Quantum Solution for the 3-SAT Problem Based on IBM Q

Ying Zhang[1,2], Yu-xiang Bian[2,3], Qiang Fan[2,3], and Junxiu Chen[4(✉)]

[1] NARI Information and Communication Technology Co., Ltd., Nanjing 210003, China
zhang_ying2@sgepri.sgcc.com.cn

[2] NARI Group Corporation/State Grid Electric Power Research Institute, Nanjing 211106, China
bianyuxiang@sgepri.sgcc.com.cn

[3] NRGD Quantum Technology Co., Ltd., Nanjing 211106, China
nrgd_lw@163.com

[4] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
cjxccc981@163.com

Abstract. Quantum computing is currently considered to be a new type of computing model that has a subversive impact on the future. Based on its leading information and communication technology advantages, IBM launched IBM Q Experience cloud service platform, and achieved phased research results in the quantum simulator and programming framework. In this paper, we propose a quantum solution for the 3-SAT problem, which includes three steps: constructing the initial state, computing the unitary U_f implementing the black-box function f and performing the inversion about the average. In addition, the corresponding experimental verification for an instance of the Exactly-1 3-SAT problem with QISKit, which can connect to IBM Q remotely, is depicted. The experimental result not only show the feasibility of the quantum solution, but also serve to evaluate the functionality of IBM Q devices.

Keywords: Quantum computing · 3-SAT problem · IBM Q · QISKit · Grover algorithm

1 Introduction

With the continues development of quantum field, quantum computing has become a hot research field. And a key to quantum computing is the study of quantum algorithms. Firstly, Feynman [1] proposed the idea of combining

J. Chen—This work is supported by Science and Technology Project of NRGD Quantum Technology Co., Ltd., "Research on Power Quantum Security Service Platform and Key Technologies of Multi-mode Access".

X. Zhang et al. (Eds.): CloudComp 2019/SmartGift 2019, LNICST 322, pp. 410–423, 2020.
https://doi.org/10.1007/978-3-030-48513-9_33

quantum mechanics with computational problems in 1982. Then, Deutsch and Jozsa proposed Deutsch-Jozsa algorithm [2] which is the first quantum algorithm. Compared with the traditional calculation, the algorithm has obvious acceleration. In 1994, the emergence of Shor algorithm [3] not only accelerates the computing speed, and also affects the running basis of RSA encryption technology. In addition, the search algorithm proposed by Grover [4] in 1996 is also a very classic algorithm. Generally speaking, the search complexity of computers can be depicted as N (N is the size of database). Because of its wide application, Grover algorithm further demonstrates the advantages of quantum computing. The above mentioned quantum algorithms, especially Shor algorithm and Grover algorithm, fully demonstrate the development potential of quantum computing, which can be widely used in data security and commercial development. At present, there are many new quantum algorithms come into being, such as quantum secure sharing (QSS) [5,6], quantum key agreement (QKA) [7,8], quantum secure direct communication (QSDC) [9–11], quantum private comparison (QPC) [12–15], quantum sealed-bid auction (QSBA) [16,17], remote preparation of quantum states [18–21], quantum steganography [22–24], delegating quantum computation [25,26], and quantum machine learning algorithms [27,28]. However, these algorithms also have some shortcomings and can be improved. However, these improvements are theory research, few would give the experimental results show that, after all, from the real it's early, in other words, the advent of quantum computer algorithm to improve the existence of a real error, whether can the real experiment, whether real applications remains to be further investigation Therefore, we need a quantum simulation cloud platform, which can provide us with a good environment to verify and improve quantum algorithms.

At the end of 2017, an open-source quantum computing framework, namely QISKit [29], is released by IBM, which allows the users to implement remote quantum experimental verification of IBM Q [30] through localized python programming. The SAT problem is the oldest and well-known NP-Complete problem. There have been many algorithms and techniques that have been invented to solve it. The best SAT solvers can solve tens of thousands of variables. We know that the NP-Complete problem could be converted in polynomial time, and the SAT solver is very efficient. Then the SAT is naturally very practical. In this paper, we use QISKit to directly program the algorithm for an Exactly-1 3-SAT problem, and connect the remote IBM 5-qubit device (i.e., ibmqx4) to verify the quantum solution in the real quantum computer.

The remaining structure of the paper is as follows: In Sect. 2, preliminaries about quantum computation, IBM Q and Grover's algorithm are briefly introduced. In Sect. 3, the quantum solution based on the Grover's algorithm, which is used to solve the 3-SAT problem, is depicted. In addition, we give an instance of this problem, and then its implementation circuit and QISKit program are presented, respectively. Subsequently, the experimental result of the solution for Exactly-1 3-SAT problem are analyzed in detail in graphical form. Finally, Sect. 4 is dedicated for conclusion.

2 Preliminaries

2.1 Quantum Computation

According to the analysis of the superposition principle, we can find that the change of state of quantum information units can add up with several possible situation. For a quantum state, we can control a quantum bit to get two quantum state. If we can control N qubits at the same time, it represents that we can get 2^N condition for effective control, information storage times to grow exponentially. This is a quantum computer parallel computing ability. We can design different quantum circuits by referring to quantum gate sets, so that we can use quantum states to transmit information and achieve the purpose of quantum communication.

In traditional computer, a bit represents a Boolean variable with a range of $\{0, 1\}$. However, a quantum bit is a vector which is in 2-dimensional complex Hilbert space, which represents the state of a two-state quantum system. $|0\rangle$ and $|1\rangle$ are used to refer to the corresponding "0" and "1" state, where "$|\rangle$" is called Dirac notation. Quantum ratio has two unique properties: superposition and entanglement, which can be used for specific calculations. Specifically, superposition state refers to the state of a single quantum bit that could be described as the superposition of two ground states, just like schrodinger's cat, which is a superposition of life and death. The quantum superposition state $|\varphi\rangle$ is expressed as

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{1}$$

α and β represent the probability amplitude of $|0\rangle$ and $|1\rangle$ respectively, where $|\alpha|^2$ can be understood as the probability of observing the quantum bit is 0, and $|\beta|^2$ could be understood as the probability of observing the quantum bit is 1. The state of quantum bit satisfies normalization, namely: $|\alpha|^2 + |\beta|^2 = 1$. The superposition state of a qubit could be viewed as the linear superposition of states 0 and 1. It is linear and entangled state.

Quantum computers are built from quantum circuits that contain wires and basic quantum gates, which is in order to carry and manipulate some of the quantum bits that carry communication information. Quantum gates can be divided into two types: single qubit gates and multiple qubit gates. Quantum gate can all be represented in the form of a matrix U. The unitary limit ($U^\dagger U = I$, where $U^\dagger$ is a conjugate transpose of U, obtained by U transpose and complex conjugate of U) is the only limitation on quantum gates [31], each valid quantum gate can be represented as a unitary matrix. For visual display, in Table 1 below we list some line symbols and matrix representations used in this paper.

In the actual quantum circuit, we use special line symbol to represent the quantum gate, and a line symbol represents a quantum gate that can manipulate the quantum state.

Table 1. Common quantum gate and line symbols

Quantum gate	Line symbol	Matrix form
Hadamard		$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
Pauli-X		$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$
Pauli-Y		$\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$
Pauli-Z		$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$
Controlled-NOT		$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$
Controlled-Z		$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{bmatrix}$

$$X \equiv \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}. \tag{2}$$

If X-gate is applied to manipulate the quantum state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, the result of the operation can be obtained by multiplying the vector:

$$X\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}\beta\\ \alpha\end{bmatrix}. \tag{3}$$

2.2 IBM Q

In 2016, IBM opened the IBM Q experience prototype 5-qubit device to the public. Then, they launched IBM Q [30], which is the first general-purpose quantum computer for commercial and scientific research. In the same year, they proposed two devices with 5 qubits named ibmqx2 and ibmqx4. In 2018, a third public device with 16 qubits (ibmqx5) is added which can be accessed using QISKit. Recently, they have announced that they successfully built and tested a 20-qubit

device for their client. Meanwhile, their simulator is up to 32 qubits for more and more people to use and research. The experimental verification of quantum algorithms has become a reality, providing reliable data support for the research and development of quantum algorithms to help us assist in the analysis of the performance of the algorithm.

In IBM Q, all devices provide a lot of elementary gates, such as: X-gate, H-gate, cX-gate (control-NOT gate), cZ-gate (control-Z gate), ccX-gate (control-control-NOT gate, namely Toffoli gate) and so on. The coupling map of ibmqx2 and ibmqx4 are shown in Fig. 1. In general, two qubit gates may act between adjacent qubits, which are connected by a superconducting bus resonator. IBM Q experienced the use of cross-resonance interaction technology as the basis for quantum gate operations. When the qubits with higher frequencies are selected as control qubits and the qubits with lower frequencies are selected as targets, the interaction becomes stronger, so the qubits' frequency determines the gate's steering direction.

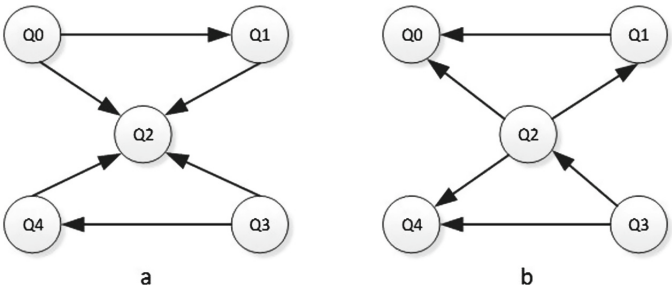

Fig. 1. The coupling map picture: ibmqx2(a) and ibmqx4(b). The arrows point from the qubit with higher frequency to that with lower frequency

Quantum Information Software Kit(QISKit) [29] is a Software Kit for designing short-depth Quantum circuits and building short-term Quantum applications and experiments on Quantum computers. Relevant data are collected after the program compilation is completed. In summary, the QISKit toolkit includes python-based software tools for creating, manipulating, visualizing, and exploring quantum states, tools for describing qubits, scripts for batch processing, and compilers for mapping required experiments to actual hardware. Different from the IBM Q web page experiment mode, this kind of programming call mode can overcome the cumbersomeness of drawing complex circuit diagrams on web pages, and has the advantage of easy expansion of composite quantum gates and easy preservation of experimental data.

2.3 Grover's Algorithm

Grover's algorithm [4] is one of the main algorithms in quantum computing. It is better than the best classical algorithm to do square acceleration when dealing with the problem of searching M target Numbers from unordered D databases.

In other words, the time required for the classical algorithm to complete the task is proportional to, while the quantum algorithm can be implemented within the time scale. If D is a very large number, it saves a lot of time. The power of the Grover algorithm lies in its versatility: its formulas are universal and can be applied to many problems, such as cryptography, matrix and graphics problems, optimization, and quantum machine learning. The detailed implementation steps of the Grover quantum search algorithm could be seen in Fig. 2:

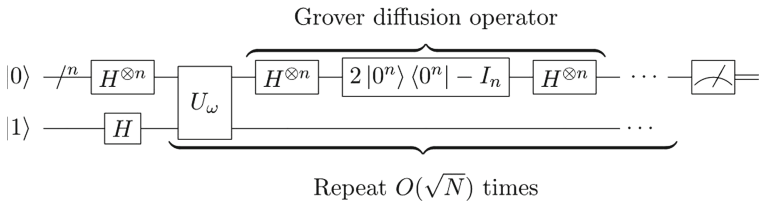

Fig. 2. Quantum circuit representation of Grover algorithm

The steps of Grover's algorithm are as follows: let $|s\rangle$ represents the uniform superposition of all states.

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle. \tag{4}$$

A "quantum oracle" operator U_ω is required in Grover's algorithm, which could identify the target solution to the search problem and turn their magnitude negative. And we apply the operator U_ω on the states. Then the operator $U_s = 2\,|s\rangle\langle s| - I$ is the grover-diffusion operator. Then the operator U_s is applied on the states after U_ω. Next, we perform the measurement Ω on then quantum state. The result will be eigenvalue λ_ω with probability approaching 100% for $N \gg 1$. From λ_ω, that Ω may be obtained.

3 Quantum Solution for the 3-SAT Problem Based on IBM Q

The SAT problem is well-known as the first NP-Complete problem [32]. The NP-Complete problem could be rotated within polynomial time. And the efficiency of the SAT solver is good. So the SAT problem is naturally very practical. SAT is used for test verification in many fields such as EDA. And it also can be used in AI fields such as automatic theorem proving and so on.

A quantum solution for the 3-SAT problem is proposed in this paper. The problem of 3-SAT can be described as follows: the assignment requires that each clause contain a truth value. For the input data, the formula in conjunctive normal form $\wedge_{k=1}^{m} C_k$ over n Boolean variables $x_1, \ldots, x_n$, with m literals per clause $C_1, \ldots, C_m$. For the Output data, is there an assignment of $x_1, \ldots, x_n$

such that every clause $C_1, \ldots, C_m$ has exactly one true literal? In order to solve this problem, a quantum solution for the 3-SAT problem based on the Grover's algorithm is proposed, which is to find satisfying assignment.

The quantum solution can be divided into three steps:

(1) constructing the initial state.
(2) computing the unitary U_f with the black-box function f.
(3) performing the inversion about the average.

Program 1: Quantum solution for the 3-SAT problem

input : SAT formula in conjunctive normal form $\wedge_{k=1}^{m} C_k$ over n Boolean variables x_1, x_2, x_3, with 3 literals per clause C_1, C_2, C_3.
output: The answer of x_1, x_2, x_3 such that every clause C_1, C_2, C_3 has exactly one true literal.

```
#constructing a 3-qubit initial state, the x4 is an auxiliary qubit;
circuit.x(x4);
for j in 4 do
    circuit.h(xj);
end
#computing the unitary Uf implementing the black-box function f;
formula=[C1, C2, C3];
def Uf(circuit, x1, x2, x3, formula)
#performing the inversion about the average
for j in range(3) do
    circuit.h(x[j]);
    circuit.x(x[j]);
end
for j in range(3) do
    circuit.ccZ(circuit,[x[j] for j in range(2)], x[2]);
end
for j in range(3) do
    circuit.x(x[j]);
    circuit.h(x[j]);
end
```

Program 1 shows the detailed procedure of the quantum solution for SAT problems.

In our experimental implementation, an Exactly-1 3-SAT instance is specified as a list of clauses, where there are three integers in each clause. In this paper, we agree that a positive integer is an index of positive text and a negative integer is the opposite. For example, the corresponding python list is shown as follows.

$$(x_1 \nabla x_2 \nabla \neg x_3) \wedge (\neg x_1 \nabla \neg x_2 \nabla \neg x_3) \wedge (\neg x_1 \nabla x_2 \nabla x_3) \tag{5}$$

We use the up formula as an example and the symbol ∇ is used to emphasize that this is an Exactly-1 3-SAT formula, rather than the usual $\wedge$. This is like a problem definition that requires each clause to have only one real word. Based on Grover's algorithm, we are going to use three subsections to show the processes of our solution.

3.1 Constructing the Initial State

We take 3 qubits as an example to show how to prepare an initial quantum state. In Grover's algorithm, a function with an n-qubit input and a single-qubit output is applied. We need n qubit and an ancilla qubit to prepare the initial state. The circuit for constructing a 3-qubit initial state can be seen in Fig. 3. And the corresponding program in python using QISKit can be seen in Program 2.

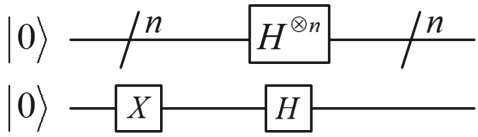

Fig. 3. Circuit for constructing an initial quantum state.

Program 2: Constructing the initial state

```
1 def input_state(n):
2     for j in range (n) do
3         circuit.h(f_in[j]);
4 end
5     circuit.x(f_out);
6     circuit.h(f_out);
```

3.2 Computing Unitary U_f Implementing the Black-Box Function f

The implementation of U_f for an Exactly-1 3-SAT problem is the most complex part of the code, and there are several ways to complete it. To reduce computational complexity and save resources, the problem of computing U_f is decomposed by introducing m ancilla qubits. For each clause, a highly efficient and feasible quantum circuit is constructed in which the phase of the corresponding auxiliary qubit is reversed if and only if there is only one true word in the clause (here, these auxiliary qubits will be initialized in state $|0\rangle$).

Taking $x_1 \wedge (\neg x_2) \wedge x_3 \wedge (x_1 \nabla \neg x_2 \nabla x_3)$ as an example, we show the corresponding circuit in Fig. 4. We apply three H gates to get initial states, and then the next three C-NOT (Control-Not) gates complete the function of $x_1 \wedge (\neg x_2) \wedge x_3$ and store the result in $q[3]$. And the next two CC-NOT (Control-Control-Not) gates complete the function of $(x_1 \nabla \neg x_2 \nabla x_3)$ and the result is stored in $q[3]$ finally.

Then, inspired by the above circuit, we drawn the circuit for $(x_1 \nabla x_2 \nabla \neg x_3) \wedge (\neg x_1 \nabla \neg x_2 \nabla \neg x_3) \wedge (\neg x_1 \nabla x_2 \nabla x_3)$ in Fig. 5. $q[3], q[4]$ and $q[5]$ are auxiliary qubits. And the final result is stored in $q[6]$. The corresponding program in python using QISKit can be seen in Program 3.

Program 3: Computing the unitary U_f implementing the black-box function f

```
def black_box_u_f(circuit, f_in, f_out, aux, n, exactly_1_3_sat_formula):
    num_clauses = len(exactly_1_3_sat_formula);
    for (k, clause) in enumerate(exactly_1_3_sat_formula) do
        for literal in clause do
            if literal > 0 then
                circuit.cx(f_in[literal − 1], aux[k]);
        end
            else
                circuit.x(f_in[−literal − 1]);
                circuit.cx(f_in[−literal − 1], aux[k]);
        end
    end
        circuit.ccx(f_in[0], f_in[1], aux[num_clauses]);
        circuit.ccx(f_in[2], aux[num_clauses], aux[k]);
        circuit.ccx(f_in[0], f_in[1], aux[num_clauses]);
        for literal in clause do
            if literal < 0 then
                circuit.x(f_in[−literal − 1]);
        end
    end
        if (num_clauses==1) then
            circuit.cx(aux[0], f_out[0]);
    end
        else if (num_clauses==2) then
            circuit.ccx(aux[0], aux[1], f_out[0]);
    end
        else if (num_clauses==3) then
            circuit.ccx(aux[0], aux[1], aux[num_clauses]);
            circuit.ccx(aux[2], aux[num_clauses], f_out[0]);
            circuit.ccx(aux[0], aux[1], aux[num_clauses]);
    end
end
    for (k , clause ) in enumerate(exactly_1_3_sat_formula) do
        for literal in clause do
            if literal > 0 then
                circuit.cx(f_in[literal − 1], aux[k]);
        end
            else
                circuit.x(f_in[−literal − 1]);
                circuit.cx(f_in[−literal − 1], aux[k]);
        end
    end
        circuit.ccx(f_in[0], f_in[1], aux[num_clauses]);
        circuit.ccx(f_in[2], aux[num_clauses], aux[k]);
        circuit.ccx(f_in[0], f_in[1], aux[num_clauses]);
        for literal in clause do
            if literal < 0 then
                circuit.x(f_in[−literal − 1]);
        end
    end
end
```

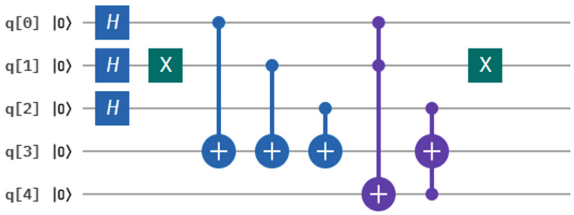

Fig. 4. Circuit for $x_1 \wedge (\neg x_2) \wedge x_3 \wedge (x_1 \nabla \neg x_2 \nabla x_3)$.

3.3 Performing the Inversion About the Average

In order to carry out inversion operation on the average value of all the obtained quantum state amplitudes, we need to perform the following operations:

$$\sum_{j\in\{0,1\}^n} \alpha_{j|j\rangle_n} \rightarrow \sum_{j\in\{0,1\}^n} \left(2\left(\sum_{k\in\{0,1\}^n} \frac{\alpha_k}{2^n} \right) - \alpha_{\bar{j}} \right) |j\rangle_n \tag{6}$$

where $\sum_{k\in\{0,1\}^n} \frac{\alpha_k}{2^n}$ is the average value of all the obtained quantum state amplitudes, and therefore we update the corresponding amplitudes by taking twice the average and subtracting each coefficient from it. This mapping is realized by the matrix as follows.

$$W = \begin{pmatrix} \frac{2}{2^n}-1 & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \frac{2}{2^n} & \frac{2}{2^n}-1 & \cdots & \frac{2}{2^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n}-1 \end{pmatrix} = \begin{pmatrix} \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \end{pmatrix} - I^{\otimes n} \tag{7}$$

And the corresponding program in python using QISKit can be seen in Program 4.

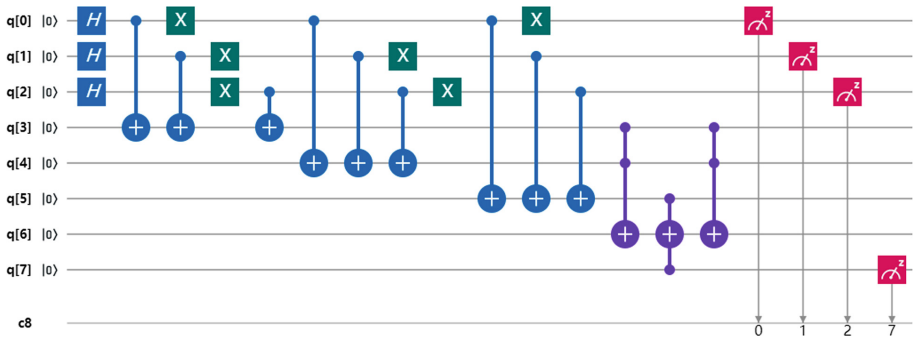

Fig. 5. Circuit for $(x_1 \nabla x_2 \nabla \neg x_3) \wedge (\neg x_1 \nabla \neg x_2 \nabla \neg x_3) \wedge (\neg x_1 \nabla x_2 \nabla x_3)$.

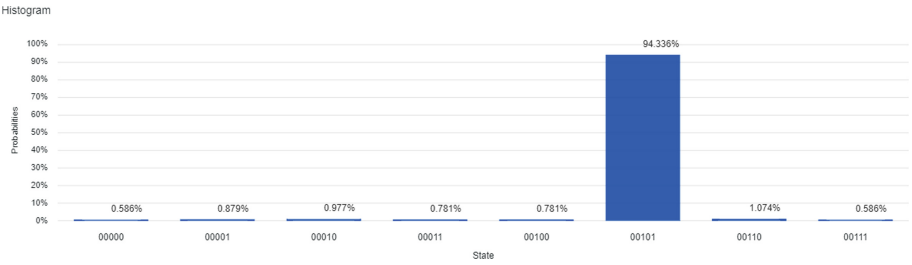

Fig. 6. Result for $(x_1 \nabla x_2 \nabla \neg x_3) \wedge (\neg x_1 \nabla \neg x_2 \nabla \neg x_3) \wedge (\neg x_1 \nabla x_2 \nabla x_3)$.

Together with the above three programs, Program 5 shows the whole program for quantum solution in python using QISKit, which is used to solve the Exactly-1 3-SAT problem.

After executing Program 5, we can get the result for an instance of Exactly-1 3-SAT in Eq. 5, which can be seen in Fig. 6. Obviously, $|101\rangle$ (i.e., $x_1 = 1,\ x_2 = 0,\ x_3 = 1$) is the answer to $(x_1 \nabla x_2 \nabla \neg x_3) \wedge (\neg x_1 \nabla \neg x_2 \nabla \neg x_3) \wedge (\neg x_1 \nabla x_2 \nabla x_3)$. Then, we could get the answer with 94% probability through the quantum solution.

Program 4: Inversion about the average

```
def inversion_about_average(circuit, f_in, n):
    for j in range(n) do
        circuit.h(f_in[j]);
end
    for j in range(n) do
        circuit.x(f_in[j]);
end
    3_controlled_Z(circuit, [f_in[j]forjinrange(n − 1)], f_in[n − 1]);
    for j in range(n) do
        circuit.x(f_in[j]);
end
    for j in range(n) do
        circuit.h(f_in[j]);
end
def 3_controlled_Z(circuit, controls, target):
    circuit.h(target);
    circuit.ccx(controls[0], controls[1], target);
    circuit.h(target);
```

Program 5: Quantum solution for Exactly-1 3-SAT problem

```
import sys;
from qiskit import QuantumRegister, ClassicalRegister;
from qiskit import QuantumCircuit;
from qiskit import compile, Aer;
from qiskit.tools import visualization;
n = 3;
exactly_1_3_sat_formula = [[1, 2, −3], [−1, −2, −3], [−1, 2, 3]];
f_in = QuantumRegister(n);
f_out = QuantumRegister(1);
aux = QuantumRegister(len(exactly_1_3_sat_formula) + 1);
ans = ClassicalRegister(n);
qc = QuantumCircuit(f_in, f_out, aux, ans, name =' grover');
input_state(qc, f_in, f_out, n);
black_box_u_f(qc, f_in, f_out, aux, n, exactly_1_3_sat_formula);
inversion_about_average(qc, f_in, n);
black_box_u_f(qc, f_in, f_out, aux, n, exactly_1_3_sat_formula);
inversion_about_average(qc, f_in, n);
for  j in range (n) do
  qc.measure(f_in[j], ans[j]);
quantum_simulator = Aer.get_backend('qasm_simulator_py');
qobj = compile(qc, quantum_simulator, shots = 2048);
job = quantum_simulator.run(qobj);
result = job.result();
counts = result.get_counts('grover');
visualization.plot_histogram(counts);
```

4 Conclusion

In this paper, a quantum solution for the 3-SAT problem based on Grover's algorithm is proposed, which includes three steps: constructing the initial state, computing the unitary U_f, implementing the black-box f and performing the inversion of the average amplitude of all quantum states. Next, the corresponding experimental verification for an Exactly-1 3-SAT problem instance with QISKit, which can connect to IBM Q remotely, is depicted. We can get the answer $|101\rangle$ with 94% probability through the quantum solution. The experimental result not only show the feasibility of the quantum solution, but also serve to evaluate the functionality of IBM Q devices. For this kind of localized programming mode based on QISKit, the design of quantum functional circuits can be packaged in the form of functions for reusing and expansion. With the increasing scale of practical problems, we need to consider the design of feasible quantum circuit optimization scheme.

References

1. Feynman, R.P.: Simulating physics with computers. Int. J. Theor. Phys. **21**, 467–488 (1982). https://doi.org/10.1007/BF02650179
2. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proc. R. Soc. Lond. Ser. A (Math. Phys. Sci.) **439**(1907), 553–558 (1992)
3. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. **41**(2), 303–332 (1999)
4. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of ACM Symposium on the Theory of Computing, pp. 212–219 (1996)
5. Liu, Z.-H., Chen, H.-W., Xu, J., Liu, W.-J., Li, Z.-Q.: High-dimensional deterministic multiparty quantum secret sharing without unitary operations. Quantum Inf. Process. **11**(6), 1785–1795 (2011). https://doi.org/10.1007/s11128-011-0333-z
6. Chen, X.B., Tang, X., Xu, G., Dou, Z., Chen, Y.L., Yang, Y.X.: Cryptanalysis of secret sharing with a single d-level quantum system. Quantum Inf. Process. **17**(9), 225 (2018)
7. Huang, W., Su, Q., Liu, B., He, Y.H., Fan, F., Xu, B.J.: Efficient multiparty quantum key agreement with collective detection. Sci. Rep. **7**(1), 15264 (2017)
8. Liu, W.J., Xu, Y., Yang, C.N., Gao, P.P., Yu, W.B.: An efficient and secure arbitrary N-party quantum key agreement protocol using bell states. Int. J. Theor. Phys. **57**(1), 195–207 (2018). https://doi.org/10.1007/s10773-017-3553-x
9. Liu, W.J., Chen, H.W., Li, Z.Q., Liu, Z.H.: Efficient quantum secure direct communication with authentication. Chin. Phys. Lett. **25**(7), 2354–2357 (2008)
10. Liu, W.J., Chen, H.W., Ma, T.H., Li, Z.Q., Liu, Z.H., Hu, W.B.: An efficient deterministic secure quantum communication scheme based on cluster states and identity authentication. Chin. Phys. B **18**(10), 4105–4109 (2009)
11. Xu, G., Chen, X.-B., Li, J., Wang, C., Yang, Y.-X., Li, Z.: Network coding for quantum cooperative multicast. Quantum Inf. Process. **14**(11), 4297–4322 (2015). https://doi.org/10.1007/s11128-015-1098-6
12. Liu, W., Liu, C., Wang, H., Jia, T.: Quantum private comparison: a review. IETE Tech. Rev. **30**(5), 439–445 (2013)
13. Liu, W.J., Liu, C., Liu, Z.H., Liu, J.F., Geng, H.T.: Same initial states attack in Yang et al.'s quantum private comparison protocol and the improvement. Int. J. Theor. Phys. **53**(1), 271–276 (2014)
14. Liu, W.J., Liu, C., Chen, H.W., Li, Z.Q., Liu, Z.H.: Cryptanalysis and improvement of quantum private comparison protocol based on bell entangled states. Commun. Theor. Phys. **62**(2), 210–214 (2014)
15. Liu, W.-J., Liu, C., Wang, H., Liu, J.-F., Wang, F., Yuan, X.-M.: Secure quantum private comparison of equality based on asymmetric W state. Int. J. Theor. Phys. **53**(6), 1804–1813 (2014). https://doi.org/10.1007/s10773-013-1979-3
16. Liu, W.-J., et al.: Multiparty quantum sealed-bid auction using single photons as message carrier. Quantum Inf. Process. **15**(2), 869–879 (2015). https://doi.org/10.1007/s11128-015-1202-y
17. Liu, W.J., Wang, F., Ji, S., Qu, Z.G., Wang, X.J.: Attacks and improvement of quantum sealed-bid auction with EPR pairs. Commun. Theor. Phys. **61**(6), 686–690 (2014)
18. Liu, W.J., Chen, Z.-F., Liu, C., Zheng, Y.: Improved deterministic N-to-one joint remote preparation of an arbitrary qubit via EPR pairs. Int. J. Theor. Phys. **54**(2), 472–483 (2015). https://doi.org/10.1007/s10773-014-2241-3

19. Chen, X.-B., Sun, Y.-R., Xu, G., Jia, H.-Y., Qu, Z., Yang, Y.-X.: Controlled bidirectional remote preparation of three-qubit state. Quantum Inf. Process. **16**(10), 1–29 (2017). https://doi.org/10.1007/s11128-017-1690-z
20. Qu, Z.G., Wu, S.Y., Wang, M.M., Sun, L., Wang, X.J.: Effect of quantum noise on deterministic remote state preparation of an arbitrary two-particle state via various quantum entangled channels. Quantum Inf. Process. **16**(306), 1–25 (2017)
21. Wang, M.M., Yang, C., Mousoli, R.: Controlled cyclic remote state preparation of arbitrary qubit states. CMC-Comput. Mater. Continua **55**(2), 321–329 (2018)
22. Qu, Z., Cheng, Z., Liu, W., Wang, X.: A novel quantum image steganography algorithm based on exploiting modification direction. Multimedia Tools Appl. **78**(7), 7981–8001 (2018). https://doi.org/10.1007/s11042-018-6476-5
23. Qu, Z., Chen, S., Ji, S., Ma, S., Wang, X.: Anti-noise bidirectional quantum steganography protocol with large payload. Int. J. Theor. Phys. **57**(6), 1903–1927 (2018). https://doi.org/10.1007/s10773-018-3716-4
24. Qu, Z.G., Zhu, T.C., Wang, J.W., Wang, X.J.: A novel quantum stegonagraphy based on brown states. CMC-Comput. Mater. Continua **56**(1), 47–59 (2018)
25. Liu, W.-J., Chen, Z.-Y., Ji, S., Wang, H.-B., Zhang, J.: Multi-party semi-quantum key agreement with delegating quantum computation. Int. J. Theor. Phys. **56**(10), 3164–3174 (2017). https://doi.org/10.1007/s10773-017-3484-6
26. Liu, W.J., Chen, Z.Y., Liu, J.S., Su, Z.F., Chi, L.H.: Full-blind delegating private quantum computation. CMC-Comput. Mater. Continua **56**(2), 211–223 (2018)
27. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum algorithms for supervised and unsupervised machine learning. eprint arXiv (2013)
28. Liu, W.-J., Gao, P.-P., Yu, W.-B., Qu, Z.-G., Yang, C.-N.: Quantum relief algorithm. Quantum Inf. Process. **17**(10), 1–15 (2018). https://doi.org/10.1007/s11128-018-2048-x
29. QISKit: Open Source Quantum Information Science Kit. https://qiskit.org/. Accessed 12 Apr 2018
30. IBM quantum computing platform. https://www.research.ibm.com/ibm-q/. Accessed 11 Apr 2017
31. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information, 10 Anniversary edn. Cambridge University Press, Cambridge (2011)
32. Garey, M.R., Johnson, D.S.: Computers and Intractability, vol. 29. W. H. Freeman and Company, New York (1972)