

# An Optimized Quantum Minimum Searching Algorithm with Sure-success Probability and Its Experiment Simulation with Cirq

Wenjie Liu<sup>1,2</sup>  · Qingshan Wu<sup>2</sup> · Jiahao Shen<sup>2</sup> · Jiaojiao Zhao<sup>2</sup> · Mohammed Zidan<sup>3</sup> · Lian Tong<sup>4</sup>

Received: date / Accepted: date

**Abstract** Finding a minimum is an essential part of mathematical models, and it plays an important role in some optimization problems. Durr and Hoyer proposed a quantum searching algorithm (DHA), with a certain probability of success, to achieve quadratic speed than classical ones. In this paper, we propose an optimized quantum minimum searching algorithm with sure-success probability, which utilizes Grover-Long searching to implement the optimal exact searching, and the dynamic strategy to reduce the iterations of our algorithm. Besides, we optimize the oracle circuit to reduce the number of gates by the simplified rules. The performance evaluation including the theoretical success rate and computational complexity shows that our algorithm has higher accuracy and efficiency than DHA algorithm. Finally, a simulation experiment based on Cirq is performed to verify its feasibility.

**Keywords** Quantum minimum searching algorithm · Sure-success probability · Grover-Long algorithm · Dynamic strategy · Circuit optimization · Cirq

## 1 Introduction

In recent years, the development of big data has made it urgent to deal with more data with higher speed and better efficiency. Therefore, some researchers have begun to try to use genetic algorithms(GA)[1], particle swarm optimization(PSO)[2], and some other state-of-art searching algorithms[3,4,

---

 wenjiel@163.com

<sup>1</sup> Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing 210044, China

<sup>2</sup> School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

<sup>3</sup> Center for Photonics and Smart Materials (CPSM), Zewail City of Science and Technology, October Gardens, 6th of October City, Giza 12578, Egypt

<sup>4</sup> School of Information Engineering, Jiangsu Maritime Institute, Nanjing 211100, China

5] to improve search efficiency. Using the properties of quantum mechanics, researchers have discovered some quantum algorithms to accelerate a series of algorithms in quantum computers [6, 7]. Besides, many researchers try to apply quantum mechanics in other fields, such as quantum key agreement (QKA) [8, 9], quantum secret sharing (QSS) [10, 11], blind quantum computation (BQC) [12, 13], quantum private query (QPQ) [14, 15], and even quantum machine learning (QML) [16, 17]. Shor's factoring algorithm [18] is a well-known example of a quantum algorithm outperforming the best known classical algorithm. This algorithm can effectively find discrete logarithms and factor integers on a quantum computer. In order to speed up the search problem, Grover proposed a quantum search algorithm [19] in 1996. This algorithm can solve the searching problem by using approximately  $\sqrt{N}$  operations rather than approximately  $N$  operations in classical algorithm. Later, the database search algorithm gradually attracted wide attention of many scholars. In 2010, Diao pointed out that only if the ratio of the solution  $M$  to the database size  $N$  is  $1/4$ , a strict and accurate search can be performed [20]. Especially, the highest failure rate is 50% when  $M/N = 1/2$ . To improve the efficiency of the Grover algorithm, researchers have explored various generalized and modified versions of the Grover algorithm, including phase matching methods [21], for an arbitrary initial amplitude distribution [22], recursion equations method [23], Grover-Long algorithm [24, 25] and fixed-point [26]. Among them, Grover-Long algorithm has one adjustable phase that finds the target with zero failure rate for any database and with exactly the same number of iterations as Grover algorithm.

With the development of big data, finding a minimum or maximum is a significant issue in many fields. Classically, approximately  $N$  operations are required for searching the maximum or minimum problems. But its quantum counterpart proposed by Durr and Hoyer [27] achieves quadratic speedup, which was based on the quantum exponential searching algorithm [28, 29, 30, 31]. When the number of solutions is unknown, the quantum exponential searching algorithm reduces its failure rate at the expense of repeatedly performing Grover's algorithm with different number of iterations. However, Grover algorithm is not a sure-success algorithm. Besides, the operation of marking state in the repetition approach also takes time. To solve these problems, we propose an optimized quantum minimum searching algorithm (OQMSA). The main contributions are as follows:

1. We utilize Grover-Long searching to implement the optimal exact searching, and then propose a sure-success quantum minimum searching algorithm.
2. In order to improve the efficiency of our algorithm, a dynamic strategy is proposed to reduce the iterations. The ratio of the solutions to the size of dataset is different, then we use different minimum searching methods.
3. In terms of quantum circuit implementation, we propose two simplified rules for the oracle operation, thereby reducing the number of quantum gates.

The remainder of this paper is organized as follows. In Section 2, we review DHA algorithm. In Section 3, we present OQMSA based on Grover-Long algorithm and the general quantum circuits of key steps. In Section 4, we analyze

the success rate and complexity of OQMSA. In Section 5, an experiment based on Cirq framework to solve a specific problem is presented, which shows that OQMSA is indeed more efficient. In Section 6, we draw a conclusion and look forward to its future applications.

## 2 Review of DHA Algorithm

DHA algorithm [27] can solve the problem of finding the minimum of unsorted database. It aims to find the index of a smaller item than the value determined by a particular threshold index. Then the result is chosen as the new threshold. This process repeats for times to increase the probability of finding the index of the minimum. If there are  $t \geq 1$  marked table entries, the quantum exponential searching algorithm will return one of them with equal probability after an expected number of  $O(\sqrt{N/t})$  iterations. If no entry is marked, it will run forever. The steps of DHA algorithm are as follows:

Step 1: Choose threshold index  $0 \leq y \leq N - 1$  uniformly at random.

Step 2: Repeat the following and interrupt it when the total running time is more than  $22.5\sqrt{N} + 1.4\lg^2 N$ .

(a) Prepare the initial state  $\sum_j \frac{1}{\sqrt{N}} |j\rangle |y\rangle$ . Mark every item  $j$  when  $D[j] < D[y]$ .

(b) Apply the quantum exponential searching algorithm on the initial state.

(c) Measure the first register: output  $y'$  if  $D[y'] < D[y]$ , then set threshold index  $y$  to  $y'$ .

Step 3: Return the measure value  $y$ .

DHA algorithm provides a simple quantum algorithm which solves the problem using  $O\sqrt{N}$  probes. The main subroutine is the quantum exponential searching algorithm, which is a generalization of Grover algorithm. However, we find the uncontrollability of deflection angle in origin Grover algorithm, which result in the low success rate in DHA algorithm. In the meantime, the construction of the oracle is so complicated that the complexity raises a lot. To improve the success rate and decrease the complexity, we propose an optimized quantum minimum searching algorithm.

## 3 An optimized quantum minimum searching algorithm with sure-success probability

### 3.1 The proposed OQMSA algorithm

In order to conduct the exact searching i.e., obtain the minimum with a success rate close to 1, we propose an optimized quantum minimum searching algorithm. Suppose that  $D$  is an unsorted database with  $N$  items,  $D = \{d_j | 0 \leq$

**Algorithm 1** The Proposed OQMSA Algorithm for finding the minimum

---

**Input:** Database  $D = \{d_j | 0 \leq j < N\}$ , a random item  $d' \in D$ .  
**Output:** The minimum item  $d_{min}$ .

- 1: **for**  $i = 0$  **to**  $\lceil \text{Log}N \rceil$  **do**
- 2:      $t_{max} = \lceil (\pi/2 - \arcsin(1/\sqrt{N})) / \arcsin(1/\sqrt{N}) \rceil$  ;
- 3:      $t = 1$ ,  $\lambda = \frac{6}{5}$ , and  $r \rightarrow +\infty$ ;
- 4:     **while**  $(t \leq t_{max} \ \&\& \ r > d')$  **do**
- 5:         Prepare the initial state  $|\varphi\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |d_j\rangle$  according to  $N$  data items; **iiii**
- 6:         **if**  $\frac{M}{N} > \frac{1}{9}$  **then**
- 7:              $t' = \text{randint}(0, \lceil t \rceil)$ ;
- 8:             Apply Grover-Long searching on  $|\varphi\rangle$  with  $t'$  iterations  $\rightarrow |\varphi'\rangle$ ;
- 9:              $t = t * \lambda$ ;
- 10:         **else**
- 11:             Apply Grover-Long searching on  $|\varphi\rangle$  with  $t_{max}$  iterations  $\rightarrow |\varphi'\rangle$ ; **iiii**
- 12:         **end**
- 13:         Measure  $|\varphi'\rangle$  and assign the measurement result to  $r$ ;
- 14:         **end**
- 15:         **if**  $r < d'$  **then** **iiiiiii**
- 16:              $d' = r$ ;
- 17:              $i = 0$ ; **iiii**
- 18:         **end**
- 19:     **end**
- 20: **return**  $d_{min} = d'$ .

---

**Algorithm 2** Grover-Long searching

---

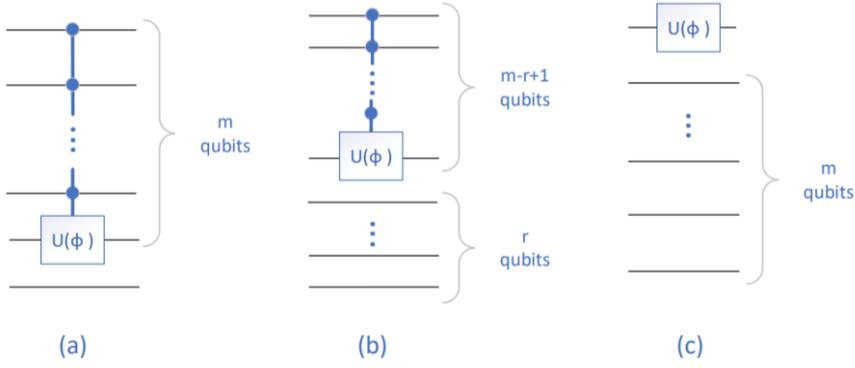
**Input:** initial state  $|\varphi\rangle$ ,  $t$ ,  $d'$ .  
**Output:** final state  $|\varphi'\rangle$ .

- 1:  $\phi = 2 \arcsin\left(\frac{\sin\frac{\pi}{4t+2}}{\sqrt{M/N}}\right)$ ;
- 2:  $|\varphi_0\rangle = |\varphi\rangle$ ;
- 3: **for**  $k = 0$  **to**  $t$  **do**
- 4:     Apply the oracle operation on  $|\varphi_k\rangle \rightarrow |\tilde{\varphi}\rangle = e^{i\phi} \sum_{d_m \leq d'} |d_m\rangle + \sum_{d_n > d'} |d_n\rangle$
- 5:     Apply the phase reverse operation on  $|\tilde{\varphi}\rangle \rightarrow |\varphi_{k+1}\rangle$
- 6: **end**
- 7: Return  $|\varphi'\rangle = |\varphi_{k+1}\rangle$ .

---

$j < N\}$ . All data items all are encoded into a n-qubit superposition state  $|\varphi\rangle$ . Algorithm 1 gives the specific steps of our OQMSA algorithm.

As shown in **Algorithm 1**, we firstly prepare the initial state  $|\varphi\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |d_j\rangle$  according to  $N$  data items. Then, the dynamic strategy is used to increase the probability of target state. The dynamic strategy is as follows: When  $M/N > 1/9$ , the number of iterations  $t'$  is an integer chosen randomly range from 0 to  $\lceil t \rceil$  ( $\lceil t \rceil$  is max number of current iteration). And then, the  $t'$ -iteration Grover-Long searching (as shown in **Algorithm 2**) is applied on  $|\varphi\rangle$ . If  $M/N < 1/9$ , the  $t_{max}$ -iteration Grover-Long searching is applied on  $|\varphi\rangle$ . Finally, we obtain the measure value  $r$  and compare  $r$  with  $d'$ , the  $d'$



**Fig. 1** The schematic diagram of the first simplified rule. (a) The circuit for marking two continuous states. (b) The circuit for marking  $2^r$  continuous states. (c) The circuit for marking  $2^m$  continuous states.

will be replaced if  $r < d'$ . This process will repeat until we get same current minimum for  $\lceil \text{Log}N \rceil$  times, and the algorithm will end with returning  $d_{min}$ .

### 3.2 Quantum circuit and its optimization

#### A. Oracle operation

The oracle operation plays a role in identifying and marking the target state in the circuit. The oracle's marker factor is a phase rotation that changes the amplitude in front of the target state to  $e^{i\phi}$ . The oracle can be described as diagonal matrix that only has  $e^{i\phi}$  and 1, as shown in Eq.(1)

$$O = e^{i\phi} \sum |v\rangle \langle v| + \sum_{\tau \neq v} |\tau\rangle \langle \tau|, \quad (1)$$

where  $\nu$  is the state which need to be marked.

**Rule 1.** When  $d' = 2^m - 1$ , where  $d'$  is current minimum, the oracle can be simplified as below:

$$O(m) = (e^{i\phi} |0_0\rangle \langle 0_0| + |1_0\rangle \langle 1_0|) \otimes I^{\otimes m}. \quad (2)$$

Since  $d' = 2^m - 1$ , we encode  $d' + 1$  into quantum state  $|1_0\rangle \otimes |0_1 0_2 \cdots 0_m\rangle$ . The first qubit of all solutions must be  $|0\rangle$ , they can be marked only if we ensure that the first qubit is  $|0\rangle$ . The oracle will become an operation that only change the amplitude of state where the first qubit is  $|0\rangle$  to  $e^{i\phi}$ . The schematic diagram is shown in Fig. 1.

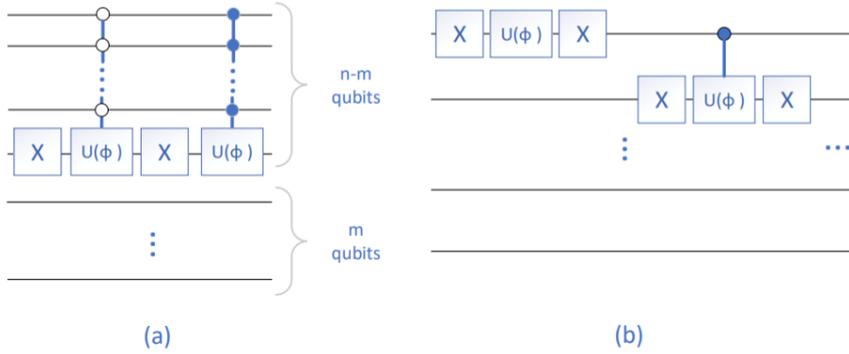
**Rule 2.** When  $d' \neq 2^m - 1$ ,  $d' + 1 = \sum_{i=0}^{n-1} a_i \cdot 2^{n-i-1}$ ,  $a_i \in \{0, 1\}$ , the oracle should be simplified iteratively through Algorithm 3.

**Algorithm 3** The iterative simplified algorithm**Input:** The binary string of  $d'$ ,  $(a_0 a_1 \cdots a_{n-1})$ .**Output:**  $O$ .

```

1:  $i = 0$ ;
2: while  $(i < n)$  do
3:   if  $a_i = 1$  then
4:     if  $i = 0$  then iiii
5:       Obtain  $O'(i) = O(n - i - 1)$  referring to Rule 1;
6:     else
7:       Obtain  $O'(i) = |a_0 a_1 \cdots a_{i-1}\rangle \langle a_0 a_1 \cdots a_{i-1}| \otimes O(n - i - 1)$ ;
8:   else
9:     if  $i = 0$  then iiii
10:      Obtain  $O'(i) = |1\rangle \langle 1| \otimes I^{\otimes n-i-1}$ ;
11:    else
12:      Obtain  $O'(i) = |a_0 a_1 \cdots a_{i-1}\rangle \langle a_0 a_1 \cdots a_{i-1}| \otimes |1\rangle \langle 1| \otimes I^{\otimes n-i-1}$ ;
13:    iiii
14:     $i = i + 1$ ;
15: Obtain  $O'(i) = |a_0 a_1 \cdots a_{n-1}\rangle \langle a_0 a_1 \cdots a_{n-1}|$ 
16: return  $O = \sum O'(i)$ 

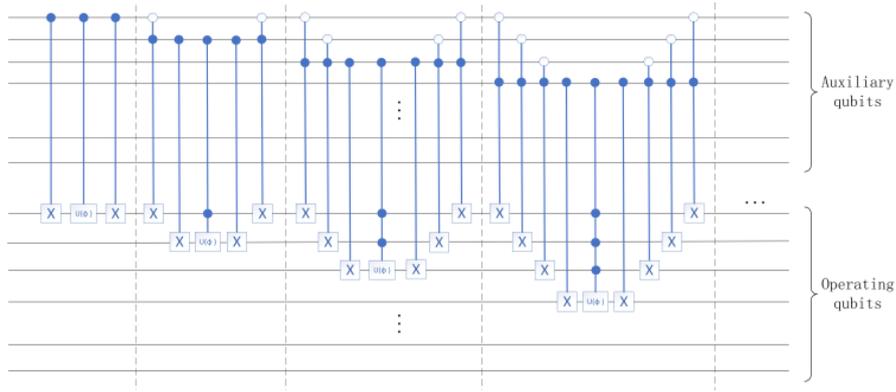
```



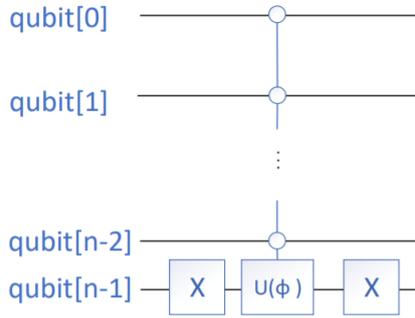
**Fig. 2** A schematic diagram of the second equivalent simplified rule. (a) The original circuit. (b) The simplified circuit.

For the case  $d' \neq 2^m - 1$ ,  $d' + 1 = \sum_{i=0}^{n-1} a_i \cdot 2^{n-i-1}$ ,  $d' + 1$  is encoded into  $|a_1 a_2 \cdots a_{n-1}\rangle$ . Algorithm 3 will determine which states need to be marked by iterating on a binary string of  $a_1 a_2 \cdots a_{n-1}$ . Then the corresponding quantum circuit can be divided into many sub-circuits such as Fig. 2(a). Finally, we can obtain the oracle which is used to mark all solutions after  $n$  iterations and it can be simplified as Fig. 2(b). The general optimized oracle circuit is showed in the Fig. 3.

For instance, when  $d' = 31 = 2^5 - 1$  and  $n=6$ , we simplify the oracle as  $O(5) = (e^{i\phi} |0\rangle \langle 0| + |1\rangle \langle 1|) \otimes I^{\otimes 5}$  according to Rule 1. We only need to operate on the first qubit to mark all the items range from 0(000000) to 31(011111). When  $d' = 35 = 2^5 + 2^2 - 1$ , we can obtain the oracle  $((e^{i\phi} |0\rangle \langle 0| + |1\rangle \langle 1|) \otimes I^{\otimes 5} + |100\rangle \langle 100| \otimes (e^{i\phi} |0\rangle \langle 0| + |1\rangle \langle 1|) \otimes I^{\otimes 2})$  according to **Rule 2**. Where



**Fig. 3** The optimized circuit for the oracle in OQMSA.



**Fig. 4** The general circuit for  $I_0$  operation.

the oracle mark the items range from 0(000000) to 31(011111), and then mark the items between 32(100000) and 35(100011). In this way, all states  $\leq 35$  are marked out.

### B. Phase deflection

The operation of phase deflection use  $e^{i\phi}$  to distinguish the states, because the oracle use  $e^{i\phi}$  to mark the target states. Besides, this operation can be divided into three parts:  $W^{-1}$ ,  $I_0$  and  $W$ , where  $W$  is an operation of preparing initial state,  $I_0$  is the core of the whole phase deflection. The operator of  $I_0$  can be described as a diagonal matrix, as shown in Eq. 3.

$$I_0 = e^{i\phi}|0\rangle\langle 0| + \sum_{\tau=1}^{2^n-1} |\tau\rangle\langle \tau| = \text{diag} [e^{i\phi}, 1, \dots, 1]_{2^n}, \quad (3)$$

where  $n$  is the number of qubits.  $I_0$  can be converted to the quantum circuit, as shown in Fig. 4.

## 4 Performance Evaluation

We analyze the performance of OQMSA from two aspects: the success rate and the computation complexity.

### 4.1 Success Rate

As described in the literature [27], the success rate of the DHA algorithm is slightly greater than 0.5. Through the following derive, we can know that our algorithm has a higher success rate. First of all, if we want to obtain the exact value of  $t_{max}$ ,  $\varphi$ ,  $\beta$ , we must know  $M/N$ . However, the dataset is not directly available. When a random value  $d'$ , is selected, we do not know how many solutions. We replace the exact values  $M, N$  with estimated values  $\widetilde{M}, \widetilde{N}$ . Therefore, the unknown database has  $\widetilde{N} = 2^n$  non-repeating values and the estimated number of marked states is  $\widetilde{M} = d' + 1$ . The distribution function of the estimated database is regarded as a uniform distribution, where the probability density function is  $\widetilde{\rho}(x) = \frac{1}{\widetilde{N}}$ . And the distribution function of the real database is unknown. It is important to quantify the impact of the gap between  $\frac{\widetilde{M}}{\widetilde{N}}$  and  $\frac{M}{N}$  on the failure rate  $\varepsilon_{GL}$ .

The cumulative distribution function of the estimated database in  $[0, d']$  is described as:

$$\widetilde{P}(x) = \int_0^{d'} \widetilde{\rho}(x) dx = \frac{\widetilde{M}}{\widetilde{N}}. \quad (4)$$

Similarly, the cumulative distribution function of the real database  $P(x)$  is  $\frac{M}{N}$ . Therefore, if the actual database follows the uniform distribution in  $[0, 2^n]$ , then  $\frac{\widetilde{M}}{\widetilde{N}} : \frac{M}{N} \approx 1$ . In other words, the success rate of our algorithm is close to 1.

### 4.2 Complexity

Note that the complexity of OQMSA is primarily composed of a total number of Grover-Long iterations and the initial state preparation. So we calculate the complexity without other steps. One main loop possesses  $t_{max}$  Grover-Long iterations.  $t_{max}$  can be described as Eq. 5

$$t_{max} = \text{floor} \left( \frac{\frac{\pi}{2} - \arcsin \left( \sqrt{\frac{M}{N}} \right)}{\arcsin \left( \sqrt{\frac{M}{N}} \right)} \right) + 1. \quad (5)$$

For convenience, we consider the case of an infinity database, so

$$\lim_{\sqrt{M/N} \rightarrow 0} \arcsin \left( \sqrt{\frac{M}{N}} \right) \approx \sqrt{\frac{M}{N}}. \quad (6)$$

We can simplify the complexity of Grover-Long algorithm as Eq. 7.

$$t_{max} = \left(\frac{\pi}{2} - 1 + 1\right) \times \sqrt{\frac{N}{M}} = \frac{\pi}{2} \sqrt{\frac{N}{M}}. \quad (7)$$

The total of Grover-Long iterations can be described as Eq. 8.

$$R_G = \sum_{k=0}^{K-1} J_k = \frac{\pi}{2} \sum_{k=0}^{K-1} \sqrt{\frac{N}{M_k}}. \quad (8)$$

where  $K$  represents the total number of main loops,  $M_k$  represents the number of marked states of the  $k$ -th main loop. Since the marked states have the same amplitude after applying Grover-Long algorithm, the number of  $(k+1)$ -th main loop's marked quantum states is nearly half of the  $(k)$ -th main loop's. Thus, the complexity of all Grover-Long algorithm iterations can be described as Eq. (9):

$$\begin{aligned} R_G &= \frac{\pi}{2} \left( \sqrt{\frac{N}{M_0}} + \sqrt{\frac{2N}{M_0}} + \dots + \sqrt{\frac{M_0 N}{M_0}} \right) = \frac{\pi}{2} \frac{\sqrt{\frac{N}{M_0}} \times (1 - \sqrt{2M_0})}{1 - \sqrt{2}} \\ &= \frac{\pi}{2} (\sqrt{2} + 1) \left( \sqrt{2N} - \sqrt{\frac{N}{M_0}} \right). \end{aligned} \quad (9)$$

Next, we consider the complexity of the initial state preparation. Since it needs to be executed about  $\log_2 N$  times and each execution takes  $\log_2 N$  steps, it can be described as Eq.(10)

$$R_{init} = (\log_2 N)^2. \quad (10)$$

We can calculate the complexity as Eq.(11)

$$R = R_G + R_{init} = \frac{\pi}{2} (\sqrt{2} + 1) \left( \sqrt{2N} - \sqrt{\frac{N}{M_0}} \right) + (\log_2 N)^2. \quad (11)$$

As claimed in Ref. [27], the complexity of DHA is  $22.5\sqrt{N} + 1.4(\log_2 N)^2$ , while the complexity of our algorithm is  $\frac{\pi}{2}(\sqrt{2} + 1)(\sqrt{2N} - \sqrt{\frac{N}{M_0}}) + (\log_2 N)^2$ . Because  $\frac{\pi}{2}(\sqrt{2} + 1)(\sqrt{2N} - \sqrt{\frac{N}{M_0}}) + (\log_2 N)^2 < 2 * 3 * 2\sqrt{N} + (\log_2 N)^2 < 22.5\sqrt{N} + 1.4(\log_2 N)^2$ , our algorithm has a smaller time complexity.

Finally, we compare DHA algorithm with our algorithm under the same conditions. For convenience, we assume that  $M_0 = \frac{1}{2}N$ . The complexity comparison of the two algorithms is shown in the Fig. 5. It can be seen from the figure that as the algorithm increases, our algorithm has a greater advantage in the complexity of algorithm. Because OQMSA requires fewer gates in Oracle and decreases the number of iteration in dynamic construction of circuit with two simplified rules.

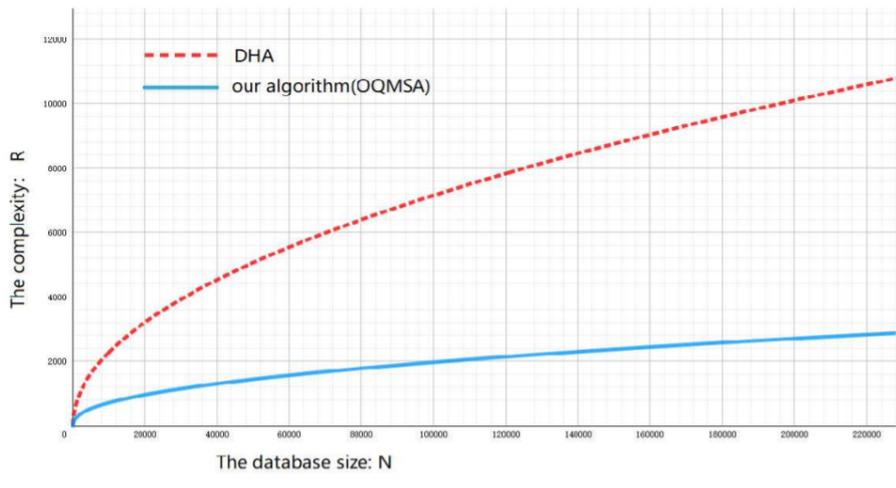


Fig. 5 Complexity comparison between DHA and our algorithms.

## 5 Experiment Simulation

Cirq[32], launched by Google in 2018, is a software library for writing, manipulating, and optimizing quantum circuits and then running them against quantum computers and simulators. Cirq attempts to expose the details of hardware, instead of abstracting them away, because, in the Noisy Intermediate-Scale Quantum (NISQ) regime, these details determine whether or not it is possible to execute a circuit at all. In order to verify the feasibility of our algorithm, we choose cirq as the experiment simulation platform, and design and implement a 6-qubits simulation experiment to perform the minimum searching in a data set (such as Table 1 or Table 2). Our experiments are conducted on a computer equipped with Intel Xeon 5218, double 2.3Ghz CPU, 64G RAM, and the version of Cirq is 5.0 under python 3.6.5.

In beginning of simulation, a random item is selected to be the current minimum  $d'$ , the unsorted database is encoded into  $n$  qubits according binary encoding method,  $d'$  is also encoded into auxiliary qubit  $|x_0x_1x_2x_3x_4x_5\rangle$ , the amplitude of each state is the same value  $\frac{1}{\sqrt{N}}$ . In the early stage of the algorithm, because the current minimum is likely to be very large, we choose the program of increasing the number of iterations dynamically to detect the upper bound in one algorithm. The advantage of this scheme is that we can get more accurate phase deflection without knowing how many solution sets there are. When the current minimum value decreases gradually, the previous scheme will cause the problem of building circuits many times. In this case, we set a threshold  $1/9$ . When  $d'/N$  is less than  $1/9$ , the advantages of the original algorithm are incarnated. We estimated the number of solutions  $\widetilde{M} = d' + 1$  and the estimated database size  $\widetilde{N} = 2^n$ , where  $n = 6$ . At the end of an algorithm, we can obtain a measurement result  $r$ . If  $r \leq d'$ , we replace the

**Table 1** Dataset A for searching the minimum

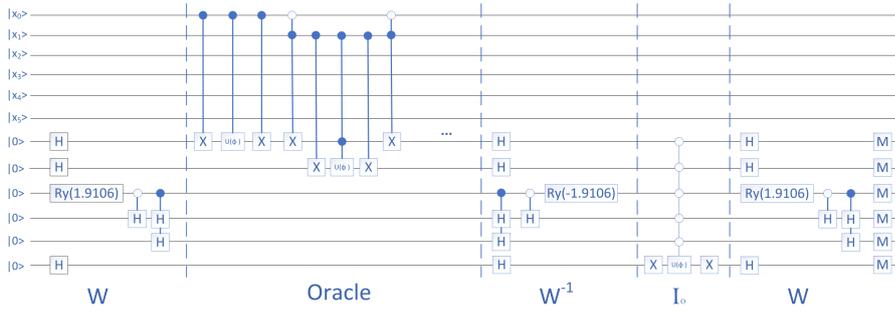
Value	After Encoding	Value	After Encoding	Value	After Encoding
2	000010	24	011000	44	101100
18	010010	40	101000	60	111100
34	100010	56	111000	14	001110
50	110010	10	001010	30	011110
6	000110	26	011010	46	101110
22	010110	42	101010	62	111110
38	100110	58	111010	3	000011
54	110110	12	001100	19	010011
8	001000	28	011100	35	100011
51	110011	7	000111	23	010111
9	001001	55	110111	39	100111
25	011001	41	101001	57	111001
43	101011	27	011011	11	001011
59	111011	13	001101	29	011101
15	001111	61	111101	45	101101
31	011111	47	101111	63	111111

**Table 2** Dataset B for searching the minimum

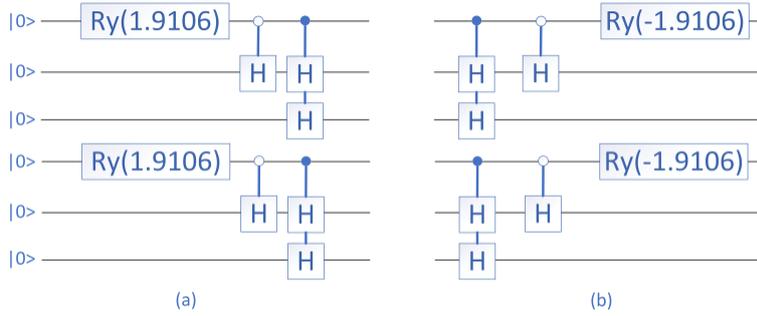
Value	After Encoding	Value	After Encoding	Value	After Encoding
45	101101	46	101110	42	101010
37	100101	38	100110	34	100010
21	010101	22	010110	18	010010
61	111101	62	111110	58	111010
53	110101	54	110110	50	110010
5	000101	6	000110	2	000010
44	101100	40	101000	47	101111
36	100100	32	100000	39	100111
20	010100	16	010000	23	010111
60	111100	56	111000	63	111111
52	110100	48	110000	55	110111
4	000100	0	000000	7	000111

current minimum with  $r$  and the algorithm is thought to operate successfully. By constantly changing the current minimum value, we will finally get the minimum value with a great probability. The complete circuit is shown in Fig. 6.

When the circuit is executed 1000 times, the average experimental results are shown in Fig. 8, the minimum represents the result of finding the minimum successfully and the others deputies the failure results of the experiment. As shown in the chart, we get the minimum for 982 times in one thousand experiments. Obviously, this consequence proves that our algorithm has a high success rate. Even if we can't find the minimum with 100 percent probability, we still think that the algorithm is efficient under some conditions. The results of 20 experiments are shown in Table 3.



**Fig. 6** A 12-qubits circuit of Grover-Long algorithm with 48 items.



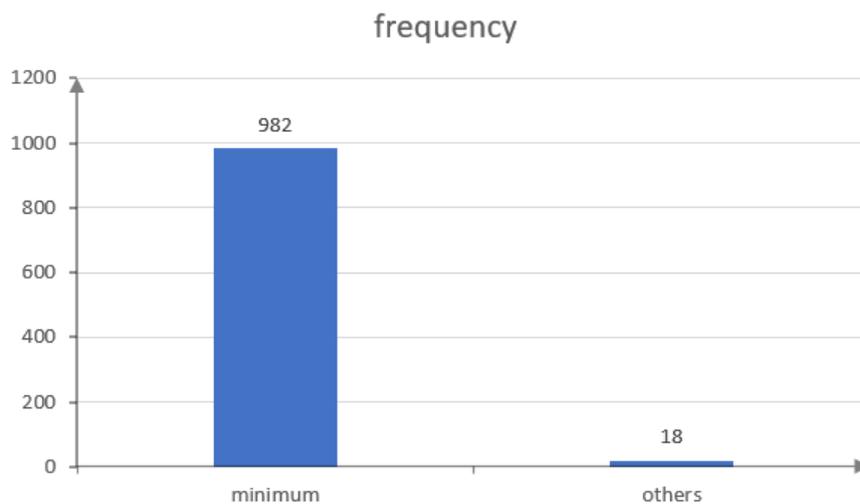
**Fig. 7**  $W$  and  $W^{-1}$  operation of Dataset B. (a) The  $W$  operation of Dataset B. (b) The  $W^{-1}$  operation of Dataset B.

**Table 3** The measurement results of 20 experiments

sequence	Dataset A		Dataset B	
	minimum	others	minimum	others
1	984	16	985	15
2	987	13	979	21
3	986	14	980	20
4	982	18	986	14
5	984	16	976	24
6	984	16	982	18
7	988	12	985	15
8	985	15	980	20
9	987	13	978	22
10	986	14	983	17

## 6 Conclusion

Based on Grover-Long algorithm, we proposed OQMSA, which is an improved version of DHA algorithm. Compared with classical algorithm, we have show the advantages of quantum algorithms in finding the minimum values to alleviate some of the challenges brought by the rapidly increasing amount



**Fig. 8** The average consequence of 20 experiments.

of data. Besides, as the size of the database increases, it has a higher probability of success and a greater advantage in terms of complexity than DHA algorithm. In addition, we provide corresponding general-purpose quantum circuits. The optimized circuits are easy to implement on any general-purpose quantum computer. Meanwhile, the general circuit design methods can be implemented on the quantum platform. We demonstrate the advantage of our algorithm through a group 12-qubit experiment (6 of them are auxiliary bits) which is executed on Cirq platform and a real issue that is numerical simulated. In addition to the computational tasks we show in this paper, the algorithm can be a subroutine in other quantum algorithms that need to find a minimum value.

We hope that the theoretical and experimental results we present here will push further research and motivate innovations of other mathematical models. The paradigm combining classical steps and quantum steps may work as an efficient solution in the era of big data.

## Acknowledgements

The authors would like to express heartfelt gratitude to the anonymous reviewers and editor for their comments that improved the quality of this paper. This work was supported by and in part by the Natural Science Foundation of China under Grant Nos. 62071240 and 61802002, the Natural Science Foundation of Jiangsu Higher Education Institutions of China under Grant No. 19KJB520028, the Graduate Research and Innovation Projects of Jiangsu Province under Grant No. KYCX20\_0978, the Practice Innova-

tion Training Program Projects for Jiangsu College Students under Grant No. 201910300140Y, and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

## References

1. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE T EVOLUT COMPUT* 6:182-197. <https://doi.org/10.1109/4235.996017>
2. Eberhart, Yuhui S (2001) Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation, IEEE 1:81-86. <https://doi.org/10.1109/CEC.2001.934374>
3. Ali H, Zhai XJ, Liu L, Tariq UU, Ieee (2018) Energy Efficient Heuristic Algorithm for Task Mapping on Shared-Memory Heterogeneous MPSoCs. In: Proceedings of 2018 IEEE 20th International Conference on High Performance Computing and Communications, IEEE 16th International Conference on Smart City, IEEE 4th International Conference on Data Science and Systems. IEEE, pp 1099-1104. <https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00183>
4. Tariq UU, Ali H, Liu L, Zhai X (2019) A Novel Meta-Heuristic for Green Computing on VFI-NoC-HMPSoCs. In: Proceedings of 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation. IEEE, pp 1545-1552. <https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00278>
5. Ali H, Tariq UU, Hussain M, Lu L, Panneerselvam J, Zhai X (2020) ARSH-FATI a Novel Metaheuristic for Cluster Head Selection in Wireless Sensor Networks. *IEEE SYST J Online*. <https://doi.org/10.1109/JSYST.2020.2986811>
6. Shang T, Liu J, Pei Z, Chen R, (2019) Quantum homomorphic signature with repeatable verification. *CMC-COMPUT MATER CON* 59(1):149-165. <https://doi.org/10.32604/cmc.2019.05360>
7. Liu W, Yu W, Xu Y, Yang JCN, Chi L (2019) Privacy-Preserving Quantum Two-Party Geometric Intersection. *CMC-COMPUT MATER CON* 60(3):1237-1250. <https://doi.org/10.32604/cmc.2019.03551>
8. Wei H, Qi S, Bin L, Yuan-Hang H, Fan F, Bing-Jie X (2017) Efficient multiparty quantum key agreement with collective detection. *SCI REP-UK* 7(1):1-9. <https://doi.org/10.1038/s41598-017-15227-6>
9. Liu WJ, Xu Y, Yang CN, Gao PP, Yu WB (2018) An efficient and secure arbitrary n-party quantum key agreement protocol using bell states. *INT J THEOR PHYS* 57(1):195-207. <https://doi.org/10.1007/s10773-017-3553-x>
10. Hillery M, Buzek, Berthiaume A (1999) Quantum secret sharing. *PHYS REV A* 59(3):1829-1834. <https://doi.org/10.1103/PhysRevA.59.1829>
11. Liu WJ, Xu Y, Zhang M, Chen J, Yang CN (2019) A Novel Quantum Visual Secret Sharing Scheme. *IEEE ACCESS* 7:114374-114384. <https://doi.org/10.1109/ACCESS.2019.2931073>
12. Fitzsimons JF, Kashefi E (2017) Unconditionally verifiable blind quantum computation. *PHYS REV A* 96(1):012303. <https://doi.org/10.1103/PhysRevA.96.012303>
13. Liu W, Xu Y, Liu W, Wang H, Lei Z (2019) Quantum Searchable Encryption for Cloud Data Based on Full-Blind Quantum Computation. *IEEE ACCESS* 7:186284-186295. <https://doi.org/10.1109/ACCESS.2019.2960592>
14. Gao F, Qin SJ, Huang W, Wen QY (2019) Quantum private query: A new kind of practical quantum cryptographic protocol. *SCI CHINA PHYS MECH* 62(7):70301. <https://doi.org/10.1007/s11433-018-9324-6>
15. Liu WJ, Gao PP, Liu ZH, Chen HW, Zhang MJ (2019) A Quantum-Based Database Query Scheme for Privacy Preservation in Cloud Environment. *SECUR COMMUN NETW* 2019:4923590. <https://doi.org/10.1155/2019/4923590>

16. Schuld M, Killoran N (2019) Quantum machine learning in feature Hilbert spaces. *PHYS REV LETT* 122(4):040504. <https://doi.org/10.1103/PhysRevLett.122.040504>
17. Liu WJ, Chen JX, Wang YX, Gao PP, Lei ZB, Ma X (2020) Quantum-Based Feature Selection for Multiclassification Problem in Complex Systems with Edge Computing. *COMPLEXITY* 2020:8216874. <https://doi.org/10.1155/2020/8216874>
18. Shor PW (1994) Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In: Proceedings of the 35th Annual Symposium on the Foundations of Computer Science. IEEE, pp 124-134. <https://doi.org/10.1109/SFCS.1994.365700>
19. Grover LK (1996) A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing (STOC'96). ACM, pp 212-219. <https://doi.org/10.1145/237814.237866>
20. Diao Z (2010) Exactness of the original Grover search algorithm. *PHYS REV A* 82(4):044301. <https://doi.org/10.1103/PhysRevA.82.044301>
21. Long GL, Li YS, Zhang WL, Niu L (1999) Phase matching in quantum searching. *PHYS LETT A* 262(1):27-34. [https://doi.org/10.1016/S0375-9601\(99\)00631-3](https://doi.org/10.1016/S0375-9601(99)00631-3)
22. Biham E, Biham O, Biron D, Grassl M, Lidar DA (1999) Grover's quantum search algorithm for an arbitrary initial amplitude distribution. *PHYS REV A* 60(4):2742 <https://doi.org/10.1103/PhysRevA.60.2742>
23. Biham E, Biham O, Biron D, Grassl M, Lidar DA, Shapira D (2000) Analysis of generalized Grover quantum search algorithms using recursion equations. *PHYS REV A* 63(1):012310. <https://doi.org/10.1103/PhysRevA.63.012310>
24. Long GL (2001) Grover algorithm with zero theoretical failure rate. *PHYS REV A* 64(4):022307. <https://doi.org/10.1103/PhysRevA.64.022307>
25. Long GL, Li X, Sun Y (2002) Phase matching condition for quantum search with a generalized initial state. *PHYS LETT A* 294(3-4):143. [https://doi.org/10.1016/S0375-9601\(02\)00055-5](https://doi.org/10.1016/S0375-9601(02)00055-5)
26. Grover L (2005) Fixed-point quantum search. *PHYS LETT A* 95(15):150501. <https://doi.org/10.1103/PhysRevLett.95.150501>
27. Durr C, Hoyer P (1999) A quantum algorithm for finding the minimum. [arXiv:quant-ph/9607014v2](https://arxiv.org/abs/quant-ph/9607014v2)
28. Woronowicz SL (2000) Quantum Exponential Function. *REV MATH PHYS* 12(6):873-C920. <https://doi.org/10.1142/S0129055X00000344>
29. Boyer M, Brassard G, Hoyer P, Tapp A (1998) Tight bounds on quantum searching. *FORTSCHR PHYS* 46(4-5):493-505. [https://doi.org/10.1002/\(SICI\)1521-3978\(199806\)46:4/5<493::AID-PROP493>3.0.CO;2-P](https://doi.org/10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P)
30. Brassard G, Hoyer P, Tapp A (1998) Quantum counting. In: Proceedings of 1998 International Colloquium on Automata, Languages, and Programming. Springer, pp 820-831. <https://doi.org/10.1007/BFb0055105>
31. Castagnoli G (2016) Highlighting the mechanism of the quantum speedup by time-symmetric and relational quantum mechanics. *FOUND PHYS* 46(3):360-381. <https://doi.org/10.1007/s10701-015-9968-4>
32. Cirq (2019) A python library for NISQ circuits. <http://cirq.readthedocs.io/en/stable/>. Accessed 18 November 2019