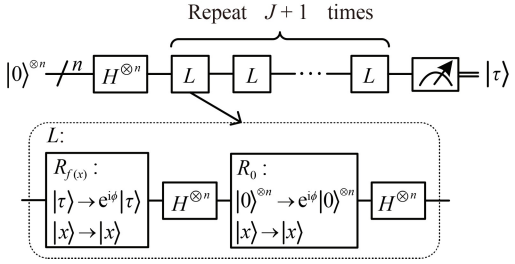


Fig. 3 Quantum circuit of the modified Grover's algorithm.

$$j_{op} = \lfloor(\pi/2-\theta)/(2\theta)\rfloor \tag{14}$$

or $j_{op}+1$ times of the Grover operator G, so that $(2j_{op}+1)\theta$ and $[2(j_{op}+1)+1]\theta$ are close to $\pi/2$, where $\theta = \arcsin\left(\sqrt{1/2^n}\right)$. However, Grover's algorithm is unable to search for the target state accurately, except in the case of 1/4 of the data in the database meet the search conditions.

After substituting phase rotation $R_0 = I^{\otimes n} + (\mathrm{e}^{\mathrm{i}\phi}-1)(|0\rangle\langle 0|)^{\otimes n}$ and $R_{f(x)} = I^{\otimes n} + (\mathrm{e}^{\mathrm{i}\phi}-1)|\tau\rangle\langle\tau|$ for phase inversion $U_0 = I^{\otimes n} - 2(|0\rangle\langle 0|)^{\otimes n}$ and $U_{f(x)} = I^{\otimes n} - 2|\tau\rangle\langle\tau|$, we have

$$\begin{aligned} L =& -H^{\otimes n}R_0H^{\otimes n}R_{f(x)} \\ =& -H^{\otimes n}\left[I^{\otimes n} + (\mathrm{e}^{\mathrm{i}\phi}-1)(|0\rangle\langle 0|)^{\otimes n}\right] \\ & \cdot H^{\otimes n}\left[I^{\otimes n} + (\mathrm{e}^{\mathrm{i}\phi}-1)|\tau\rangle\langle\tau|\right] \\ =& -\left[I^{\otimes n} + (\mathrm{e}^{\mathrm{i}\phi}-1)|h\rangle\langle h|\right]\left[I^{\otimes n} + (\mathrm{e}^{\mathrm{i}\phi}-1)|\tau\rangle\langle\tau|\right] \\ =& \begin{pmatrix} -\mathrm{e}^{\mathrm{i}\phi}(1+(\mathrm{e}^{\mathrm{i}\phi}-1)\sin^2\theta) & -\mathrm{e}^{\mathrm{i}\phi}(1-\mathrm{e}^{-\mathrm{i}\phi})\sin\theta\cos\theta \\ -\mathrm{e}^{\mathrm{i}\phi}(\mathrm{e}^{\mathrm{i}\phi}-1)\sin\theta\cos\theta & -\mathrm{e}^{\mathrm{i}\phi}(1-(1-\mathrm{e}^{-\mathrm{i}\phi})\sin^2\theta) \end{pmatrix} \end{aligned} \tag{15}$$

$$= -\mathrm{e}^{\mathrm{i}\phi}\left[\cos\left(\frac{\alpha}{2}\right)I + \mathrm{i}\sin\left(\frac{\alpha}{2}\right)(n_xX + n_yY + n_zZ)\right], \tag{16}$$

where

$$\alpha = 4\beta, \quad \sin\beta = \sin\left(\frac{\phi}{2}\right)\sin\theta, \quad \theta = \arcsin\left(\sqrt{1/2^n}\right), \tag{17}$$

and

$$n_x = \frac{\cos\theta}{\cos\beta}\cos\left(\frac{\phi}{2}\right),\quad n_y = \frac{\cos\theta}{\cos\beta}\sin\left(\frac{\phi}{2}\right),$$
$$n_z = \frac{\cos\theta}{\cos\beta}\cos\left(\frac{\phi}{2}\right)\tan\theta, \tag{18}$$

and I, X, Y, Z represent the Pauli gates,

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$
$$Y = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{19}$$

The equivalence of Eq. (15) and Eq. (16) will be detailed in Appendix A. It can be seen that the phase rotation angles of two reflections in L are equal, which is called phase-matching condition. Furthermore, L can be regarded as the three-dimensional rotation about the axis $\boldsymbol{l}$ in the Bloch sphere spanned by $\{|\tilde{x}\rangle, |\tau\rangle\}$ and the rotation angel is α, where

$$\boldsymbol{l} = [n_x, n_y, n_z]^{\mathrm{T}} = \frac{\cos\theta}{\cos\beta}\left[\cos\left(\frac{\phi}{2}\right), \sin\left(\frac{\phi}{2}\right), \cos\left(\frac{\phi}{2}\right)\tan\theta\right]^{\mathrm{T}}. \tag{20}$$

The initial state $|h\rangle$ and the target state $|\tau\rangle$ in the three-dimensional Bloch sphere can represent by the following vectors

$$\boldsymbol{r}_i = [\sin(2\theta), 0, -\cos(2\theta)]^{\mathrm{T}}, \tag{21}$$

$$\boldsymbol{r}_f = [0, 0, 1]^{\mathrm{T}}. \tag{22}$$

Denote $\boldsymbol{r}_0$ as the projections of $\boldsymbol{r}_i$ and $\boldsymbol{r}_f$ on axis $\boldsymbol{l}$,

$$\boldsymbol{r}_0 = c\left[\cos^2\left(\frac{\phi}{2}\right)\tan\theta, \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\phi}{2}\right)\tan\theta, \cos^2\left(\frac{\phi}{2}\right)\tan^2\theta\right]^{\mathrm{T}}, \tag{23}$$

where

$$c = \frac{1}{1 + \left(\cos^2\left(\frac{\phi}{2}\right)/\tan^2\theta\right)}. \tag{24}$$

As shown in Fig. 4, the desired rotation angle ω between $\boldsymbol{r}_i - \boldsymbol{r}_0$ and $\boldsymbol{r}_f - \boldsymbol{r}_0$ satisfies

$$\cos\omega = \frac{(\boldsymbol{r}_i - \boldsymbol{r}_0)\cdot(\boldsymbol{r}_f - \boldsymbol{r}_0)}{|\boldsymbol{r}_i - \boldsymbol{r}_0||\boldsymbol{r}_f - \boldsymbol{r}_0|} = \cos\left(2\arccos\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right)\right), \tag{25}$$

then we will have

$$\omega = 2\arccos\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right) = 2\left[\frac{\pi}{2} - \arcsin\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right)\right]. \tag{26}$$

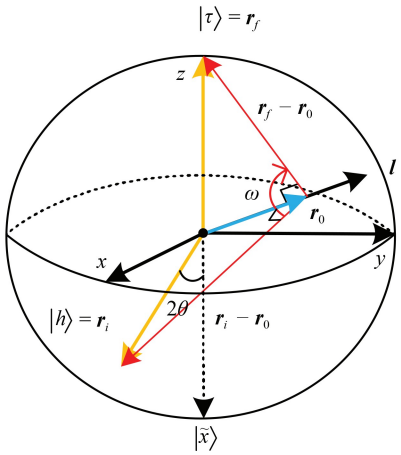


Fig. 4 Visualization of the modified Grover's algorithm. In the three-dimensional Bloch sphere spanned by $\{|\tilde{x}\rangle, |\tau\rangle\}$, the initial state $\boldsymbol{r}_i$ rotate ω around axis $\boldsymbol{l}$, and then $\boldsymbol{r}_f$ is obtained, where ω is the desired rotation angle.

Besides, in order to obtain the target state with certainty, angle ω ought to satisfy

$$\omega = (J+1)\alpha = 4(J+1)\arcsin\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right) \tag{27}$$

after $J+1$ times iteration of operator L. Combining Eq. (25) and Eq. (27), we can get

$$\sin\left(\frac{\pi}{4J+6}\right) = \sin\left(\frac{\phi}{2}\right)\sin\theta. \tag{28}$$

In other words, we have

$$\phi = 2\arcsin\left(\sin\left(\frac{\pi}{4J+6}\right)/\sin\theta\right). \tag{29}$$

3 Distributed exact Grover's algorithm

In this section, we mainly describe the distributed exact Grover's algorithm (DEGA), which decomposes the original search problem into $\lfloor n/2 \rfloor$ parts.

3.1 Subfunctions

As mentioned in [27], an n-qubit Boolean function $f: \{0,1\}^n \to \{0,1\}$ can be split into two subfunctions $f_{\text{even/odd}}: \{0,1\}^{n-1} \to \{0,1\}$,

$$f_{\text{even}}(x) = f(x_0x_1\cdots x_{i-1}0x_{i+1}\cdots x_{n-2}x_{n-1}), \tag{30}$$

$$f_{\text{odd}}(x) = f(x_0x_1\cdots x_{i-1}1x_{i+1}\cdots x_{n-2}x_{n-1}), \tag{31}$$

where $x = x_0x_1\cdots x_{i-1}x_{i+1}\cdots x_{n-2}x_{n-1} \in \{0,1\}^{n-1}$.

Furthermore, we can divide the original function into 2^k subfunctions $f_p: \{0,1\}^{n-k} \to \{0,1\}$:

$$f_p(x) = f(\underbrace{x_0x_1\cdots x_{n-k-1}}_{n-k}\underbrace{y_{p_0}y_{p_1}\cdots y_{p_{k-1}}}_{k}), \tag{32}$$

where $k \in \{1,2,\cdots,n-1\}$, $x = x_0x_1\cdots x_{n-k-1} \in \{0,1\}^{n-k}$, and $y_{p_0}y_{p_1}\cdots y_{p_{k-1}} \in \{0,1\}^k$ is the binary representation of $p \in \{0,1,\cdots,2^k-1\}$.

More generally, $y_{p_0}y_{p_1}\cdots y_{p_{k-1}}$ can be at any k positions in x of the original function f. For instance, we can also obtain another 2^k subfunctions $f_q : \{0,1\}^{n-k} \to \{0,1\}$:

$$f_q(x) = f(\underbrace{y_{q_0}y_{q_1}\cdots y_{q_{k-1}}}_{k}\underbrace{x_kx_{k+1}\cdots x_{n-1}}_{n-k}), \tag{33}$$

where $k \in \{1,2,\cdots,n-1\}$, $x = x_kx_{k+1}\cdots x_{n-1} \in \{0,1\}^{n-k}$, and $y_{q_0}y_{q_1}\cdots y_{q_{k-1}} \in \{0,1\}^k$ is the binary representation of $q \in \{0,1,\cdots,2^k-1\}$.

Next, we need to describe our partitioning strategy for the original function before providing our algorithm. We will generate a total of $\lfloor n/2 \rfloor$ subfunctions $g_i(m_i)$ for our algorithm, where $i \in \{0,1,\cdots,\lfloor n/2 \rfloor - 1\}$. The specific generation method is as follows.

When $i \in \{0,1,\cdots,\lfloor n/2 \rfloor - 2\}$, we choose first $2i$ and last $n-2(i+1)$ bits in x to divide f, then we can get 2^{n-2} subfunctions $f_{i,j} : \{0,1\}^2 \to \{0,1\}$:

$$f_{i,j}(m_i) = f(\underbrace{y_{j_0}y_{j_1}\cdots y_{j_{2i-1}}}_{2i} m_i \underbrace{y_{j_{2i}}y_{j_{2i+1}}\cdots y_{j_{n-3}}}_{n-2(i+1)}), \tag{34}$$

where $m_i \in \{0,1\}^2$ and $y_{j_0}y_{j_1}\cdots y_{j_{2i-1}}y_{j_{2i}}y_{j_{2i+1}}\cdots y_{j_{n-3}} \in \{0,1\}^{n-2}$ is the binary representation of $j \in \{0,1,\cdots,2^{n-2}-1\}$. Afterwards, we generate a new function $g_i : \{0,1\}^2 \to \{0,1\}$ through these subfunctions,

$$g_i(m_i) = \mathrm{OR}(f_{i,0}(m_i), f_{i,1}(m_i), \cdots, f_{i,2^{n-2}-1}(m_i)), \tag{35}$$

where $m_i \in \{0,1\}^2$. Besides,

$$\mathrm{OR}(x) = \begin{cases} 1, & |x| \geq 1, \\ 0, & |x| = 0, \end{cases} \tag{36}$$

where $|x|$ is the Hamming weight of $x \in \{0,1\}^n$ (its number of 1's). It means we have acquired $\lfloor n/2 \rfloor - 1$ subfunctions $g_i : \{0,1\}^2 \to \{0,1\}$, where $i \in \{0,1,\cdots,\lfloor n/2 \rfloor - 2\}$.

When $i = \lfloor n/2 \rfloor - 1$, we choose first $2i$ bits in x to divide f, then we can get 2^{2i} subfunctions $f_{i,j} : \{0,1\}^{n-2i} \to \{0,1\}$:

$$f_{i,j}(m_i) = f(\underbrace{y_{j_0}y_{j_1}\cdots y_{j_{2i-1}}}_{2i}\underbrace{m_i}_{n-2i}), \tag{37}$$

where $m_i \in \{0,1\}^{n-2i}$ and $y_{j_0}y_{j_1}\cdots y_{j_{2i-1}} \in \{0,1\}^{2i}$ is the binary representation of $j \in \{0,1,\cdots,2^{2i}-1\}$. Afterwards, we also generate a new function $g_i : \{0,1\}^{n-2i} \to \{0,1\}$ by these subfunctions,

$$g_i(m_i) = \mathrm{OR}(f_{i,0}(m_i), f_{i,1}(m_i), \cdots, f_{i,2^{2i}-1}(m_i)), \tag{38}$$

where $m_i \in \{0,1\}^{n-2i}$. Specifically, when n is an even number, we have

$$n - 2i = n - 2(\lfloor n/2 \rfloor - 1) = n - 2\left(\frac{n}{2} - 1\right) = 2. \tag{39}$$

When n is an odd number, we have

$$n - 2i = n - 2(\lfloor n/2 \rfloor - 1) = n - 2\left(\frac{n-1}{2} - 1\right) = 3. \tag{40}$$

It means we have acquired the last subfunction $g_i : \{0,1\}^{n-2i} \to \{0,1\}$, where $i = \lfloor n/2 \rfloor - 1$.

3.2 Distributed exact Grover's algorithm

So far, we have successfully generated a total of $\lfloor n/2 \rfloor$ subfunctions $g_i(m_i)$ according to the original function $f(x)$ and n, where $i \in \{0,1,\cdots,\lfloor n/2 \rfloor - 1\}$. Additionally, we assume that obtaining the Oracle for each subfunction is simple. In other words, we can have $\lfloor n/2 \rfloor - 1$ Oracles

$$U_{g_i(x)} : |x\rangle \to (-1)^{g_i(x)}|x\rangle, \tag{41}$$

where $x \in \{0,1\}^2$, $i \in \{0,1,\cdots,\lfloor n/2 \rfloor - 2\}$, and τ_i is the target string of $g_i(x)$ such that $g_i(\tau_i) = 1$.

For the last subfunctions $g_i : \{0,1\}^{n-2i} \to \{0,1\}$, we first need to determine the parity of n. If n is an even number, we can also have the Oracle as in Eq. (41), where $i = \lfloor n/2 \rfloor - 1$. Otherwise, we can have another Oracle

$$R_{g_i(x)} : |x\rangle \to \mathrm{e}^{\mathrm{i}\phi \cdot g_i(x)}|x\rangle, \tag{42}$$

where $x \in \{0,1\}^3$, $i = \lfloor n/2 \rfloor - 1$, τ_i is the target string of $g_i(x)$ such that $g_i(\tau_i) = 1$, $\phi = 2\arcsin\left(\sin\left(\frac{\pi}{4J+6}\right)/\sin\theta\right)$, $J = \lfloor(\pi/2-\theta)/(2\theta)\rfloor$, and $\theta = \arcsin\left(\sqrt{1/2^3}\right)$.

Next, we provide a detailed introduction to the DEGA in Algorithm 3. The whole quantum circuit is shown in Fig. 5.

The target index string $\tau = \tau_0\tau_1\cdots\tau_{\lfloor n/2 \rfloor - 1} \in \{0,1\}^n$ will be successfully searched exactly after following the above algorithm steps. Therefore, the entire process of the DEGA is accomplished.

4 Analysis

In this section, we will first demonstrate the correctness of the DEGA. The original and modified Grover's algorithms, and existing distributed Grover's algorithms will next be compared to the approach.

4.1 Correctness

To verify the correctness of the DEGA, it is equivalent to proving that we can obtain the target index string $\tau \in \{0,1\}^n$ by Algorithm 3 exactly. Firstly, we give the

Algorithm 3 Distributed exact Grover's algorithm (includes only one target).

Input: (1) The number of qubits $n \geq 2$; (2) A function $f(x)$, where $f(x) = 0$ for all $x \in \{0,1\}^n$ except τ, for which $f(\tau) = 1$; (3) $\lfloor n/2 \rfloor$ subfunctions $g_i(x)$ as in Eq. (35) and Eq. (38) generated according to $f(x)$ and n, where $i \in \{0, 1, \cdots, \lfloor n/2 \rfloor - 1\}$.

Output: The target index string $\tau = \tau_0\tau_1\cdots\tau_{\lfloor n/2 \rfloor - 1} \in \{0,1\}^n$ such that $f(\tau) = 1$ with a probability of 100%.

Procedure:

Step 1. Initialize n qubits $|0\rangle^{\otimes n}$.

Step 2. Apply n Hadamard gates $H^{\otimes n}$.

Step 3. Let $i = 0$.

Step 4. Execute Grover operator $G_i = -H^{\otimes 2}U_0H^{\otimes 2}U_{g_i(x)}$ once, where $U_0 = I^{\otimes 2} - 2(|0\rangle\langle 0|)^{\otimes 2}$ and $U_{g_i(x)} = I^{\otimes 2} - 2|\tau_i\rangle\langle\tau_i|$ act on the $(2i)$-th and $(2i+1)$-th qubits, and τ_i is the target string of $g_i(x)$ such that $g_i(\tau_i) = 1$.

Step 5. Let $i = i + 1$.

Step 6. Repeat Step 4 to Step 5 until $i = \lfloor n/2 \rfloor - 1$.

Step 7. Determine the parity of n.

Step 8. If n is an even number, execute Grover operator $G_i = -H^{\otimes 2}U_0H^{\otimes 2}U_{g_i(x)}$ once, where $U_0 = I^{\otimes 2} - 2(|0\rangle\langle 0|)^{\otimes 2}$ and $U_{g_i(x)} = I^{\otimes 2} - 2|\tau_i\rangle\langle\tau_i|$ act on the $(2i)$-th and $(2i+1)$-th qubits, and τ_i is the target string of $g_i(x)$ such that $g_i(\tau_i) = 1$.

Step 9. If n is an odd number, execute operator $L_i = -H^{\otimes 3}R_0H^{\otimes 3}R_{g_i(x)}$ twice, where $R_0 = I^{\otimes 3} + (e^{i\phi} - 1)(|0\rangle\langle 0|)^{\otimes 3}$ and $R_{g_i(x)} = I^{\otimes 3} + (e^{i\phi} - 1)|\tau_i\rangle\langle\tau_i|$ act on the $(2i)$-th, $(2i+1)$-th, and $(2i+2)$-th qubits, and τ_i is the target string of $g_i(x)$ such that $g_i(\tau_i) = 1$.

Step 10. Measure each qubit in the basis $\{|0\rangle, |1\rangle\}$.

following theorem.

Theorem 1. *Each subfunction $g_i(x)$ [as in Eq. (35) and Eq. (38)] has its corresponding target string τ_i such that $g_i(\tau_i) = 1$, which satisfies*

$$\tau = \tau_0\tau_1\cdots\tau_{\lfloor n/2 \rfloor - 1}, \tag{43}$$

where $i \in \{0, 1, \cdots, \lfloor n/2 \rfloor - 1\}$ and $\tau \in \{0,1\}^n$ is the target string of the original function $f(x)$.

Proof. Without losing generality, suppose the target string can be expressed as $\tau = x_0x_1\cdots x_{n-1} \in \{0,1\}^n$.

The proof is divided into two parts.

Part 1: For $i \in \{0, 1, \cdots, \lfloor n/2 \rfloor - 2\}$, we first prove $\tau_i = x_{2i}x_{2i+1} \in \{0,1\}^2$.

As can be seen from Eq. (34), we can get 2^{n-2} subfunctions $f_{i,j} : \{0,1\}^2 \to \{0,1\}$:

$$f_{i,j}(m_i) = f(\underbrace{y_{j_0}y_{j_1}\cdots y_{j_{2i-1}}}_{2i}\, m_i\, \underbrace{y_{j_{2i}}y_{j_{2i+1}}\cdots y_{j_{n-3}}}_{n-2(i+1)}), \tag{44}$$

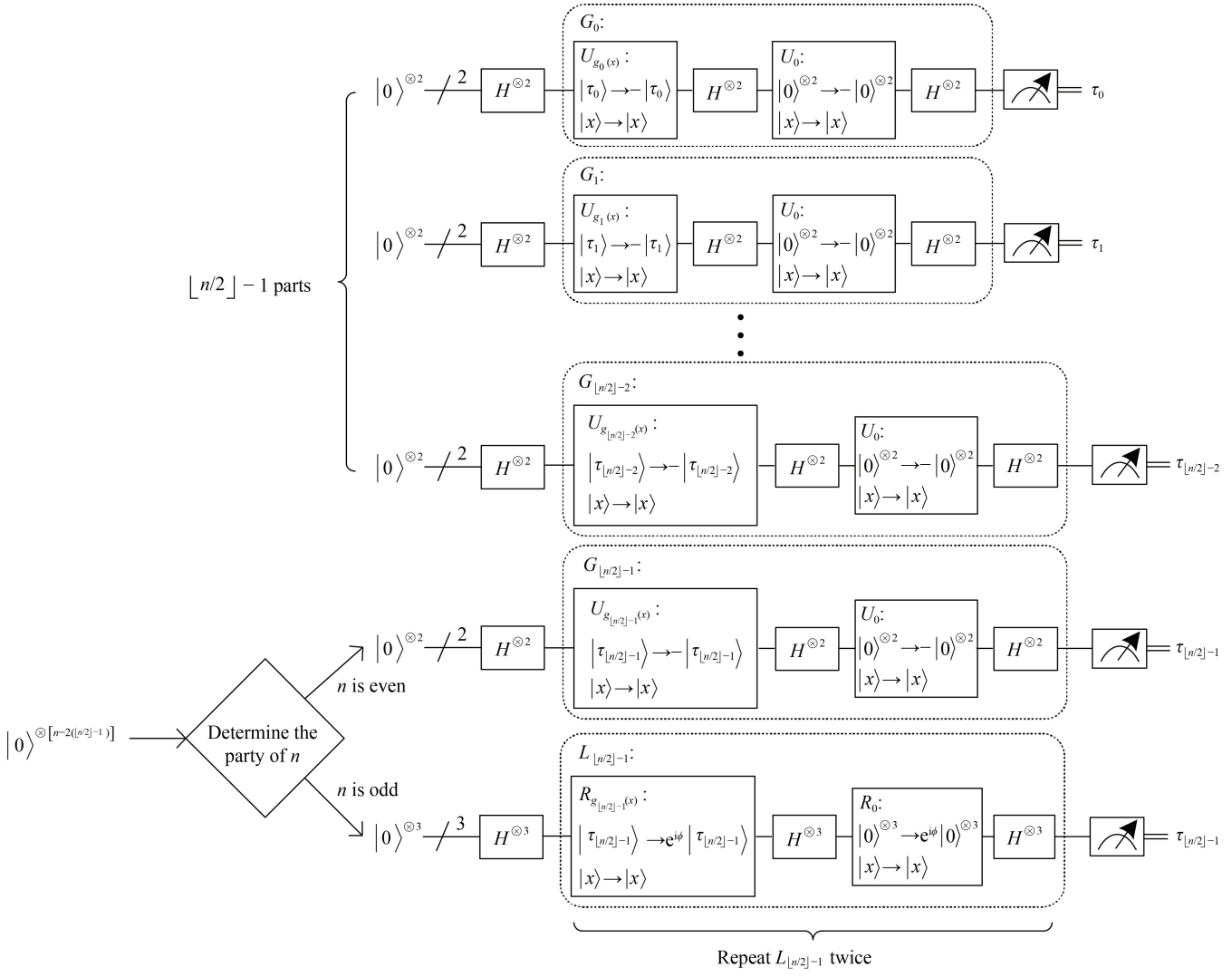


Fig. 5 Quantum circuit of the distributed exact Grover's algorithm (DEGA).

where $m_i \in \{0,1\}^2$ and $y_{j_0}y_{j_1}\cdots y_{j_{2i-1}}y_{j_{2i}}y_{j_{2i+1}}\cdots y_{j_{n-3}} \in \{0,1\}^{n-2}$ is the binary representation of $j \in \{0,1,\cdots, 2^{n-2}-1\}$. Thus, there exists $j' = y_{j'_0}y_{j'_1}\cdots y_{j'_{2i-1}}y_{j'_{2i}}y_{j'_{2i+1}}\cdots y_{j'_{n-3}} \in \{0,1\}^{n-2}$, such that

$$\underbrace{y_{j'_0}y_{j'_1}\cdots y_{j'_{2i-1}}}_{2i}\underbrace{y_{j'_{2i}}y_{j'_{2i+1}}\cdots y_{j'_{n-3}}}_{n-2(i+1)} = x_0x_1\cdots x_{2i-1}x_{2i+2}\cdots x_{n-1}, \tag{45}$$

where $x_0x_1\cdots x_{2i-1}x_{2i+2}\cdots x_{n-1} \in \{0,1\}^{n-2}$ represents $\tau = x_0x_1\cdots x_{2i-1}x_{2i}x_{2i+1}x_{2i+2}\cdots x_{n-1} \in \{0,1\}^n$ removes $x_{2i}x_{2i+1} \in \{0,1\}^2$.

In other words, we have a subfunction $f_{i,j'}: \{0,1\}^2 \to \{0,1\}$:

$$f_{i,j'}(m_i) = \begin{cases} 1, & m_i = x_{2i}x_{2i+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{46}$$

where $m_i \in \{0,1\}^2$. For the remaining subfunctions $f_{i,j\neq j'}: \{0,1\}^2 \to \{0,1\}$:

$$f_{i,j\neq j'}(m_i) \equiv 0, \tag{47}$$

where $m_i \in \{0,1\}^2$.

Afterwards, we generate a new function $g_i: \{0,1\}^2 \to \{0,1\}$ by these subfunctions,

$$g_i(m_i) = \text{OR}(f_{i,0}(m_i), f_{i,1}(m_i), \cdots, f_{i,2^{n-2}-1}(m_i)) \tag{48}$$

$$= \begin{cases} 1, & m_i = x_{2i}x_{2i+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{49}$$

where $m_i \in \{0,1\}^2$. Therefore, let $\tau_i = x_{2i}x_{2i+1}$. It is the target string of $g_i(m_i)$ such that $g_i(\tau_i) = 1$.

Part 1 has been proved.

Part 2: For $i = \lfloor n/2 \rfloor - 1$, we prove that

$$\tau_i = x_{2i}x_{2i+1}\cdots x_{n-1} = \begin{cases} x_{2i}x_{2i+1} = x_{n-2}x_{n-1} \in \{0,1\}^2, & n \text{ is even}, \\ x_{2i}x_{2i+1}x_{2i+2} = x_{n-3}x_{n-2}x_{n-1} \in \{0,1\}^3, & n \text{ is odd}. \end{cases} \tag{50}$$

The proof of Part 2 is similar to Part 1. As can be seen from Eq. (37), we can get 2^{2i} subfunctions $f_{i,j}: \{0,1\}^{n-2i} \to \{0,1\}$:

$$f_{i,j}(m_i) = f(\underbrace{y_{j_0}y_{j_1}\cdots y_{j_{2i-1}}}_{2i}\underbrace{m_i}_{n-2i}), \tag{51}$$

where $m_i \in \{0,1\}^{n-2i}$ and $y_{j_0}y_{j_1}\cdots y_{j_{2i-1}} \in \{0,1\}^{2i}$ is the binary representation of $j \in \{0,1,\cdots, 2^{2i}-1\}$.

Thus, there exists $j'' = y_{j''_0}y_{j''_1}\cdots y_{j''_{2i-1}} \in \{0,1\}^{2i}$, such that

$$\underbrace{y_{j''_0}y_{j''_1}\cdots y_{j''_{2i-1}}}_{2i} = x_0x_1\cdots x_{2i-1}, \tag{52}$$

where $x_0x_1\cdots x_{2i-1} \in \{0,1\}^{2i}$ represents $\tau = x_0x_1\cdots x_{2i-1}x_{2i}x_{2i+1}\cdots x_{n-1} \in \{0,1\}^n$ removes $x_{2i}x_{2i+1}\cdots x_{n-1} \in \{0,1\}^{n-2i}$.

In other words, we have a subfunction $f_{i,j''}: \{0,1\}^{n-2i} \to \{0,1\}$:

$$f_{i,j''}(m_i) = \begin{cases} 1, & m_i = x_{2i}x_{2i+1}\cdots x_{n-1}, \\ 0, & \text{otherwise}, \end{cases} \tag{53}$$

where $m_i \in \{0,1\}^{n-2i}$. For the remaining subfunctions $f_{i,j\neq j''}: \{0,1\}^{n-2i} \to \{0,1\}$:

$$f_{i,j\neq j''}(m_i) \equiv 0, \tag{54}$$

where $m_i \in \{0,1\}^{n-2i}$.

Afterwards, we also generate a new function $g_i: \{0,1\}^{n-2i} \to \{0,1\}$ by these subfunctions,

$$g_i(m_i) = \text{OR}(f_{i,0}(m_i), f_{i,1}(m_i), \cdots, f_{i,2^{2i}-1}(m_i)) \tag{55}$$

$$= \begin{cases} 1, & m_i = x_{2i}x_{2i+1}\cdots x_{n-1}, \\ 0, & \text{otherwise}, \end{cases} \tag{56}$$

where $m_i \in \{0,1\}^{n-2i}$. Specifically, when n is an even number, we have

$$n - 2i = n - 2(\lfloor n/2 \rfloor - 1) = n - 2\left(\frac{n}{2} - 1\right) = 2. \tag{57}$$

When n is an odd number, we have

$$n - 2i = n - 2(\lfloor n/2 \rfloor - 1) = n - 2\left(\frac{n-1}{2} - 1\right) = 3. \tag{58}$$

Therefore, let

$$\tau_i = x_{2i}x_{2i+1}\cdots x_{n-1} = \begin{cases} x_{2i}x_{2i+1} = x_{n-2}x_{n-1} \in \{0,1\}^2, & n \text{ is even}, \\ x_{2i}x_{2i+1}x_{2i+2} = x_{n-3}x_{n-2}x_{n-1} \in \{0,1\}^3, & n \text{ is odd}. \end{cases} \tag{59}$$

It is the target string of $g_i(m_i)$ such that $g_i(\tau_i) = 1$.

Part 2 has been proved.

Gather all subfunction target strings, then the target string of the original function $f(x)$ is

$$\tau = \tau_0\tau_1\cdots\tau_{\lfloor n/2 \rfloor - 1} = x_0x_1\cdots x_{n-1} \in \{0,1\}^n. \tag{60}$$

□

So far, we have proved that the target string corresponding to each subfunction $g_i(m_i)$ is τ_i, which satisfies $\tau = \tau_0\tau_1\cdots\tau_{\lfloor n/2 \rfloor - 1}$, where $i \in \{0,1,\cdots, \lfloor n/2 \rfloor - 1\}$.

In addition, only when 1/4 of the data in the database meet the search conditions, the success probability of Grover's algorithm is 1, which has been strictly proved mathematically [15]. In other words, a 2-qubit Grover's algorithm of finding one in four is exact. The modified version of Grover's algorithm [16] searches a target state with full successful rate. In particular, a 3-qubit modified Grover's algorithm is also exact when $n = 3$.

Next, we demonstrate the correctness of Algorithm 3 by the following Theorem 2.

Theorem 2. (**Correctness of DEGA**) *For an n-qubit search problem that includes only one target index string, suppose that the goal can be expressed as $\tau = x_0x_1\cdots x_{n-1} \in \{0,1\}^n$ of the original function $f(x)$. Then we can obtain $\tau = \tau_0\tau_1\cdots\tau_{\lfloor n/2\rfloor-1}$ by Algorithm 3 exactly, where τ_i is the target string of $g_i(x)$ and satisfies*

$$\tau_i = \begin{cases} x_{2i}x_{2i+1}, & i \in \{0,1,\cdots,\lfloor n/2\rfloor-2\}, \\ x_{2i}x_{2i+1}\cdots x_{n-1}, & i = \lfloor n/2\rfloor-1, \end{cases} \tag{61}$$

where

$$x_{2i}x_{2i+1}\cdots x_{n-1} = \begin{cases} x_{2i}x_{2i+1} = x_{n-2}x_{n-1}, & n \text{ is even}, \\ x_{2i}x_{2i+1}x_{2i+2} = x_{n-3}x_{n-2}x_{n-1}, & n \text{ is odd}. \end{cases} \tag{62}$$

Proof. From Section 3.1, we have declared how to generate $\lfloor n/2\rfloor$ subfunctions $g_i(x)$ as in Eq. (35) and Eq. (38) according to $f(x)$ and n, which decomposes the original search problem into $\lfloor n/2\rfloor$ parts, where $i \in \{0,1,\cdots,\lfloor n/2\rfloor-1\}$.

Furthermore, suppose that the goal can be expressed as $\tau = x_0x_1\cdots x_{n-1} \in \{0,1\}^n$ of the original function $f(x)$. Theorem 1 has already explained each subfunction $g_i(x)$ has its corresponding target string τ_i such that $g_i(\tau_i) = 1$, which satisfies

$$\tau = \tau_0\tau_1\cdots\tau_{\lfloor n/2\rfloor-1}. \tag{63}$$

Last but not least, we will accurately obtain the solution of each part. For $i \in \{0,1,\cdots,\lfloor n/2\rfloor-2\}$, by running the 2-qubit Grover's algorithm, we can precisely determine $\tau_i = x_{2i}x_{2i+1} \in \{0,1\}^2$. For $i = \lfloor n/2\rfloor-1$, by running the 2-qubit Grover's algorithm when n is even or the 3-qubit modified Grover's algorithm when n is odd, we can exactly obtain

$$\tau_i = x_{2i}x_{2i+1}\cdots x_{n-1} = \begin{cases} x_{2i}x_{2i+1} = x_{n-2}x_{n-1} \in \{0,1\}^2, & n \text{ is even}, \\ x_{2i}x_{2i+1}x_{2i+2} = x_{n-3}x_{n-2}x_{n-1} \in \{0,1\}^3, & n \text{ is odd}. \end{cases} \tag{64}$$

In conclusion, we can obtain $\tau = \tau_0\tau_1\cdots\tau_{\lfloor n/2\rfloor-1}$ through Algorithm 3 with a probability of 100%, where τ_i is the target string of $g_i(x)$, where $i \in \{0,1,\cdots,\lfloor n/2\rfloor-1\}$. □

4.2 Comparison

In this subsection, the original and modified Grover's algorithms, and existing distributed Grover's algorithms will be compared to our approach.

Compared with the original Grover's algorithm, the DEGA is as exact as the modified version of Grover's algorithm by Long, which means the theoretical probability of finding the objective state is 100%.

In order to compare the circuit depth, we give the following definition.

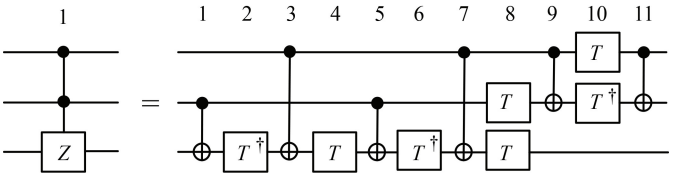


Fig. 6 Equivalent quantum circuit of Toffoli gate (or C^2Z gate), where T represent $\pi/8$ gate and $T^\dagger$ represents the conjugate transpose of T.

Definition 1. (**Depth of circuit**) The depth of circuit is defined as the longest path from the input to the output, moving forward in time along qubit wires.

The depth of the circuit on the left, for example, is 1, whereas the right is 11 (see Fig. 6).

Furthermore, the depth of the quantum circuit depends on the circuit composed of specific quantum gates. Then we need to clarify the construction circuit of the Oracle in the quantum search algorithms. Due to the method by Qiu *et al.* [28], we will introduce a construction for the Oracle in the original and modified Grover's algorithms.

$U_{f(x)} = I^{\otimes n} - 2|\tau\rangle\langle\tau|$ is employed in the Grover operator $G = -H^{\otimes n}U_0H^{\otimes n}U_{f(x)}$ to flip the phase of the target state $|\tau\rangle = |x_0x_1\cdots x_{n-1}\rangle$, where $\tau = x_0x_1\cdots x_{n-1} \in \{0,1\}^n$. Thus, the specific construction circuit of $U_{f(x)}$ (see Fig. 7) is as follows:

$$U_{f(x)} = \left(\bigotimes_{i=0}^{n-1} X^{1-x_i}\right)\left(C^{n-1}Z\right)\left(\bigotimes_{i=0}^{n-1} X^{1-x_i}\right), \tag{65}$$

where $C^{n-1}Z$ will only flip the phase of $|1\rangle^{\otimes n}$,

$$\left(C^{n-1}Z\right)|x\rangle = \begin{cases} -|x\rangle, & |x\rangle = |1\rangle^{\otimes n}, \\ |x\rangle, & \text{otherwise}, \end{cases} \tag{66}$$

where $x = x_0x_1\cdots x_{n-1} \in \{0,1\}^n$. Besides, it should be noted that

$$X^{1-x_i} = \begin{cases} X, & x_i = 0, \\ I, & x_i = 1, \end{cases} \tag{67}$$

where $i \in \{0,1,\cdots,n-1\}$. In particular, the construction circuit of $U_0 = I^{\otimes n} - 2(|0\rangle\langle 0|)^{\otimes n}$ (see Fig. 8) is

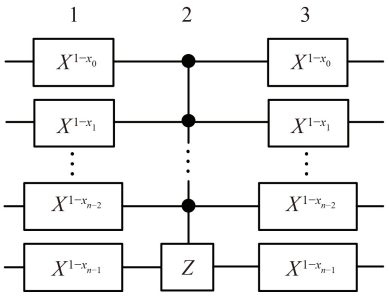


Fig. 7 The quantum circuit corresponding to the Oracle $U_{f(x)}$.

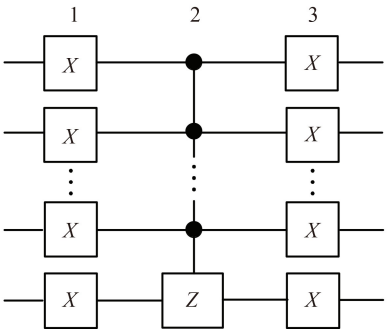


Fig. 8 The quantum circuit corresponding to the Oracle U_0.

$$U_0 = \left(X^{\otimes n}\right)\left(C^{n-1}Z\right)\left(X^{\otimes n}\right). \tag{68}$$

It can be found that the quantum circuit in Fig. 7 or Fig. 8 has a circuit depth of 3.

Similarly, we can also give the specific construction circuit of $R_{f(x)} = I^{\otimes n} + (\mathrm{e}^{\mathrm{i}\phi} - 1)|\tau\rangle\langle\tau|$ in the operator $L = -H^{\otimes n}R_0H^{\otimes n}R_{f(x)}$ (see Fig. 9) as follows:

$$R_{f(x)} = \left(\bigotimes_{i=0}^{n-1} X^{1-x_i}\right)\left(C^{n-1}R(\phi)\right)\left(\bigotimes_{i=0}^{n-1} X^{1-x_i}\right), \tag{69}$$

where $R(\phi)$ represents the rotation gate,

$$R(\phi) = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\phi} \end{pmatrix}, \tag{70}$$

and $C^{n-1}R(\phi)$ will only rotate the phase of $|1\rangle^{\otimes n}$,

$$\left(C^{n-1}R(\phi)\right)|x\rangle = \begin{cases} \mathrm{e}^{\mathrm{i}\phi}|x\rangle, & |x\rangle = |1\rangle^{\otimes n}, \\ |x\rangle, & \text{otherwise}, \end{cases} \tag{71}$$

where $x = x_0x_1\cdots x_{n-1} \in \{0,1\}^n$. In particular, the construction circuit of $R_0 = I^{\otimes n} + (\mathrm{e}^{\mathrm{i}\phi} - 1)(|0\rangle\langle 0|)^{\otimes n}$ in operator L (see Fig. 10) is

$$R_0 = \left(X^{\otimes n}\right)\left(C^{n-1}R(\phi)\right)\left(X^{\otimes n}\right). \tag{72}$$

Quantum circuit in Fig. 9 or Fig. 10 also has a circuit depth of 3.

If the target state is $|1\rangle^{\otimes n}$, then the depth of circuit is

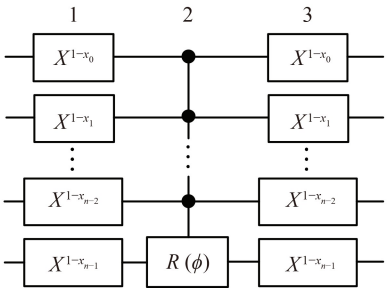


Fig. 9 The quantum circuit corresponding to the Oracle $R_{f(x)}$.

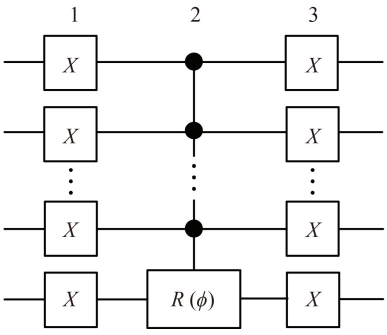


Fig. 10 The quantum circuit corresponding to the Oracle R_0.

only 1. We ignore this special case. By the way, perhaps there are other better methods to construct the quantum circuits of Oracle, but we will not cover it here. Throughout this paper, we use the same methods described above to construct the Oracle circuits in the Grover operator G and operator L.

After identifying the specific quantum gates in the original and modified Grover's algorithms, we might calculate the circuit depth of these two methods.

Theorem 3. *The circuit depths for the original and modified Grover's algorithms are* $1 + 8\left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor$ *and* $9 + 8\left\lfloor \frac{\pi}{4}\sqrt{2^n} - \frac{1}{2} \right\rfloor$, *respectively.*

Proof. In Grover's algorithm, Grover operator $G = -H^{\otimes n}U_0H^{\otimes n}U_{f(x)}$ is executed with $\left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor$ times, where $dep(U_f) = dep(U_0) = 3$ and $dep(H^{\otimes n}) = 1$. Thus, the circuit depth of Grover's algorithm is

$$\begin{aligned} dep(\text{Grover}) =& 1 + \left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor \cdot (3 + 1 + 3 + 1) \\ =& 1 + 8\left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor. \end{aligned} \tag{73}$$

Similarly, since operator $L = -H^{\otimes n}R_0H^{\otimes n}R_{f(x)}$ is executed with $J + 1$ times, where $J = \lfloor(\pi/2 - \theta)/(2\theta)\rfloor$, $\theta = \arcsin\left(\sqrt{1/2^n}\right)$, and $dep(R_f) = dep(R_0) = 3$, the circuit depth of the modified Grover's algorithm is

$$\begin{aligned} dep(\text{modified Grover}) =& 1 + \left(\left\lfloor \frac{\pi}{4}\sqrt{2^n} - \frac{1}{2} \right\rfloor + 1\right) \\ & \cdot (3 + 1 + 3 + 1) \\ =& 9 + 8\left\lfloor \frac{\pi}{4}\sqrt{2^n} - \frac{1}{2} \right\rfloor. \end{aligned} \tag{74}$$

□

It has been discovered that the circuit depths of the original and modified Grover's algorithms deepen as n increases.

Next, we give the circuit depth of the DEGA by the following theorem.

Theorem 4. *The circuit depth of Algorithm 3 is* $8(n \bmod 2) + 9$.

Proof. When n is an even number, the DEGA will run $\lfloor n/2 \rfloor$ parts 2-qubit Grover's algorithm, and each part

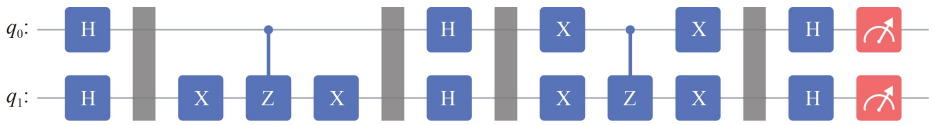


Fig. 11 Quantum circuit of the 2-qubit DEGA (target string $\tau = 01$).

only needs to iterate the Grover operator G once, so the circuit depth is

$$dep(\text{DEGA}, n \text{ is even}) = 1 + 1 \cdot (3 + 1 + 3 + 1) = 9. \tag{75}$$

When n is an odd number, the DEGA will run $\lfloor n/2 \rfloor - 1$ parts 2-qubit Grover's algorithm and 1 part 3-qubit modified Grover's algorithm, and the last part needs to iterate the operator L twice, so the circuit depth is

$$dep(\text{DEGA}, n \text{ is odd}) = 1 + 2 \cdot (3 + 1 + 3 + 1) = 17. \tag{76}$$

In other words, the actual depth of our circuit is

$$dep(\text{DEGA}) = 8(n \bmod 2) + 9 = \begin{cases} 9, & n \text{ is even}, \\ 17, & n \text{ is odd}. \end{cases} \tag{77}$$

□

Besides, each portion of the DEGA only comprises two or three qubits, making physical experiment verification trivial.

Therefore, the actual circuit depth will be shallower comparing with the original and modified Grover's algorithms. The circuit depth of the DEGA only depends on the parity of n, and it is not deepened as n increases.

Next, we will contrast the DEGA with the existing distributed Grover's algorithms.

In Ref. [27], they split the original n-qubit Boolean function $f : \{0,1\}^n \to \{0,1\}$ two subfunctions $f_{\text{even/odd}} : \{0,1\}^{n-1} \to \{0,1\}$ as in Eq. (30) and Eq. (31), and then executed $(n-1)$-qubit Grover's algorithm for each subfunction. After that, they will decide whether 0 or 1 to add to the i-th bit based on which subfunction can correctly run the result (0 for f_{even} and 1 for f_{odd}). That is to say, they only acquired $n-1$ qubits of the target state rather than the target state itself directly. The total number of qubits used in their scheme is $2(n-1)$.

In Ref. [28], they have solved the case of multiple targets, and presented the serial and parallel distributed Grover's algorithms and divided the original function f into 2^k subfunctions $f_p : \{0,1\}^{n-k} \to \{0,1\}$ as in Eq. (32), where 2^k represents the number of computing nodes. The total number of qubits used in their parallel scheme is $2^k(n-k)$.

Finally, we will analyse why DEGA is not suitable for solving multiple target strings. To begin with, when the target string is a single string, the final state is a direct product state, which may be thought of as the direct product of several quantum states. At this stage, each node can accurately search for some information of the initial target string, yielding the original target string. However, when the target string consists of multiple strings, the final state may be an entangled state, which cannot be decomposed into a direct product of multiple quantum states at this point. In other words, DEGA is not applicable when the target string includes several strings and its related quantum state is an entangled state. Therefore, how to design a distributed quantum algorithm to solve the exact search problem with the case of multiple targets is still an open problem. Perhaps additional quantum operations, such as performing quantum gates between nodes, are needed to enable entanglement of the final state.

5 Experiment

In this section, we will provide particular situations of the DEGA on MindQuantum (a quantum software). These experiments will further demonstrate the correctness of the DEGA and explicate the specific steps of implement n-qubit DEGA, where $n \in \{2,3,4,5\}$. After that, we will simulate the 5-qubit DEGA running in the depolarization channel and compare with the original and modified Grover's algorithms. Since our circuit is shallower, it will be more resistant to the depolarization channel noise.

5.1 The 2-qubit DEGA, the target string $\tau = 01$

Let Boolean function $f : \{0,1\}^2 \to \{0,1\}$. Suppose

$$f(x) = \begin{cases} 1, & x = 01, \\ 0, & x \neq 01, \end{cases} \tag{78}$$

where $x \in \{0,1\}^2$ and $\tau = 01$ is the target string. When $n = 2$, the DEGA is a 2-qubit Grover's algorithm. Thus, we can build its circuit (see Fig. 11).

By sampling the circuit in Fig. 11 10 000 times, the sampling results can be found in Fig. 12.

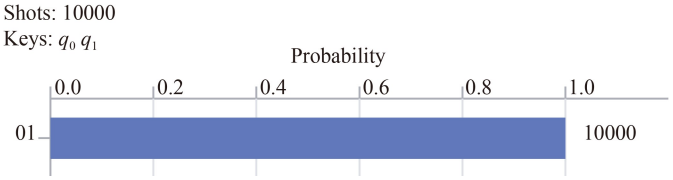


Fig. 12 The sampling results of the 2-qubit DEGA (target string $\tau = 01$).

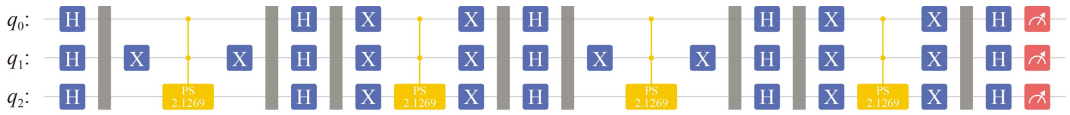


Fig. 13 Quantum circuit of the 3-qubit DEGA (target string $\tau = 101$). The parameter in the circuit is $\phi = 2\arcsin\left(\sin\left(\frac{\pi}{4J+6}\right)/\sin\theta\right) = 2.1268800471555034 \approx 2.1269$, where $J = \lfloor(\pi/2-\theta)/(2\theta)\rfloor$ and $\theta = \arcsin\left(\sqrt{1/2^3}\right)$.

The sampling results demonstrate that $\tau = 01$ is obtained exactly after measurement. Besides, the total number of quantum gates is 14 and the circuit depth is 9.

5.2 The 3-qubit DEGA, the target string $\tau = 101$

Let Boolean function $f : \{0,1\}^3 \to \{0,1\}$. Suppose

$$f(x) = \begin{cases} 1, & x = 101, \\ 0, & x \neq 101, \end{cases} \tag{79}$$

where $x \in \{0,1\}^3$ and $\tau = 101$ is the target string. When $n = 3$, the DEGA is a 3-qubit modified Grover's algorithm. Thus, we can build its circuit (see Fig. 13). Note that PhaseShift (called the PS gate) is a single-qubit gate, whose matric is as follows:

$$\mathrm{PS}(\phi) = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\phi} \end{pmatrix}. \tag{80}$$

By sampling the circuit in Fig. 13 10 000 times, the sampling results can be found in Fig. 14.

The sampling results demonstrate that $\tau = 101$ is obtained exactly after measurement. Besides, the total number of quantum gates is 35 and the circuit depth is 17.

5.3 The 4-qubit DEGA, the target string $\tau = 1001$

Let Boolean function $f : \{0,1\}^4 \to \{0,1\}$. Suppose

$$f(x) = \begin{cases} 1, & x = 1001, \\ 0, & x \neq 1001, \end{cases} \tag{81}$$

where $x \in \{0,1\}^4$ and $\tau = 1001$ is the target string. When $n = 4$, we can decomposes the original search problem into $\lfloor n/2 \rfloor = 2$ parts. To be specific, we choose last 2 bits in x to divide f, then we can get $2^2 = 4$ subfunctions $f_{0,j} : \{0,1\}^2 \to \{0,1\}$:

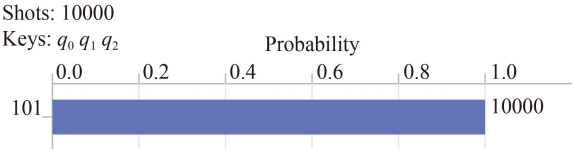


Fig. 14 The sampling results of the 3-qubit DEGA (target string $\tau = 101$).

$$f_{0,0}(m_0) = f(m_0 00), \tag{82}$$

$$f_{0,1}(m_0) = f(m_0 01), \tag{83}$$

$$f_{0,2}(m_0) = f(m_0 10), \tag{84}$$

$$f_{0,3}(m_0) = f(m_0 11), \tag{85}$$

where $m_0 \in \{0,1\}^2$ and $j \in \{0,1,2,3\}$. Obviously, $f_{0,0}(m_0) = f_{0,2}(m_0) = f_{0,3}(m_0) \equiv 0$, and

$$f_{0,1}(m_0) = \begin{cases} 1, & m_0 = 10, \\ 0, & m_0 \neq 10, \end{cases} \tag{86}$$

where $m_0 \in \{0,1\}^2$ and 10 is the target substring.

Afterwards, we generate a new function $g_0 : \{0,1\}^2 \to \{0,1\}$ through above four subfunctions,

$$\begin{aligned} g_0(m_0) =& \mathrm{OR}\left(f_{0,0}(m_0), f_{0,1}(m_0), f_{0,2}(m_0), f_{0,3}(m_0)\right) \\ =& \begin{cases} 1, & m_0 = 10, \\ 0, & m_0 \neq 10, \end{cases} \end{aligned} \tag{87}$$

where $m_0 \in \{0,1\}^2$.

Similarly, we choose first 2 bits in x to divide f, then we can get $2^2 = 4$ subfunctions $f_{1,j} : \{0,1\}^2 \to \{0,1\}$:

$$f_{1,0}(m_1) = f(00 m_1), \tag{88}$$

$$f_{1,1}(m_1) = f(01 m_1), \tag{89}$$

$$f_{1,2}(m_1) = f(10 m_1), \tag{90}$$

$$f_{1,3}(m_1) = f(11 m_1), \tag{91}$$

where $m_1 \in \{0,1\}^2$ and $j \in \{0,1,2,3\}$. Obviously, $f_{1,0}(m_1) = f_{1,1}(m_1) = f_{1,3}(m_1) \equiv 0$, and

$$f_{1,2}(m_1) = \begin{cases} 1, & m_1 = 01, \\ 0, & m_1 \neq 01, \end{cases} \tag{92}$$

where $m_1 \in \{0,1\}^2$ and 01 is the target substring.

Afterwards, we generate a new function $g_1 : \{0,1\}^2 \to \{0,1\}$ through above four subfunctions,

$$\begin{aligned} g_1(m_1) =& \mathrm{OR}\left(f_{1,0}(m_1), f_{1,1}(m_1), f_{1,2}(m_1), f_{1,3}(m_1)\right) \\ =& \begin{cases} 1, & m_1 = 01, \\ 0, & m_1 \neq 01, \end{cases} \end{aligned} \tag{93}$$

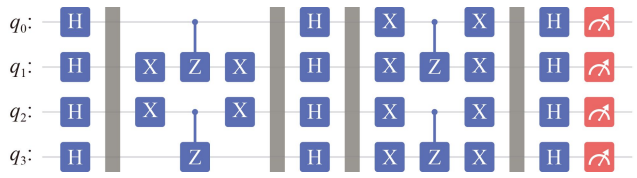


Fig. 15 Quantum circuit of the 4-qubit DEGA (target string $\tau = 1001$).

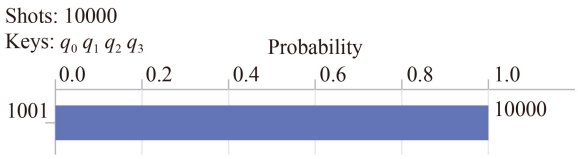


Fig. 16 The sampling results of the 4-qubit DEGA (target string $\tau = 1001$).

where $m_1 \in \{0,1\}^2$.

So far, we have successfully obtained the subfunctions of two parts, $g_0(m_0)$ and $g_1(m_1)$. Next, we need to run the 2-qubit Grover's algorithm for each part. Thus, we can build the completed circuit (see Fig. 15). Note that since the reading order on MindQuantum is from right to left, which is the opposite of our reading order from left to right, the 2-qubit Grover's algorithm corresponding to $g_1(m_1)$ is located on the top of the circuit.

By sampling the circuit in Fig. 15 10 000 times, the sampling results can be found in Fig. 16.

The sampling results demonstrate that $\tau = 1001$ is obtained exactly after measurement. Besides, the total number of quantum gates is 28 and the circuit depth is 9.

If the original or modified Grover's algorithm is employed to search 1001, we can build the completed circuits (see Fig. 17 and Fig. 18) respectively. By sampling the circuits in Fig. 17 and Fig. 18 10 000 times respectively, the sampling results can be found in Fig. 19 and Fig. 20.

It can be found that the probability of Grover's algorithm acquiring target string $\tau = 1001$ is 0.9627, which is not exact, while the modified Grover's algorithm can accurately obtain $\tau = 1001$. The number of quantum gates required by two algorithms is 70, and the circuit depth is 25.

5.4 The 5-qubit DEGA, the target string $\tau = 01001$

Let Boolean function $f : \{0,1\}^5 \to \{0,1\}$. Suppose

$$f(x) = \begin{cases} 1, & x = 01001, \\ 0, & x \neq 01001, \end{cases} \tag{94}$$

where $x \in \{0,1\}^5$ and $\tau = 01001$ is the target string. When $n = 5$, we can decomposes the original search problem into $\lfloor n/2 \rfloor = 2$ parts. To be specific, we choose last 2 bits in x to divide f, then we can get $2^2 = 4$ subfunctions $f_{0,j} : \{0,1\}^3 \to \{0,1\}$:

$$f_{0,0}(m_0) = f(m_0 00), \tag{95}$$

$$f_{0,1}(m_0) = f(m_0 01), \tag{96}$$

$$f_{0,2}(m_0) = f(m_0 10), \tag{97}$$

$$f_{0,3}(m_0) = f(m_0 11), \tag{98}$$

where $m_0 \in \{0,1\}^3$ and $j \in \{0,1,2,3\}$. Obviously, $f_{0,0}(m_0) = f_{0,2}(m_0) = f_{0,3}(m_0) \equiv 0$, and

$$f_{0,1}(m_0) = \begin{cases} 1, & m_0 = 010, \\ 0, & m_0 \neq 010, \end{cases} \tag{99}$$

where $m_0 \in \{0,1\}^3$ and 010 is the target substring.

Afterwards, we generate a new function $g_0 : \{0,1\}^3 \to \{0,1\}$ through above four subfunctions,

$$\begin{aligned} g_0(m_0) =& \text{OR}\left(f_{0,0}(m_0), f_{0,1}(m_0), f_{0,2}(m_0), f_{0,3}(m_0)\right) \\ =& \begin{cases} 1, & m_0 = 010, \\ 0, & m_0 \neq 010, \end{cases} \end{aligned} \tag{100}$$

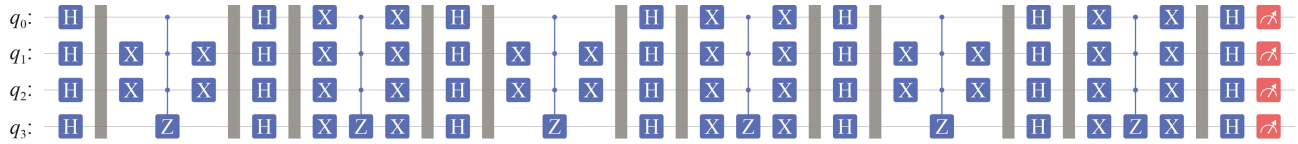


Fig. 17 Quantum circuit of the 4-qubit Grover's algorithm (target string $\tau = 1001$).

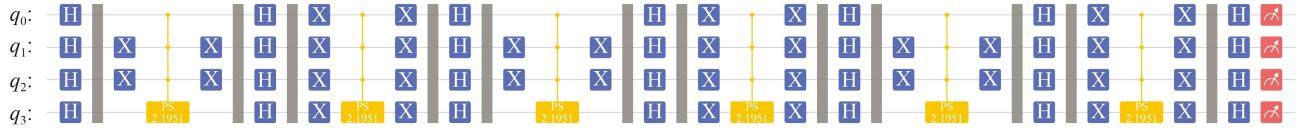


Fig. 18 Quantum circuit of the 4-qubit modified Grover's algorithm (target string $\tau = 1001$). The parameter in the circuit is $\phi = 2\arcsin\left(\sin\left(\frac{\pi}{4J+6}\right)/\sin\theta\right) = 2.195057699090115 \approx 2.1951$, where $J = \lfloor(\pi/2-\theta)/(2\theta)\rfloor$ and $\theta = \arcsin\left(\sqrt{1/2^4}\right)$.

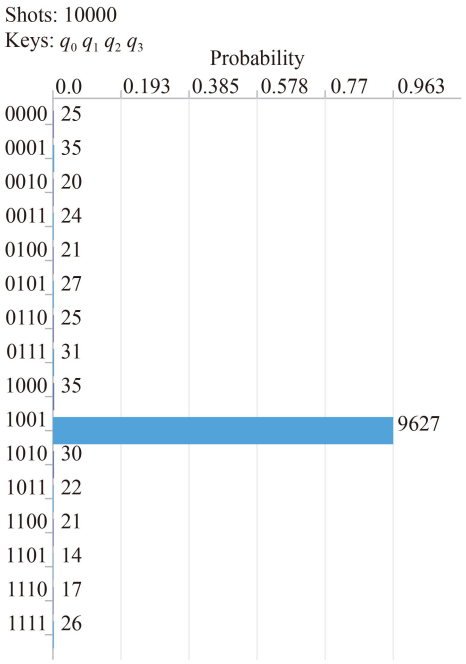


Fig. 19 The sampling results of the 4-qubit Grover's algorithm (target string $\tau = 1001$).

where $m_0 \in \{0,1\}^3$.

Similarly, we choose first 3 bits in x to divide f, then we can get $2^3 = 8$ subfunctions $f_{1,j} : \{0,1\}^2 \to \{0,1\}$:

$$f_{1,0}(m_1) = f(000m_1),\ f_{1,4}(m_1) = f(100m_1), \tag{101}$$

$$f_{1,1}(m_1) = f(001m_1),\ f_{1,5}(m_1) = f(101m_1), \tag{102}$$

$$f_{1,2}(m_1) = f(010m_1),\ f_{1,6}(m_1) = f(110m_1), \tag{103}$$

$$f_{1,3}(m_1) = f(011m_1),\ f_{1,7}(m_1) = f(111m_1), \tag{104}$$

where $m_1 \in \{0,1\}^2$ and $j \in \{0,1,\cdots,7\}$. Obviously, $f_{1,0}(m_1) = f_{1,1}(m_1) = f_{1,3}(m_1) = f_{1,4}(m_1) = f_{1,5}(m_1) = f_{1,6}(m_1) = f_{1,7}(m_1) \equiv 0$, and

$$f_{1,2}(m_1) = \begin{cases} 1, & m_1 = 01, \\ 0, & m_1 \neq 01, \end{cases} \tag{105}$$

where $m_1 \in \{0,1\}^2$ and 01 is the target substring.

Afterwards, we generate a new function $g_1 : \{0,1\}^2 \to \{0,1\}$ through above eight subfunctions,

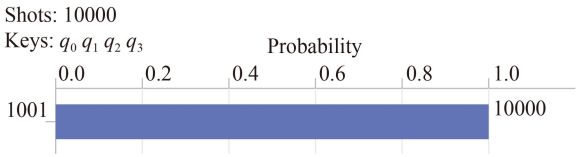


Fig. 20 The sampling results of the 4-qubit modified Grover's algorithm (target string $\tau = 1001$).

$$\begin{aligned} g_1(m_1) =& \mathrm{OR}\left(f_{1,0}(m_1), f_{1,1}(m_1), \cdots, f_{1,7}(m_1)\right) \\ =& \begin{cases} 1, & m_1 = 01, \\ 0, & m_1 \neq 01, \end{cases} \end{aligned} \tag{106}$$

where $m_1 \in \{0,1\}^2$.

So far, we have successfully obtained the subfunctions of two parts, $g_0(m_0)$ and $g_1(m_1)$. Next, we need to run the 3-qubit modified Grover's algorithm for the part of subfunction $g_0(m_0)$, and the 2-qubit Grover's algorithm for the part of subfunction $g_1(m_1)$. Thus, we can build the completed circuit (see Fig. 21). Once again since the reading order on MindQuantum is from right to left, which is the opposite of our reading order from left to right, the 2-qubit Grover's algorithm corresponding to $g_1(m_1)$ is located on the top of the circuit.

By sampling the circuit in Fig. 21 10 000 times, the sampling results can be found in Fig. 22.

The sampling results demonstrate that $\tau = 01001$ is obtained exactly after measurement. Besides, the total number of quantum gates is 53 and the circuit depth is 17.

If the original or modified Grover's algorithm is employed to search $\tau = 01001$, we can build the completed circuits (see Fig. 23 and Fig. 24 respectively). By sampling the circuits in Fig. 23 and Fig. 24 10 000 times respectively, the sampling results can be found in Fig. 25 and Fig. 26.

It can be found that the probability of Grover's algorithm acquiring target string $\tau = 01001$ is 0.9993, which is not exact, while the modified Grover's algorithm can accurately obtain $\tau = 01001$. The number of quantum gates required by two algorithms is 117, and the circuit depths are 33.

Through the above experiments, we can know the specific steps of implement n-qubit DEGA, where $n \in \{2,3,4,5\}$. Besides, we found that the modified Grover's algorithm and the DEGA can achieve exact search, while the Grover's algorithm can obtain target

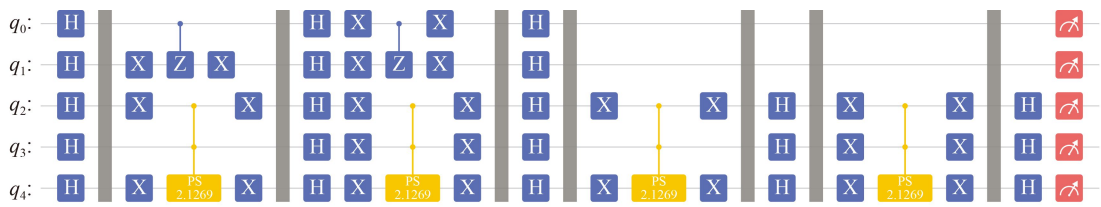


Fig. 21 Quantum circuit of the 5-qubit DEGA (target string $\tau = 01001$). The parameter in the circuit is $\phi = 2\arcsin\left(\sin\left(\frac{\pi}{4J+6}\right)/\sin\theta\right) = 2.1268800471555034 \approx 2.1269$, where $J = \lfloor(\pi/2-\theta)/(2\theta)\rfloor$ and $\theta = \arcsin\left(\sqrt{1/2^3}\right)$.

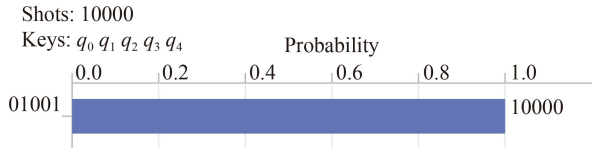


Fig. 22 The sampling results of the 4-qubit DEGA (target string $\tau = 01001$).

strings with high probability (see Fig. 27). In addition, the number of quantum gates and circuit depth required by the original and modified Grover's algorithms are the same, which will deepen as n increases. However, the DEGA requires fewer quantum gates (see Fig. 28) and shallower circuit depth (Fig. 29). The circuit depth of the DEGA only depends on the parity of n, and it does not deepen as n increases.

5.5 The depolarization channel

In the above experiments, we ignore the influence of noise, which means that we can achieve theoretically exact or high probability search. However, due to the existence of noise, there are certain errors in the operation of each type of quantum gate on a real quantum computer.

In this subsection, we consider the depolarization channel that is a essential type of quantum noise. After that, we will simulate the 5-qubit DEGA running in the depolarization channel and compare with the original and modified Grover's algorithms. Through such simulations, it shows the DEGA will be more resistant to the depolarization channel noise.

In the depolarization channel [35], a single qubit will be replaced by the completely mixed state $I/2$ with a probability of p and remain unaltered with a probability of $1-p$. It means the state of the quantum system has changed to

$$\varepsilon(\rho) = (1-p)\rho + p\left(\frac{I}{2}\right), \tag{107}$$

where ρ represents the initial quantum state. For arbitrary state ρ, we have

$$\frac{I}{2} = \frac{\rho + X\rho X + Y\rho Y + Z\rho Z}{4}, \tag{108}$$

where X, Y and Z are Pauli operators. Inserting Eq. (108) into Eq. (107) gives

$$\varepsilon(\rho) = \left(1 - \frac{3p}{4}\right)\rho + \frac{p}{4}\left(X\rho X + Y\rho Y + Z\rho Z\right). \tag{109}$$

For simplicity, the depolarization channel is sometimes written as

$$\varepsilon(\rho) = (1-p)\,\rho + \frac{p}{3}\left(X\rho X + Y\rho Y + Z\rho Z\right), \tag{110}$$

which indicates the state ρ will remain unaltered with the a probability of $1-p$, and the Pauli operators X, Y, Z will be applied with a probability of $p/3$ for each operators.

Furthermore, the depolarizing channel can be extended to a d-dimensional ($d > 2$) quantum system. Similarly, the depolarizing channel of a d-dimensional quantum system replaces the completely mixed state I/d with a probability of p, and remains unaltered with a probability of $1-p$. The corresponding quantum operation is

$$\varepsilon(\rho) = (1-p)\rho + p\left(\frac{I}{d}\right). \tag{111}$$

On MindQuantum, we simulate the quantum algorithms running in the depolarization channel by adding noise behind each quantum gate. That is, we can rebuild the quantum circuits of the original and modified Grover's algorithms and the DEGA in the depolarization channel (see Fig. 30 to Fig. 32). In the process of simulation, we set the noise parameter $p = 0.01$ and DC represents the depolarization channel.

By sampling the above three circuits 10 000 times,

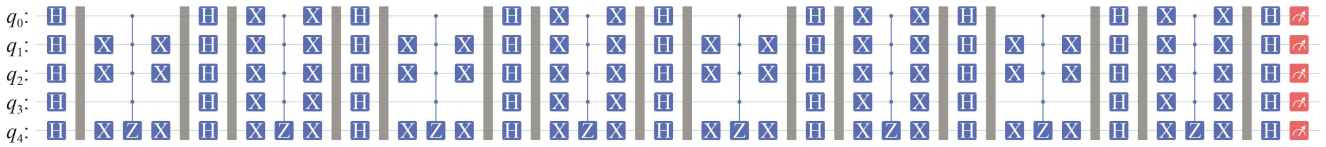


Fig. 23 Quantum circuit of the 5-qubit Grover's algorithm (target string $\tau = 01001$).

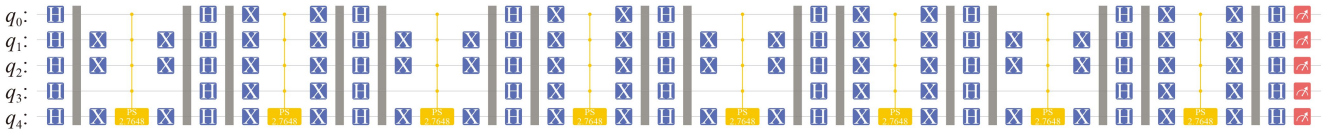


Fig. 24 Quantum circuit of the 5-qubit modified Grover's algorithm (target string $\tau = 01001$). The parameter in the circuit is $\phi = 2\arcsin\left(\sin\left(\frac{\pi}{4J+6}\right)/\sin\theta\right) = 2.764763603060391 \approx 2.7648$, where $J = \lfloor(\pi/2-\theta)/(2\theta)\rfloor$ and $\theta = \arcsin\left(\sqrt{1/2^5}\right)$.

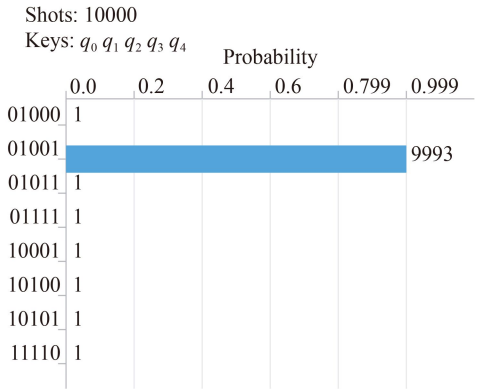


Fig. 25 The sampling results of the 5-qubit Grover's algorithm (target string $\tau = 01001$).

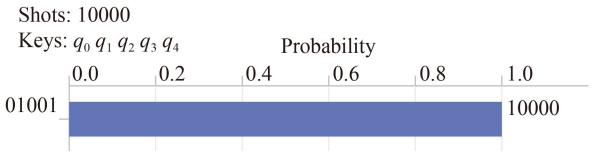


Fig. 26 The sampling results of the 5-qubit modified Grover's algorithm (target string $\tau = 01001$).

respectively. In order to better reproduce the experimental results, we set the random seeds of simulator and sampling both being 42. It is worth noting that modifying the random seed will alter the sampling outcome. In actuality, the random seed is a function to generate random number. The simulator will sample the quantum circuit according to the generated random number. As a result, various random numbers provide varied sampling outcomes in the simulator. Then, if we set the random seed to a fixed value (set the random seed being 42 in the full text), the experimental results in this paper can be reproduced when repeating the same experiment. At last, the sampling results can be found in Fig. 33 to Fig.

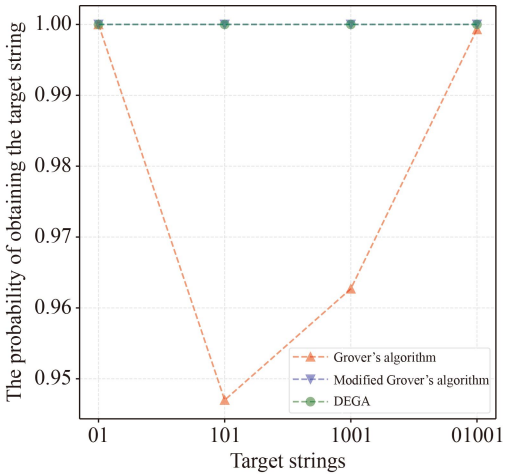


Fig. 27 The probability of obtaining the target string by different algorithms.

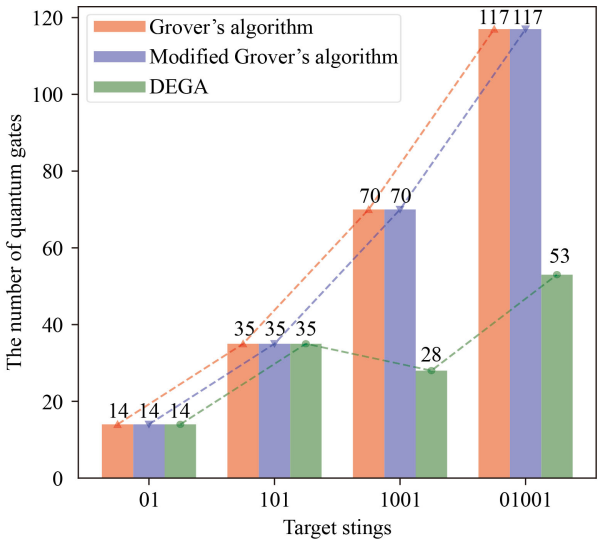


Fig. 28 The number of quantum gates required by different algorithms.

35, respectively.

It can be seen that due to the existence of noise, it will not achieve exact search. Compared with other states, $\tau = 01001$ can be obtained with a higher probability, which means that the target string can be successfully searched. The probability obtained by Grover's algorithm is 0.3495, the probability by the modified Grover's algorithm is 0.3501, and the probability by the DEGA is 0.6208. Obviously, the probability of the DEGA is higher.

Furthermore, we set the noise parameter p as $0, 0.01, \cdots, 0.09$, and count the probability of obtaining $\tau = 01001$ in each sampling by different algorithms. The statistical chart as shown in Fig. 36.

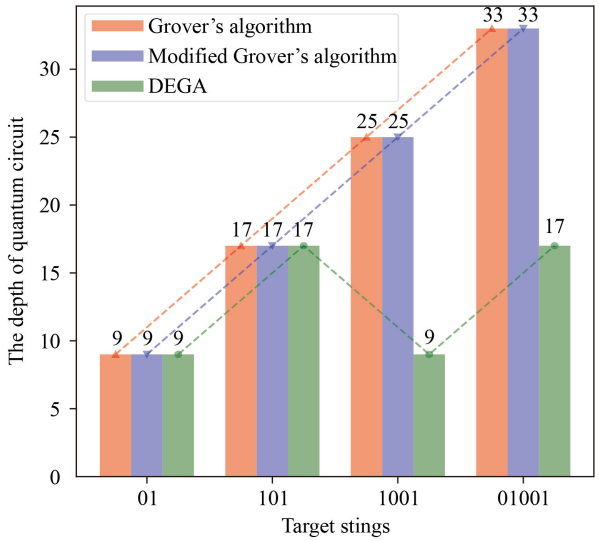


Fig. 29 The depth of quantum circuit of different algorithms.

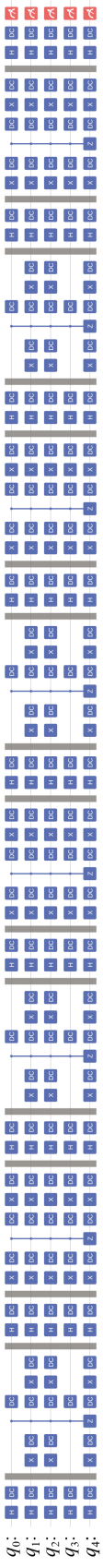


Fig. 30 Quantum circuit of the 5-qubit Grover's algorithm (target string $\tau = 01001$) in the depolarizing channel.

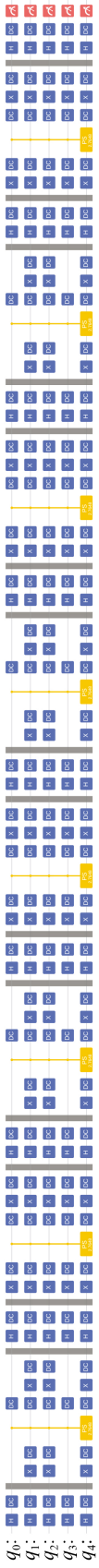


Fig. 31 Quantum circuit of the 5-qubit modified Grover's algorithm (target string $\tau = 01001$) in the depolarizing channel. The parameter in the circuit is $\phi = 2\arcsin\left(\sin\left(\frac{\pi}{4J+6}\right)/\sin\theta\right) = 2.7647636060391 \approx 2.7648$, where $J = \lfloor(\pi/2-\theta)/(2\theta)\rfloor$ and $\theta = \arcsin\left(\sqrt{1/2^5}\right)$.

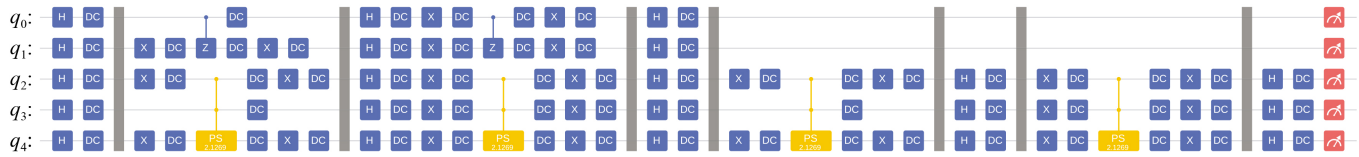


Fig. 32 Quantum circuit of the 5-qubit DEGA (target string $\tau = 01001$) in the depolarizing channel. The parameter in the circuit is $\phi = 2\arcsin\left(\sin\left(\frac{\pi}{4J+6}\right)/\sin\theta\right) = 2.1268800471555034 \approx 2.1269$, where $J = \lfloor(\pi/2-\theta)/(2\theta)\rfloor$ and $\theta = \arcsin\left(\sqrt{1/2^3}\right)$.

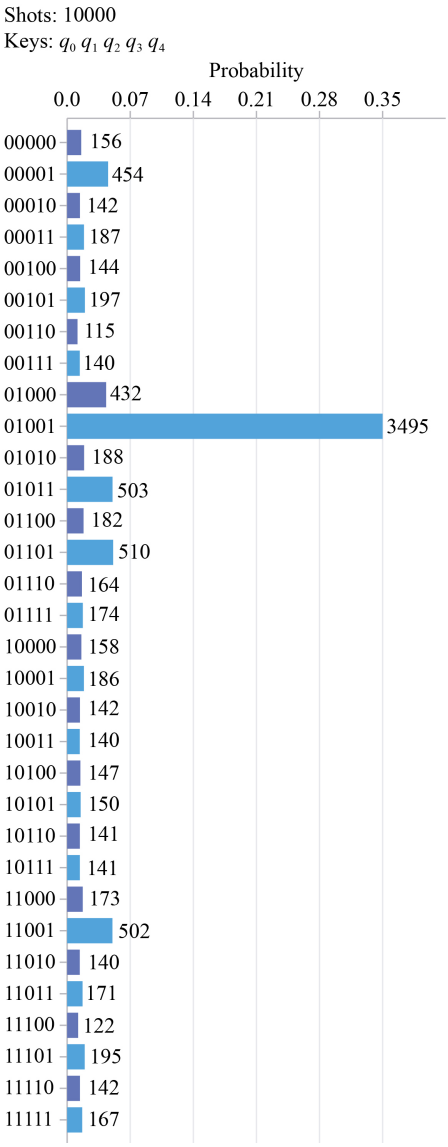


Fig. 33 The sampling results of the 5-qubit Grover's algorithm in the depolarizing channel. The noise parameter $p = 0.01$.

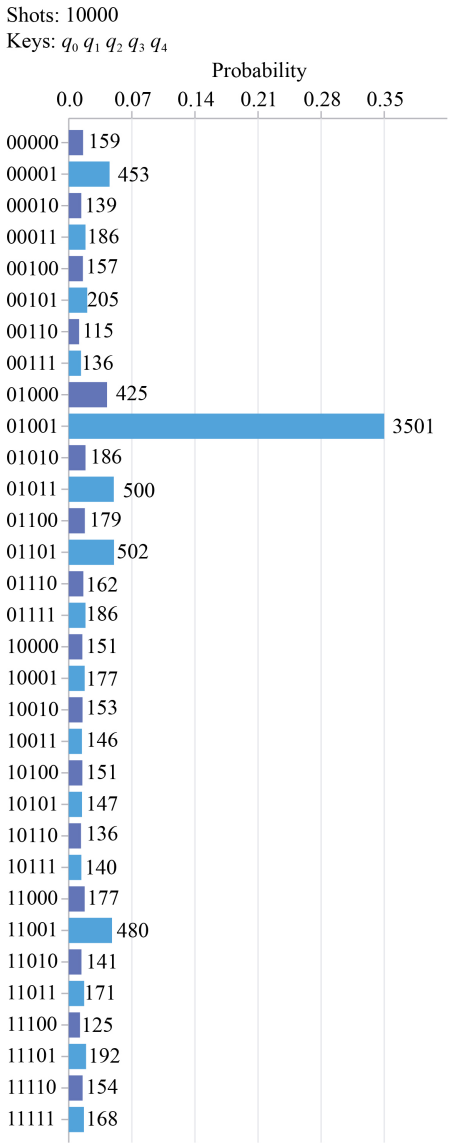


Fig. 34 The sampling results of the 5-qubit modified Grover's algorithm in the depolarizing channel. The noise parameter $p = 0.01$.

With the increase of noise parameter p, the probability of obtaining $\tau = 01001$ decreases. In addition, in the noisy environment, the probabilities of getting target string by the original and modified Grover's algorithms are close. At the same time, under the premise of the same noise parameter, the probability of acquiring $\tau = 01001$ by the DEGA is higher than those of the original and modified Grover's algorithms.

When $p = 0.07$, the sampling results can be found in Fig. 37 to Fig. 39, respectively. The probability of obtaining the target string $\tau = 01001$ are 0.0363, 0.0366 and 0.0899, respectively. The desired results were drown out by the channel noise in the original and modified Grover's algorithms. However, $\tau = 01001$ can still be

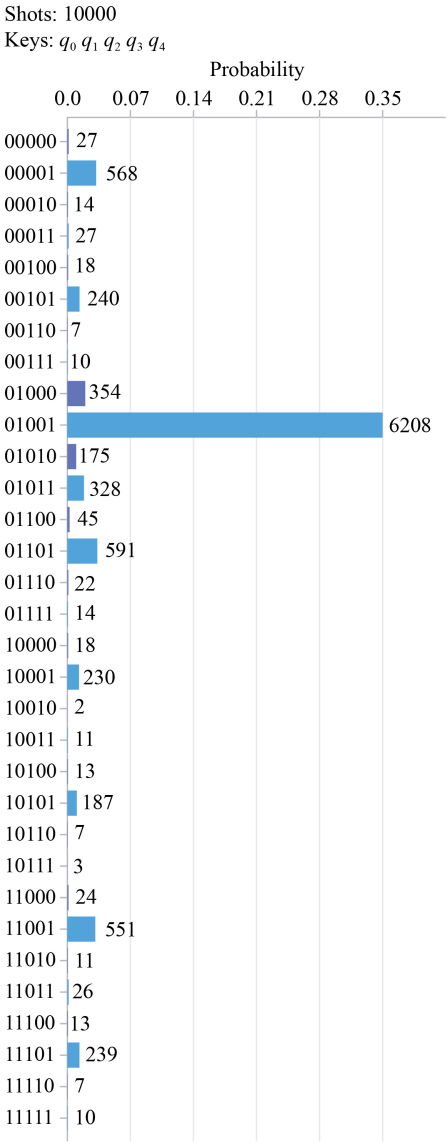


Fig. 35 The sampling results of the 5-qubit DEGA in the depolarizing channel. The noise parameter $p = 0.01$.

obtained with a higher probability than other states in the DEGA. Even if $p = 0.09$, the DEGA still works (see Fig. 40 to Fig. 42).

It can be found that although the number of qubits of the DEGA is the same as those of the original and modified Grover's algorithms, our circuit depth is shallower, so that our scheme is more resistant to the depolarization channel noise. In other words, it further illustrates that distributed quantum algorithm has the superiority of resisting noise.

6 Conclusion

Quantum computation is entering the noisy intermediate-scale quantum (NISQ) era. Currently, small-qubit quantum computers are much easier to implement than large-scale universal quantum computers. Hence, researchers are considering how several small-scale devices might collaborate to accomplish a task on a large scale.

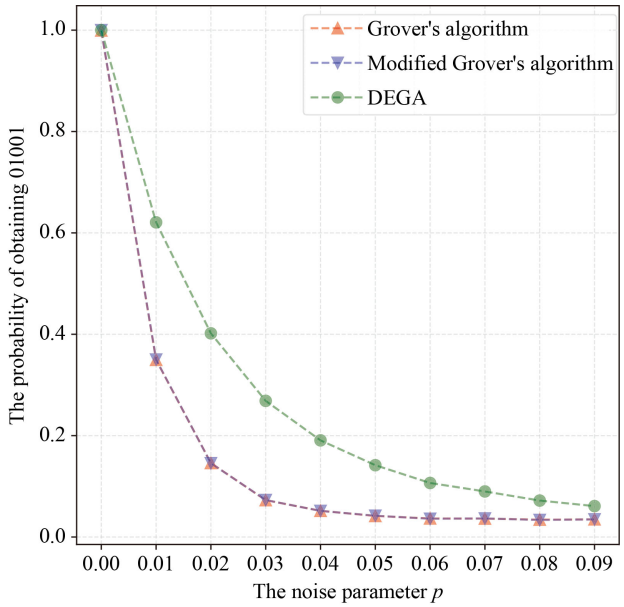


Fig. 36 The probability of obtaining $\tau = 01001$ by different algorithms.

In this paper, we have focused on the combination of distributed computing and the exact Grover's algorithm, and considered a search problem that includes only one target item in the unordered database. After that, we have proposed a distributed exact Grover's algorithm (DEGA), which decomposes the original search problem into $\lfloor n/2 \rfloor$ parts.

Specifically, (i) our algorithm is as exact as the modified version of Grover's algorithm by Long, which means the theoretical probability of finding the objective state is 100%; (ii) the actual depth of our circuit is $8(n \bmod 2) + 9$, which is less than the circuit depths of the original and modified Grover's algorithms, $1 + 8\left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor$ and $9 + 8\left\lfloor \frac{\pi}{4}\sqrt{2^n} - \frac{1}{2} \right\rfloor$, respectively. It only depends on the parity of n, and it does not deepen as n increases; (iii) additional auxiliary qubits will not be employed. In our framework, the total number of qubits is n rather than $2^k(n-k)$, where 2^k represents the number of computing nodes in the previous distributed method.

Eventually, we have provided particular situations of the DEGA on MindQuantum (a quantum software). It has further demonstrated the correctness of the DEGA and explicated the specific steps of implement n-qubit DEGA, where $n \in \{2, 3, 4, 5\}$. Through the experiments, we have found that the modified Grover's algorithm and the DEGA can achieve exact search, while Grover's algorithm can obtain target strings with high probability. In addition, the number of quantum gates and circuit depth required by the original and modified Grover's algorithms are the same, which will deepen as n

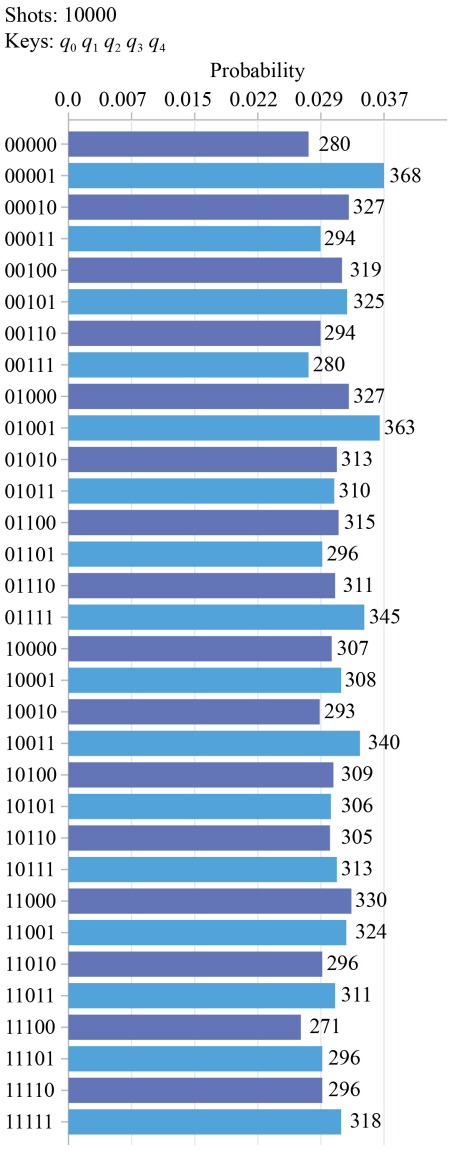


Fig. 37 The sampling results of the 5-qubit Grover's algorithm in the depolarizing channel. The noise parameter $p = 0.07$.

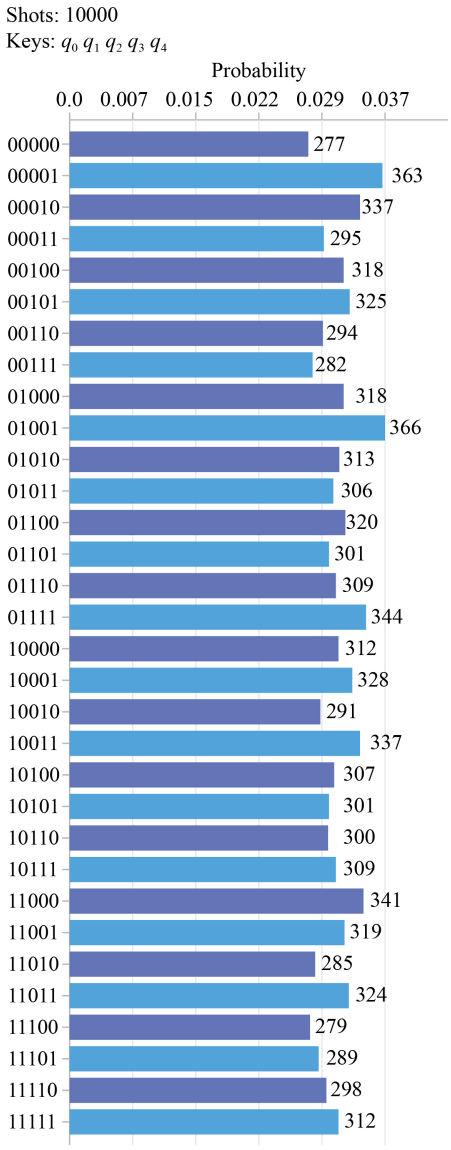


Fig. 38 The sampling results of the 5-qubit modified Grover's algorithm in the depolarizing channel. The noise parameter $p = 0.07$.

increases. However, the DEGA requires fewer quantum gates and shallower circuit depth. The circuit depth of the DEGA only depends on the parity of n, and it does not deepen as n increases.

After that, we have simulated a 5-qubit DEGA running in the depolarization channel and compared with the original and modified Grover's algorithms. Through such simulations, it shows the DEGA will be more resistant to the depolarization channel noise. In other words, it further illustrates that distributed quantum algorithm has the superiority of resisting noise.

However, we have only designed a distributed exact quantum algorithm for the case of unique target in search problem, so it is still open that designing a distributed exact quantum algorithm solves the search problem with the case of multiple targets.

Declarations The authors declare that they have no competing interests and there are no conflicts.

Data availability statement No data associated in the manuscript.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (Nos. 61572532 and 61876195) and the Natural Science Foundation of Guangdong Province of China (No. 2017B030311011).

Appendix A: The equivalence of Eq. (15) and Eq. (16)

Let

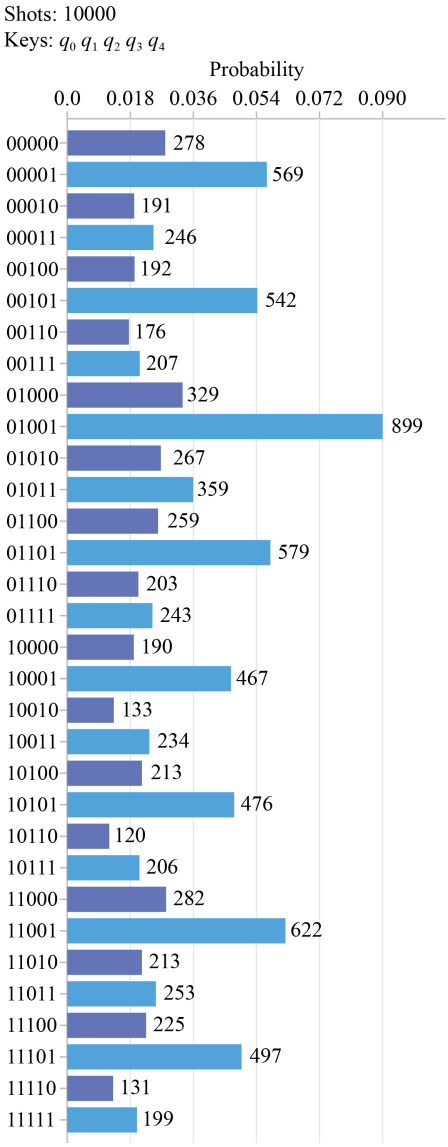


Fig. 39 The sampling results of the 5-qubit DEGA in the depolarizing channel. The noise parameter $p = 0.07$.

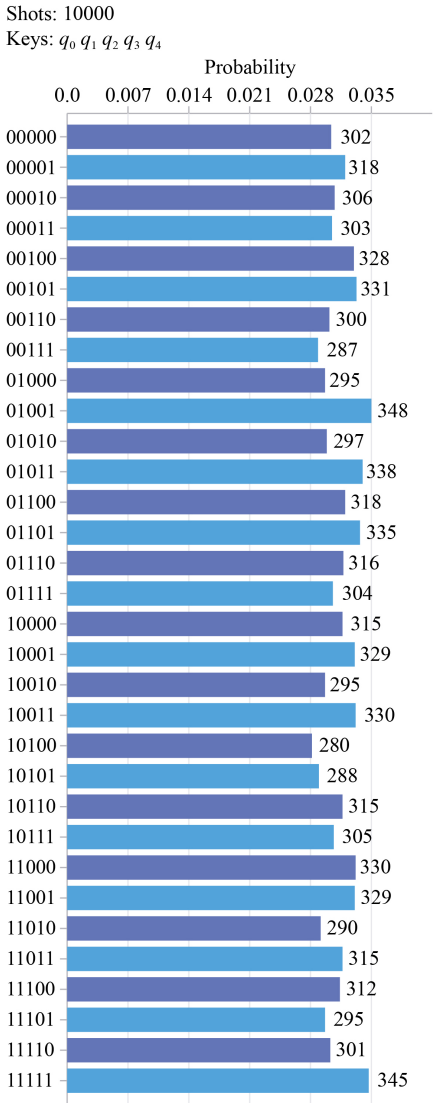


Fig. 40 The sampling results of the 5-qubit Grover's algorithm in the depolarizing channel. The noise parameter $p = 0.09$.

$$\begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix} = -\mathrm{e}^{\mathrm{i}\phi}\left[\cos\left(\frac{\alpha}{2}\right) I + \mathrm{i}\sin\left(\frac{\alpha}{2}\right)(n_x X + n_y Y + n_z Z)\right]. \tag{112}$$

In order to prove the equivalence of Eq. (15) and Eq. (16), it is equivalent to proving

$$\begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix} = \begin{pmatrix} -\mathrm{e}^{\mathrm{i}\phi}(1+(\mathrm{e}^{\mathrm{i}\phi}-1)\sin^2\theta) & -\mathrm{e}^{\mathrm{i}\phi}(1-\mathrm{e}^{-\mathrm{i}\phi})\sin\theta\cos\theta \\ -\mathrm{e}^{\mathrm{i}\phi}(\mathrm{e}^{\mathrm{i}\phi}-1)\sin\theta\cos\theta & -\mathrm{e}^{\mathrm{i}\phi}(1-(1-\mathrm{e}^{-\mathrm{i}\phi})\sin^2\theta) \end{pmatrix}. \tag{113}$$

Inserting Pauli gates to the right of Eq. (112), we have

$$\begin{aligned} &-\mathrm{e}^{\mathrm{i}\phi}\left[\cos\left(\frac{\alpha}{2}\right) I + \mathrm{i}\sin\left(\frac{\alpha}{2}\right)(n_x X + n_y Y + n_z Z)\right] \\ =&-\mathrm{e}^{\mathrm{i}\phi}\left[\cos\left(\frac{\alpha}{2}\right)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \mathrm{i}\sin\left(\frac{\alpha}{2}\right)\left(n_x\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + n_y\begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix} + n_z\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\right)\right] \\ =&-\mathrm{e}^{\mathrm{i}\phi}\begin{pmatrix} \cos\left(\frac{\alpha}{2}\right) + \mathrm{i}\sin\left(\frac{\alpha}{2}\right) n_z & \mathrm{i}\sin\left(\frac{\alpha}{2}\right) n_x + \sin\left(\frac{\alpha}{2}\right) n_y \\ \mathrm{i}\sin\left(\frac{\alpha}{2}\right) n_x - \sin\left(\frac{\alpha}{2}\right) n_y & \cos\left(\frac{\alpha}{2}\right) - \mathrm{i}\sin\left(\frac{\alpha}{2}\right) n_z \end{pmatrix} = \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix}, \end{aligned} \tag{114}$$

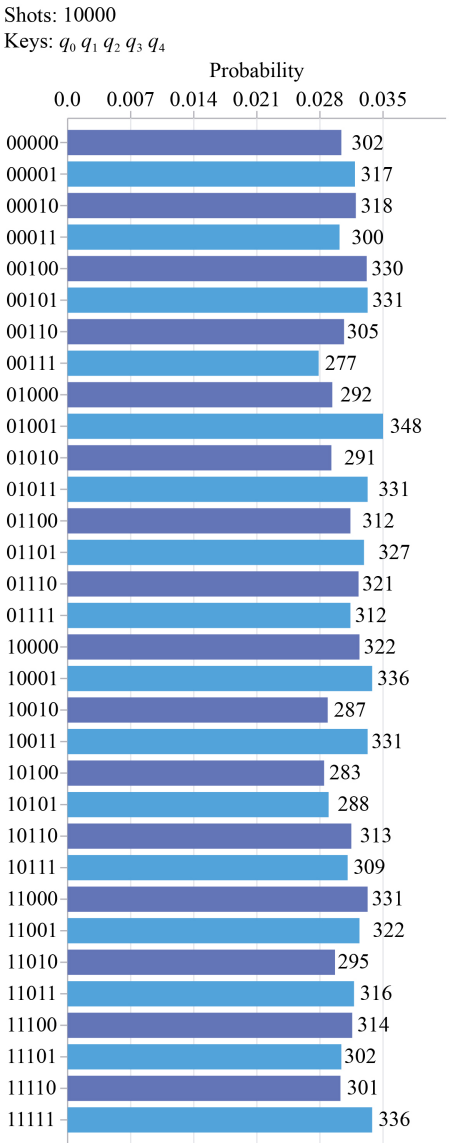


Fig. 41 The sampling results of the 5-qubit modified Grover's algorithm in the depolarizing channel. The noise parameter $p = 0.09$.

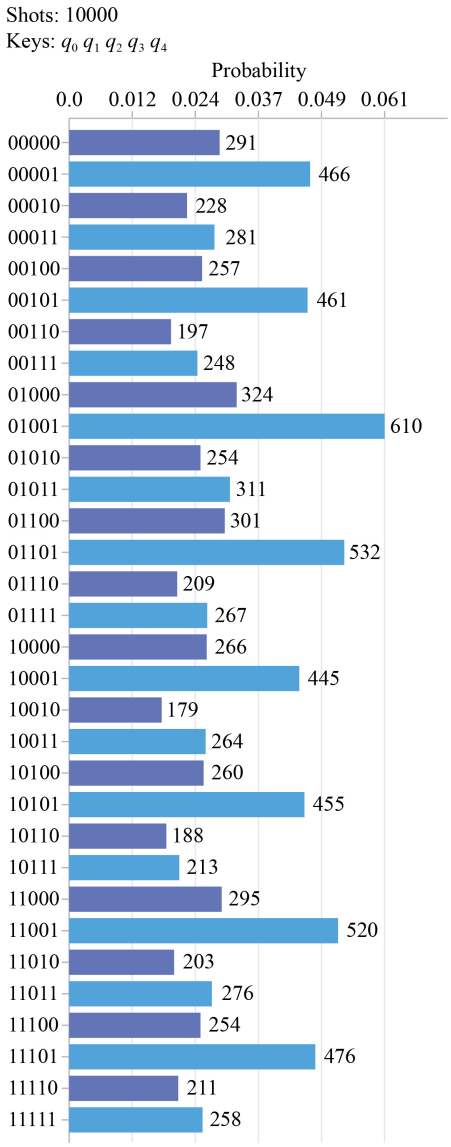


Fig. 42 The sampling results of the 5-qubit DEGA in the depolarizing channel. The noise parameter $p = 0.09$.

where

$$n_x = \frac{\cos\theta}{\cos\beta}\cos\left(\frac{\phi}{2}\right), n_y = \frac{\cos\theta}{\cos\beta}\sin\left(\frac{\phi}{2}\right), n_z = \frac{\cos\theta}{\cos\beta}\cos\left(\frac{\phi}{2}\right)\tan\theta. \tag{115}$$

Besides, we know that

$$\alpha = 4\beta,\ \sin\beta = \sin\left(\frac{\phi}{2}\right)\sin\theta. \tag{116}$$

Furthermore, we have

$$\begin{aligned} L_{11} &= -\mathrm{e}^{\mathrm{i}\phi}\left(\cos\left(\frac{\alpha}{2}\right) + \mathrm{i}\sin\left(\frac{\alpha}{2}\right)n_z\right) \\ &= -\mathrm{e}^{\mathrm{i}\phi}\left(\cos\left(\frac{\alpha}{2}\right) + \mathrm{i}\sin\left(\frac{\alpha}{2}\right)\frac{\cos\theta}{\cos\beta}\cos\left(\frac{\phi}{2}\right)\tan\theta\right) \end{aligned}$$

$$
\begin{aligned}
&= -\mathrm{e}^{\mathrm{i}\phi}\left(\cos(2\beta)+\mathrm{i}\sin(2\beta)\frac{\cancel{\cos\theta}}{\cos\beta}\cos\left(\frac{\phi}{2}\right)\frac{\sin\theta}{\cancel{\cos\theta}}\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(1-2\sin^2\beta+2\mathrm{i}\sin\beta\cancel{\cos\beta}\frac{1}{\cancel{\cos\beta}}\cos\left(\frac{\phi}{2}\right)\sin\theta\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(1-2\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right)^2+2\mathrm{i}\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right)\cos\left(\frac{\phi}{2}\right)\sin\theta\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(1-2\sin^2\left(\frac{\phi}{2}\right)\sin^2\theta+\mathrm{i}\sin\phi\sin^2\theta\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(1+\left(1-2\sin^2\left(\frac{\phi}{2}\right)+\mathrm{i}\sin\phi-1\right)\sin^2\theta\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(1+(\cos\phi+\mathrm{i}\sin\phi-1)\sin^2\theta\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}(1+(\mathrm{e}^{\mathrm{i}\phi}-1)\sin^2\theta),
\end{aligned}
\tag{117}
$$

$$
\begin{aligned}
L_{12} &= -\mathrm{e}^{\mathrm{i}\phi}\left(\mathrm{i}\sin\left(\frac{\alpha}{2}\right)n_x+\sin\left(\frac{\alpha}{2}\right)n_y\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(\mathrm{i}\sin\left(\frac{\alpha}{2}\right)\frac{\cos\theta}{\cos\beta}\cos\left(\frac{\phi}{2}\right)+\sin\left(\frac{\alpha}{2}\right)\frac{\cos\theta}{\cos\beta}\sin\left(\frac{\phi}{2}\right)\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(\mathrm{i}\sin(2\beta)\frac{\cos\theta}{\cos\beta}\cos\left(\frac{\phi}{2}\right)+\sin(2\beta)\frac{\cos\theta}{\cos\beta}\sin\left(\frac{\phi}{2}\right)\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(2\mathrm{i}\sin\beta\cancel{\cos\beta}\frac{\cos\theta}{\cancel{\cos\beta}}\cos\left(\frac{\phi}{2}\right)+2\sin\beta\cancel{\cos\beta}\frac{\cos\theta}{\cancel{\cos\beta}}\sin\left(\frac{\phi}{2}\right)\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(2\mathrm{i}\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right)\cos\theta\cos\left(\frac{\phi}{2}\right)+2\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right)\cos\theta\sin\left(\frac{\phi}{2}\right)\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(\mathrm{i}\sin\phi+1+2\sin^2\left(\frac{\phi}{2}\right)-1\right)\sin\theta\cos\theta\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(1-(\cos\phi-\mathrm{i}\sin\phi)\right)\sin\theta\cos\theta\\
&= -\mathrm{e}^{\mathrm{i}\phi}(1-\mathrm{e}^{-\mathrm{i}\phi})\sin\theta\cos\theta,
\end{aligned}
\tag{118}
$$

$$
\begin{aligned}
L_{21} &= -\mathrm{e}^{\mathrm{i}\phi}\left(\mathrm{i}\sin\left(\frac{\alpha}{2}\right)n_x-\sin\left(\frac{\alpha}{2}\right)n_y\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(\mathrm{i}\sin\left(\frac{\alpha}{2}\right)\frac{\cos\theta}{\cos\beta}\cos\left(\frac{\phi}{2}\right)-\sin\left(\frac{\alpha}{2}\right)\frac{\cos\theta}{\cos\beta}\sin\left(\frac{\phi}{2}\right)\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(\mathrm{i}\sin(2\beta)\frac{\cos\theta}{\cos\beta}\cos\left(\frac{\phi}{2}\right)-\sin(2\beta)\frac{\cos\theta}{\cos\beta}\sin\left(\frac{\phi}{2}\right)\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(2\mathrm{i}\sin\beta\cancel{\cos\beta}\frac{\cos\theta}{\cancel{\cos\beta}}\cos\left(\frac{\phi}{2}\right)-2\sin\beta\cancel{\cos\beta}\frac{\cos\theta}{\cancel{\cos\beta}}\sin\left(\frac{\phi}{2}\right)\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(2\mathrm{i}\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right)\cos\theta\cos\left(\frac{\phi}{2}\right)-2\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right)\cos\theta\sin\left(\frac{\phi}{2}\right)\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(\mathrm{i}\sin\phi+1-2\sin^2\left(\frac{\phi}{2}\right)-1\right)\sin\theta\cos\theta\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left((\cos\phi+\mathrm{i}\sin\phi)-1\right)\sin\theta\cos\theta\\
&= -\mathrm{e}^{\mathrm{i}\phi}(\mathrm{e}^{\mathrm{i}\phi}-1)\sin\theta\cos\theta,
\end{aligned}
\tag{119}
$$

$$
\begin{aligned}
L_{22} &= -\mathrm{e}^{\mathrm{i}\phi}\left(\cos\left(\frac{\alpha}{2}\right)-\mathrm{i}\sin\left(\frac{\alpha}{2}\right)n_z\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(\cos\left(\frac{\alpha}{2}\right)-\mathrm{i}\sin\left(\frac{\alpha}{2}\right)\frac{\cos\theta}{\cos\beta}\cos\left(\frac{\phi}{2}\right)\tan\theta\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(\cos(2\beta)-\mathrm{i}\sin(2\beta)\frac{\cancel{\cos\theta}}{\cos\beta}\cos\left(\frac{\phi}{2}\right)\frac{\sin\theta}{\cancel{\cos\theta}}\right)
\end{aligned}
$$

$$
\begin{aligned}
&= -\mathrm{e}^{\mathrm{i}\phi}\left(1-2\sin^2\beta-2\mathrm{i}\sin\beta\cancel{\cos\beta}\frac{1}{\cancel{\cos\beta}}\cos\left(\frac{\phi}{2}\right)\sin\theta\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(1-2\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right)^2-2\mathrm{i}\left(\sin\left(\frac{\phi}{2}\right)\sin\theta\right)\cos\left(\frac{\phi}{2}\right)\sin\theta\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(1-2\sin^2\left(\frac{\phi}{2}\right)\sin^2\theta-\mathrm{i}\sin\phi\sin^2\theta\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(1-\left(2\sin^2\left(\frac{\phi}{2}\right)-1+\mathrm{i}\sin\phi+1\right)\sin^2\theta\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}\left(1-(-\cos\phi+\mathrm{i}\sin\phi+1)\sin^2\theta\right)\\
&= -\mathrm{e}^{\mathrm{i}\phi}(1-(1-\mathrm{e}^{-\mathrm{i}\phi})\sin^2\theta). \qquad (120)
\end{aligned}
$$

To sum up, it can be seen from Eq. (117) to Eq. (120) that Eq. (15) and Eq. (16) are equivalent.

References

1. P. Benioff, The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines, *J. Stat. Phys.* 22(5), 563 (1980)
2. P. Benioff, Quantum mechanical Hamiltonian models of Turing machines, *J. Stat. Phys.* 29(3), 515 (1982)
3. D. Deutsch, Quantum theory, the Church–Turing principle and the universal quantum computer, *Proc. R. Soc. Lond. A* 400(1818), 97 (1985)
4. D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation, *Proc. R. Soc. Lond. A* 439(1907), 553 (1992)
5. E. Bernstein and U. Vazirani, Quantum complexity theory, in: Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing (STOC'93), 1993, pp 11–20
6. D. R. Simon, On the power of quantum computation, *SIAM J. Comput.* 26(5), 1474 (1997)
7. P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in: Proceedings 35th Annual Symposium on Foundations of Computer Science, IEEE, 1994, pp 124–134
8. L. K. Grover, Quantum mechanics helps in searching for a needle in a haystack, *Phys. Rev. Lett.* 79(2), 325 (1997)
9. A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* 103(15), 150502 (2009)
10. D. Christoph and H. Peter, A quantum algorithm for finding the minimum, arXiv: quant-ph/9607014v2 (1996)
11. H. Ramesh and V. Vinay, String matching in $\tilde{O}(\sqrt{n}+\sqrt{m})$ quantum time, *J. Discrete Algorithms* 1(1), 103 (2003)
12. A. Ambainis, K. Balodis, J. Iraids, et al., Quantum speedups for exponential-time dynamic programming algorithms, in: Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms. San Diego: SIAM, 2019, pp 1783–1793
13. A. Andris and L. Nikita, Quantum algorithms for computational geometry problems, in: 15th Conference on the Theory of Quantum Computation, Communication and Cryptography, 2020
14. G. Brassard, P. Hoyer, M. Mosca, et al., Quantum amplitude amplification and estimation, *Contemp. Math.* 305, 53 (2002)
15. Z. J. Diao, Exactness of the original Grover search algorithm, *Phys. Rev. A* 82(4), 044301 (2010)
16. G. L. Long, Grover algorithm with zero theoretical failure rate, *Phys. Rev. A* 64(2), 022307 (2001)
17. J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* 2, 79 (2018)
18. J. I. Cirac and P. Zoller, Quantum computations with cold trapped ions, *Phys. Rev. Lett.* 74(20), 4091 (1995)
19. C. Y. Lu, X. Q. Zhou, O. Guhne, W. B. Gao, J. Zhang, Z. S. Yuan, A. Goebel, T. Yang, and J. W. Pan, Experimental entanglement of six photons in graph states, *Nat. Phys.* 3(2), 91 (2007)
20. Y. Makhlin, G. Schön, and A. Shnirman, Quantum-state engineering with Josephson-junction devices, *Rev. Mod. Phys.* 73(2), 357 (2001)
21. J. Berezovsky, M. H. Mikkelsen, N. G. Stoltz, L. A. Coldren, and D. D. Awschalom, Picosecond coherent optical manipulation of a single electron spin in a quantum dot, *Science* 320(5874), 349 (2008)
22. R. Hanson and D. D. Awschalom, Coherent manipulation of single spins in semiconductors, *Nature* 453(7198), 1043 (2008)
23. H. Buhrman and H. Röhrig, Distributed quantum computing, in: Mathematical Foundations of Computer Science 2003: 28th International Symposium, Berlin Heidelberg, Springer, 2003, pp 1–20
24. A. Yimsiriwattan and Jr Lomonaco, Distributed quantum computing: A distributed Shor algorithm, *Quantum Inf. Comput. II.* 5436, 360 (2004)
25. R. Beals, S. Brierley, O. Gray, et al., Efficient distributed quantum computing, *Proc. R. Soc. A* 469(2153), 0120686 (2013)
26. K. Li, D. W. Qiu, L. Li, S. Zheng, and Z. Rong, Application of distributed semiquantum computing model in phase estimation, *Inf. Process. Lett.* 120, 23 (2017)
27. J. Avron, O. Casper, and I. Rozen, Quantum advantage and noise reduction in distributed quantum computing, *Phys. Rev. A* 104(5), 052404 (2021)
28. D. W. Qiu, L. Luo, and L. G. Xiao, Distributed

Grover's algorithm, arXiv: 2204.10487v3 (2022)
29. J. W. Tan, L. G. Xiao, D. W. Qiu, L. Luo, and P. Mateus, Distributed quantum algorithm for Simon's problem, *Phys. Rev. A* 106(3), 032417 (2022)
30. L. G. Xiao, D. W. Qiu, L. Luo, and P. Mateus, Distributed Shor's algorithm, *Quantum Inf. Comput.* 23(1&2), 27 (2023)
31. L. G. Xiao, D. W. Qiu, L. Luo, et al., Distributed quantum–classical hybrid Shor's algorithm, arXiv: 2304.12100 (2023)
32. X. Zhou, D. W. Qiu, and L. Luo, Distributed exact quantum algorithms for Bernstein–Vazirani and search problems, arXiv: 2303.10670 (2023)
33. H. Li, D. W. Qiu, and L. Luo, Distributed exact quantum algorithms for Deutsch–Jozsa problem, arXiv: 2303.10663 (2023)
34. MindQuantum, URL: gitee.com/mindspore/mindquantum, 2021
35. M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge: Cambridge University Press, 2002