Quantum Algorithms and Experiment Implementations Based on IBM Q

Wenjie Liu[1, 2, *], Junxiu Chen[2], Yinsong Xu[2], Jiahao Tang[2], Lian Tong[3] and Xiaoyu Song[4]

Abstract: With the rapid development of quantum theory and technology in recent years, especially the emergence of some quantum cloud computing platforms, more and more researchers are not satisfied with the theoretical derivation and simulation verification of quantum computation (especially quantum algorithms), experimental verification on real quantum devices has become a new trend. In this paper, three representative quantum algorithms, namely Deutsch-Jozsa, Grover, and Shor algorithms, are briefly depicted, and then their implementation circuits are presented, respectively. We program these circuits on python with QISKit to connect the remote real quantum devices (i.e., ibmqx4, ibmqx5) on IBM Q to verify these algorithms. The experimental results not only show the feasibility of these algorithms, but also serve to evaluate the functionality of these devices.

Keywords: Quantum algorithms, implementation circuit, IBM Q, QISKit program.

1 Introduction

Quantum computation [Nielsen and Chuang (2002)] can be understood as the method of information processing using the physical properties of quantum states on a quantum computer. With quantum mechanics utilized in the information processing, many important research findings are proposed in recent decades, such as quantum key distribution (QKD) [Bennett and Brassard (1984); Artur (1991)], quantum secure sharing (QSS) [Liu, Chen, Xu et al. (2012); Chen, Tang, Xu et al. (2018); Liu, Xu, Zhang et al. (2019)], quantum key agreement (QKA) [Huang, Su, Liu et al. (2017); Liu, Xu, Yang et al. (2018)], quantum secure direct communication (QSDC) [Liu, Chen, Li et al. (2008); Liu, Chen, Ma et al. (2009); Xu, Chen and Li (2015)], quantum private comparison (QPC) [Liu, Liu, Wang et al. (2013); Liu, Liu, Liu et al. (2014); Liu, Liu, Chen at al. (2014); Liu, Liu, Wang (2014)], quantum sealed-bid auction (QSBA) [Liu, Wang, Yuan et al. (2016); Liu, Wang, Ji et al. (2014)], remote preparation of quantum states [Liu, Chen, Liu et al. (2015); Chen, Sun, Xu

[1] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, 210044, China.

[2] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China.

[3] School of Information Engineering, Jiangsu Maritime Institute, Nanjing, 211100, China.

[4] Department of Electrical and Computer Engineering, Portland State University, Oregon, 97207, USA.

[*] Corresponding Author: Wenjie Liu. Email: wenjiel@163.com.

Received: 04 June 2020; Accepted: 11 July 2020.

CMC. doi:10.32604/cmc.2020.07564

et al. (2017); Qu, Wu, Wang et al. (2017)], quantum steganography [Qu, Cheng, Liu et al. (2019); Qu, Chen, Ji et al. (2018)], delegating quantum computation [Liu, Chen, Ji et al. (2017)], and quantum-based database query scheme [Liu, Gao, Chen et al. (2019); Liu, Xu, Wang et al. (2019)]. On the other hand, quantum parallelism greatly accelerates the computation of some special computational tasks. For example, Deutsch-Jozsa algorithm [Deutsch and Jozsa (1992)] can determine whether the function is constant or balanced with only one query; Grover algorithm [Grover (1996)] has a quadratic speedup to the problem of conducting a search through some unstructured search space; Shor algorithm [Shor (1999)] can factor the prime factor of large numbers in polynomial time (which makes quantum computer easy to crack the current RSA-based cryptosystems); and some quantum machine learning algorithms [Lloyd, Mohseni and Rebentrost (2013); Liu, Gao, Yu et al. (2018); Liu, Gao, Wang et al. (2019); Liu, Chen, Wang et al. (2020); Liu, Li, Zheng et al. (2019)] are also far superior to classical algorithm.

However, the correctness or security verification of the above algorithms or protocols is mostly based on theoretical derivation [Childs, Kothari and Somma (2015), Pan, Yu, Yi et al. (2019)] or experiment simulations [Vandersypen, Steffen, Breyta et al. (2001)]. With the release of quantum cloud computing platform in recent few years, such as D-Wave Leap [Dwave (2018)], IBM Q [IBMquantum (2017)], Alibaba's superconducting quantum computer [Superconducting (2018)], and Tsinghua's NMRCloud Q [Xin, Huang, Liu et al. (2018)], some researchers tried to verify quantum protocols or algorithms on the real quantum computers. In 2017, Gangopadhyay et al. [Gangopadhyay, Manabputra, Behera et al. (2017)] proposed two generalization algorithms based on Deutsch-Jozsa-like algorithm and demonstrated experimental verification of the first algorithm by using IBM 5-qubit device (i.e., 5-qubit IBM Q). And then Srinivasan et al. [Srinivasan, Behera and Panigrahi (2017)] verified Gaussian elimination method for solving system of equations at IBM 5-qubit device. In 2018, Roy et al. [Roy, Behera, Pan et al. (2018)] demonstrated the violation of the entropic noncontextual inequality in a four-level quantum system, by using the 5-qubit IBM Q. As far as we know, most research results of experimental verification are based on IBM 5-qubit device. Besides, their circuit design is directly carried out on the web page, which is only suitable for small-scale quantum circuits (the length of quantum circuit, i.e., maximal number of cascaded quantum gates on a single quantum line, are limited to 80 on the web page). With the increase of the scale of the problem, the feasibility and expansibility of this web mode are relatively poor.

At the end of 2017, IBM released an open-source quantum computation framework, QISKit [QISKit (2018)], which allows the users to implement remote quantum experimental verification of IBM Q through localized python programming. For this kind of localized programming mode based on QISKit, the length of quantum circuit is no longer limited to 80 and the design of quantum functional circuits can be packaged in the form of functions for reusing and expansion. In this paper, we use QISKit and Forest to directly program three representative quantum algorithms, namely Deutsch-Jozsa, Grover and Shor algorithms, and connect the remote real quantum devices (i.e., ibmqx4 and ibmqx5) to verify these algorithms in real quantum computer.

The remaining part of the paper is organized as follows: In Section 2, preliminaries about quantum computation, IBM Q and QISKit are briefly introduced. In Section 3, three representative algorithms, Deutsch-Jozsa, Grover and Shor algorithms, are depicted, and then their implementation circuits are presented. Subsequently, the experimental results of these algorithms are analyzed in detail in graphical form. Finally, Section 4 is dedicated for conclusion.

2 Preliminaries

2.1 Quantum computation

The basic concept of the classical information world is bit. Similarly, quantum computation and quantum information are based on similar concepts: qubit. Classic bit have only one state: either 0 or 1. Corresponding in the qubit, we denote 0 and 1 as: $|0\rangle$ and $|1\rangle$. $|\ \rangle$ is called the Dirac token. The qubit can fall outside $|0\rangle$ and $|1\rangle$. Qubit can be a linear combination of these two states, often referred to as superposition state,

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{1}$$

Here, α and β represent the probability amplitude of $|0\rangle$ and $|1\rangle$. $|0\rangle$ and $|1\rangle$ can be represented by vectors, which is shown in Eq. (2).

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2}$$

Then, the superposition state is represented by a vector as follows:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta\begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3}$$

In a geometric sense, the state of the qubit is required to be normalized to length 1, which means $|\alpha|^2 + |\beta|^2 = 1$.

A quantum computer is built from a quantum circuit containing wires and elementary quantum gates to carry around and manipulate the quantum information. Quantum gates are divided into 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 [Nielsen and Chuang (2002)]. Each valid quantum gate can be represented as a unitary matrix. For visual display, in Tab. 1 below we list some line symbols and matrix representations used in this paper.

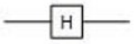
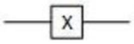
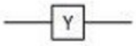
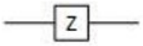
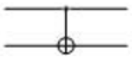
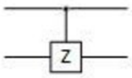
In the actual quantum circuit, we use special line symbol to represent the quantum gate, and a line symbol represents a quantum gate that can manipulate the quantum state, such as,

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{4}$$

If X is used 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,

$$X\begin{bmatrix}\alpha\\ \beta\end{bmatrix}=\begin{bmatrix}\beta\\ \alpha\end{bmatrix}. \tag{5}$$

Table 1: Common quantum gates 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		$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{bmatrix}$

A projective measurement is described by an observable M , a Hermitian operator on the state space of the system being observed. The observable has a spectral decomposition,

$$M=\sum_m mP_m\ , \tag{6}$$

where P_m is the projector onto the eigenspace of M with eigenvalue m . The possible outcomes of the measurement correspond to the eigenvalues m , of the observable. Upon measuring the state $|\psi\rangle$, the probability of getting result m is given by

$$p(m)=\langle\psi\mid P_m|\psi\rangle. \tag{7}$$

Given that outcome m occurred, the state of the quantum system immediately after the measurement is

$$\frac{P_m|\psi\rangle}{\sqrt{p(m)}}. \tag{8}$$

2.2 IBM Q

In 2016, IBM opened the IBM quantum experience prototype 5-qubit device to the public. A year later, they announced the launch of IBM Q [IBMquantum (2017)], the industry's first initiative to build commercially available universal quantum computers for business and science. 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) was 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.

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 ibmqx4 and ibmqx5 are shown in Fig. 1. Generally, two-qubit gates are possible between neighboring qubits that are connected by a super-conduction bus resonator. The IBM Q experience uses the cross-resonance interaction as the basis for the cX -gate. This interaction is stronger when choosing the qubit with higher frequency to be the control qubit, and the lower frequency qubit to be the target, so the frequencies of the qubits determine the direction of the gate.

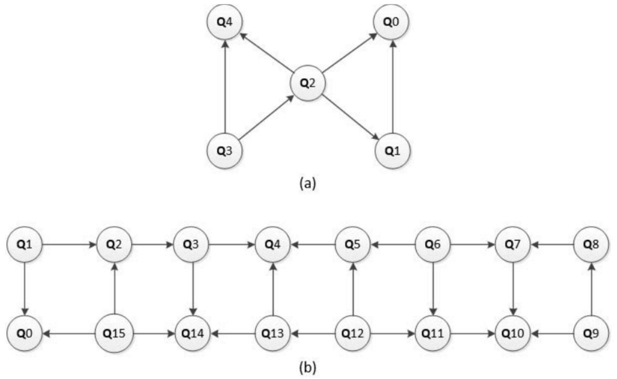


Figure 1: The coupling map picture: (a) ibmqx4 (5 qubits), (b) ibmqx5 (16 qubits). The arrows point from the qubit with higher frequency to that with lower frequency

2.3 QISKit

QISKit (Quantum Information Software Kit) [QISKit (2018)] is a collection of software for working with short depth quantum circuits and building near term applications and experiments on quantum computers. In QISKit, a quantum program is an array of quantum circuits. The program work flow consists of three stages: building, compiling and running.

(1) Building allows you to make different quantum circuits that represent the problem you are solving.

(2) Compiling allows you to rewrite them to run on different backends (simulators or real chips of different quantum volumes, sizes, fidelity, etc.).

(3) Running launches the jobs.

After the jobs have been run, the data are collected. There are methods for putting this data together, depending on the program. In other words, QISKit includes python-based tools for creating, manipulating, visualizing and studying quantum states, tools for characterizing qubits, scripts for batch processing, as well as a compiler to map the desired experiment onto real 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.

3 Quantum algorithms and experiment implementations based on IBM Q

3.1 Deutsch-Jozsa algorithm

3.1.1 Algorithm procedure

The Deutsch-Jozsa problem [Deutsch and Jozsa (1992)] is defined as follows. Consider a function $f(x)$ that takes as input n-bit strings x and returns 0 or 1. The goal is to decide whether f is a constant function that takes the same value $c \in \{0,1\}$ on all inputs x, or a balanced function that takes each value 0 and 1 on exactly half of the inputs. Classically, it requires $2^{n-1}+1$ function evaluations in the worst case. Using the Deutsch-Jozsa algorithm, the question can be answered with just one function evaluation.

The specific steps of the algorithm are depicted in Fig. 2.

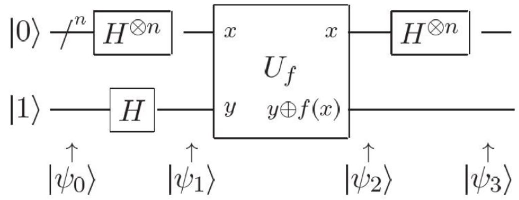


Figure 2: Quantum circuit implementing the general Deutsch-Jozsa algorithm. The wire with a '/' through it represents a set of n qubits

The input state is $|\varphi_0\rangle = |0\rangle^{\otimes n}|1\rangle$. After the Hadamard transform on the query register and the Hadamard gate on the answer register we have

$$|\varphi_1\rangle = \sum_{x\in\{0,1\}^n} \frac{|x\rangle}{\sqrt{2^n}}\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]. \tag{9}$$

Next, the function f is evaluated using $U_f|x,y\rangle \rightarrow |x, y\oplus f(x)\rangle$, giving

$$|\varphi_2\rangle = \sum_x \frac{(-1)^{f(x)}|x\rangle}{\sqrt{2^n}}\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]. \tag{10}$$

Now interfere terms in the superposition using a Hadamard transform on the query register,

$$|\varphi_3\rangle = \sum_z \sum_x \frac{(-1)^{x\cdot z+f(x)}|z\rangle}{2^n}\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]. \tag{11}$$

Note that the amplitude for the state $|0\rangle^{\otimes n}$ is $\sum_x (-1)^{f(x)}/2^n$. In the case where f is constant the amplitude for $|0\rangle^{\otimes n}$ is $+1$ or -1, depending on the constant value $f(x)$ takes. Because $|\varphi_3\rangle$ is of unit length it follows that all the other amplitudes must be 0, and an observation will yield 0s for all qubits in the query register. If f is balanced then the positive and negative contributions to the amplitude for $|0\rangle^{\otimes n}$ cancel, leaving an amplitude of zero, and a measurement must yield a result other than 0 on at least one qubit in the query register. Summarizing, if we measure all 0 s in the query register then the function f is constant; otherwise the function f is balanced.

3.1.2 Experimental implementation and analysis

Suppose $n=3$ and $f(x)=x_0 \oplus x_1x_2$, the implementation circuit of the algorithm can be as shown in Fig. 3.

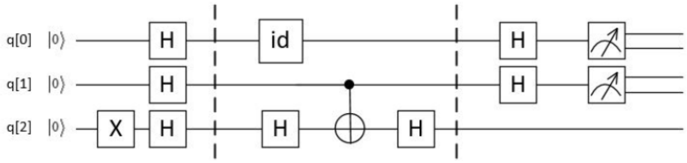
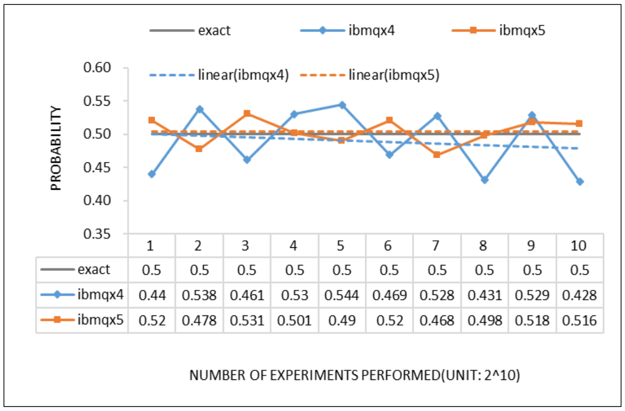


Figure 3: Implementation circuit of Deutsch-Jozsa when f is balance

We use an ID gate such that doing nothing with $|x\rangle$ and a cZ gate $cZ_{1,2}$ such that $cZ_{1,2}|x\rangle=(-1)^{x_1x_2}|x\rangle$. We take $q[0],q[1]$ as the query register and $q[2]$ as the answer register. If $q[0]$=0,$q[1]$=0 , then the function is constant; otherwise the function is balanced. Besides, we program it on python with QISKit which will be connected to ibmqx4 and ibmqx5. Then, the real experimental verification is implemented which can remotely connect the real quantum devices and the code for this circuit is stored on the local computer that exactly can be used again.

Finally, the quantum results of the implementation of Deutsch-Jozsa when f is balance can be shown in Fig. 4. In Fig. 4, $q[0]$ is measured. We run the program once to execute the circuit 1024 (2^{10}) times and calculate the average which is recorded in the table. The horizontal axis in the table indicates the number of times the code is executed (in units of 2^{10}), and the vertical axis represents the average of the experimental results (in units of %). Based on the results, the function is balanced because $q[0]$ and $q[1]$ are not all 0.



	1	2	3	4	5	6	7	8	9	10
exact	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
ibmqx4	0.44	0.538	0.461	0.53	0.544	0.469	0.528	0.431	0.529	0.428
ibmqx5	0.52	0.478	0.531	0.501	0.49	0.52	0.468	0.498	0.518	0.516

Figure 4: Experimental result of Deutsch-Jozsa when f is balance

Nothing to do with $|x\rangle$ just keep it constant. Then we can think of $f(x)$ as a constant function. The circuit design of the algorithm process can be seen in Fig. 5 when f is constant.

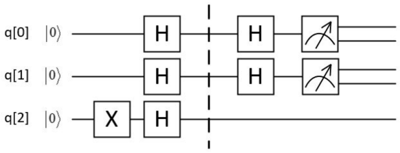
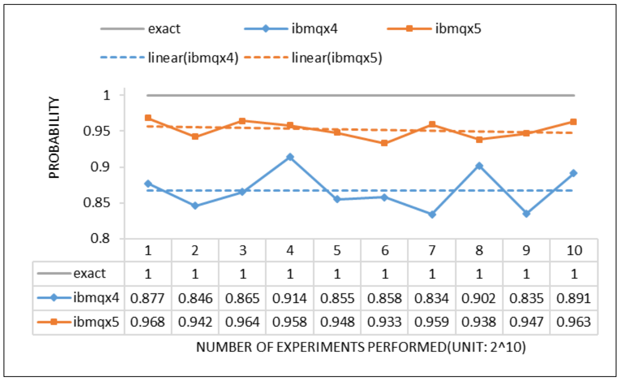


Figure 5: Implementation circuit of Deutsch-Jozsa when f is constant

Quantum results of the implementation of Deutsch-Jozsa when f is constant can be shown in Fig. 6. The exact result to get $|1\rangle$ is 1. Due to noise interference, equipment performance and experimental results will be affected. Then, we can find that the results of ibmqx4 are stable around 0.86, while the results of ibmqx5 are stable around 0.95. Comparing the mean and variance of each group of data, we can find that ibmqx5 has higher stability and computing performance.



	1	2	3	4	5	6	7	8	9	10
exact	1	1	1	1	1	1	1	1	1	1
ibmqx4	0.877	0.846	0.865	0.914	0.855	0.858	0.834	0.902	0.835	0.891
ibmqx5	0.968	0.942	0.964	0.958	0.948	0.933	0.959	0.938	0.947	0.963

Figure 6: Experimental result of Deutsch-Jozsa when f is constant

3.2 Grover algorithm

3.2.1 Algorithm procedure

The problem with the Grover algorithm [Grover (1996)] can be described as follows. Search a target item from N unclassified items. A classic computer is a query until you find the target. On average, if you look up $\frac{N}{2}$ times, the probability of finding it is one half. Based on parallel processing capability of quantum computing, we only need $\sqrt{N}$ times, and the probability of finding it is close to 1 (Grover algorithm). Grover algorithm provides only a quadratic speed up, however, even quadratic speedup is considerable when N is large. Grover quantum search algorithm is based on the basic idea of the initial amplitude superposition of unitary transformation, the repeat application of Grover quantum iterative process is aim to suppress the probability amplitude of the non-target item and enlarge the probability amplitude of the target item to be searched. Finally, in the best case, the target item is searched by the probability of approaching 1. The detailed implementation steps of the Grover quantum search algorithm can be seen in Fig. 7.

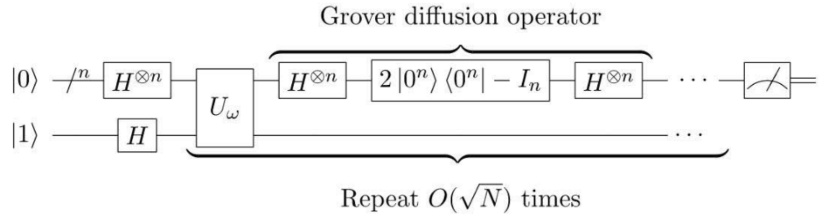


Figure 7: Quantum circuit representation of Grover algorithm

Step 1. Initialize, use H gate to produce a state of equal amplitude, and then apply $H^{\otimes n}|0\rangle^n$ to get $|x\rangle$.

Step 2. Apply the oracle reflection U_w to the state, such that $U_w|x\rangle=(-1)^{f(x)}|x\rangle$ ($f(x)$ is shown in Deutsch-Jozsa algorithm above). This transformation means that the amplitude in front of the target state becomes negative, which in turn means that the average amplitude has been lowered.

Step 3. Apply an additional reflection $U_s=2|x\rangle\langle x|-I$.

Step 4. Repeat Steps 2-3 $\sqrt{N}$ times.

The action of the reflection U_s in the amplitude bar diagram can be understood as a reflection about the average amplitude. Since the average amplitude has been lowered by the first reflection, this transformation boosts the negative amplitude of target state to roughly three times its original value, while it decreases the other amplitudes.

3.2.2 Experimental implementation and analysis

Taking 3-qubit quantum state as an example, we show the search technique of Grover algorithm. To better validate the experiment, the circuit design of the algorithm process can be as shown is shown in the Fig. 8. Besides, the equivalent circuit of Toffoli gate is shown in Fig. 9.

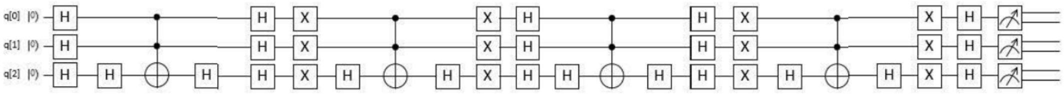


Figure 8: Implementation circuit of Grover algorithm. cZ gate is equivalent to the combination of H gate and ccX gate

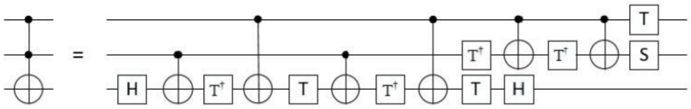
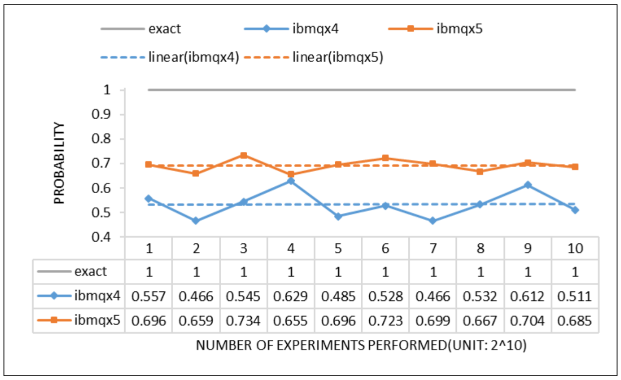


Figure 9: A Toffoli gate implemented as the product of 1-qubit gates and cNOTs

After applying H gate operation on each qubit, the equal amplitude of the quantum state is $\frac{1}{2\sqrt{2}}\left(|000\rangle+|001\rangle+|010\rangle+|011\rangle+|100\rangle+|101\rangle+|110\rangle+|111\rangle\right)$, next, we apply ccZ gate as U_w to specify a target quantum state (the amplitude of the target quantum state is negative), and then we can agree that $|111\rangle$ is the target state. Finally, apply the U_s mentioned above to enlarge the amplitude of the target quantum state $|111\rangle$. Then, the real experimental verification is implemented which can remotely connect the devices and the code for this circuit is stored on the local computer that exactly can be used again.

The results of running the circuit on python with QISKit can be shown in Fig. 10. The exact result to get $|111\rangle$ is 1. Due to noise interference, equipment performance and experimental results will be affected. Then, we can find that the results of ibmqx4 are stable around 0.53, while the results of ibmqx5 are stable around 0.69. Comparing the mean and variance of each group of data, we can find that ibmqx5 has higher stability and computing performance.



	1	2	3	4	5	6	7	8	9	10
exact	1	1	1	1	1	1	1	1	1	1
ibmqx4	0.557	0.466	0.545	0.629	0.485	0.528	0.466	0.532	0.612	0.511
ibmqx5	0.696	0.659	0.734	0.655	0.696	0.723	0.699	0.667	0.704	0.685

Figure 10: Experimental result of Grover algorithm

3.3 Shor algorithm

3.3.1 Algorithm procedure

Shor algorithm [Shor (1994)] is able to factor large numbers efficiently, which has the potential to undermine contemporary encryption. It consists of a quantum order finding algorithm (QOFA) which provids the order for code implementation to return the factors. Recently, Grosshans et al. [Grosshans, Lawson, Morain et al. (2015)] proposed a quantum factoring algorithm which optimized Shor algorithm at factoring safe semiprimes. A semiprime is a product of two primes, hence finding any nontrivial factor of a semiprime amounts to finding its complete prime factorization. An odd prime p is called safe if $(p-1)/2$ is also prime. A safe semiprime is a product of two safe primes. Suppose N is a safe semiprime, then it can also be written as $N = p_1 p_2 = (2q_1+1)(2q_2+1)$, with $q_1 \neq q_2$ and $q_1, q_2 > 2$ (p_1 and p_2 are distinct safe primes greater than 3). Then, the possible multiplicative orders for integers modulo safe semiprimes $N = (2q_1+1)(2q_2+1)$ is shown in Tab. 2.

In number theory, Euler's totient function [Kaliski (2005)] $\varphi(N)$ counts the number of integers x in the range $1 \le x \le n$ when the greatest common divisor $gcd(N,x)$ is equal to 1. According to Miller [Miller (1976)], N can be efficiently factored after knowing both N and $\varphi(N)$. If $N = p_1 p_2$ is a semiprime, then $\varphi(N) = N+1-(p_1 p_2)$. Expand the product $(X-p_1)(X-p_2)$, and we find that p_1 and p_2 are the solution of the equation

$$X^2 - (N+1-\varphi(N)) + N = 0. \tag{12}$$

Table 2: Possible multiplicative order r for integer a modulo safe semiprime N

r	Number of integers $1 \le a \le N$ of order r	Restrictions on a
1	1	$a=1$
2	3	$a = N-1$ and two other $a \ge \sqrt{N+1} > 2$
q_1	$q_1 - 1$	$a \neq 2$
q_2	$q_2 - 1$	$a \neq 2$
$2q_1$	$3(q_1-1)$	$a \neq 2$
$2q_2$	$3(q_2-1)$	$a \neq 2$
$q_1 q_2$	$(q_1-1)(q_2-1)$	
$2q_1 q_2$	$3(q_1-1)(q_2-1)$	

Referring to Tab. 2, there are 8 possibilities for the order of a modulo N. Since the value of a is obviously independent of the number being factored [Smolin, Smith and Vargo (2013)], we set $a=2$ and suppose $ord_N(2)$ is the order of 2 modulo N, then the only remaining possibilities for $ord_N(2)$ are q_1q_2 and $2q_1q_2$. Algorithm 1 shows the process of the quantum factoring algorithm which make some modifications in the classical part of Shor algorithm. Let $d=ord_N(2)$. Set $s=d$ if d is even, and $s=2d$ if d is odd; then $s=2,2q_1,2q_2,or2q_1q_2$. The case $s=2$ is trivial to recognize, since $N>3$. If s is one of the $2q_i$ then $s+1$ is one of the p_i (and $N/(s+1)$ is the other). If s is $2q_1q_2$ then we recover p_1 and p_2 by using $\varphi(N)=2s$ and applying the quadratic formula

$$(X-p_1)(X-p_2)=X^2-(p_1+p_2)X+N=0. \tag{13}$$

Let $p_1+p_2=2t$, then $X=t\pm\sqrt{t^2-N}$. Combining Eq. (12) with Eq. (13), and then we can get $p_1+p_2=2t=N+1-\varphi(N)$, so $t=(N+1-\varphi(N))/2=(N+1)/2-s$. Therefore, the factors of N are $t\pm\sqrt{t^2-N}$, where $t=(N+1)2\ -s$.

3.3.2 Experimental implementation and analysis

QOFA is the main component of this algorithm which provides the order for code implementation. Tab. 3 lists the order r of different safe semiprime N when $a=2$.

Table 3: The order r of 2 modulo N

N	r	$\gcd(N,2^{r/2}-1)$	$\gcd(N,2^{r/2}+1)$
15	4	3	5
21	6	3	7
33	10	3	11
35	12	7	5

Taking $N=21$ as an example, the mathematical calculation process to compute the period of $a=2$ modulo $N=21$ is

$$\begin{aligned}
&2^1 \bmod 21 = 2,\\
&2^2 \bmod 21 = (2\times 2)\bmod 21 = 4,\\
&2^3 \bmod 21 = (2\times 4)\bmod 21 = 8,\\
&2^4 \bmod 21 = (2\times 8)\bmod 21 = 16,\\
&2^5 \bmod 21 = (2\times 16)\bmod 21 = 11,\\
&2^6 \bmod 21 = (2\times 11)\bmod 21 = 1.
\end{aligned} \tag{14}$$

Obviously, we can find that $r=6$. Based on the constant-optimized quantum circuits for the modular multiplication and exponentiation [Igor and Saeedi (2012)], the circuit design of the modular exponentiation with $N = 21$ is shown in the Figs. 11-16.

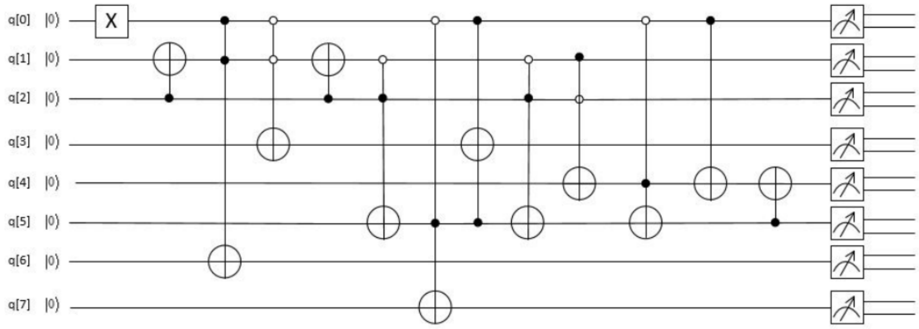


Figure 11: Implementation circuit for computing $2^1 \bmod 21 = 2$, here, the basis vector of 1 is $q[2]q[1]q[0] = |001\rangle$

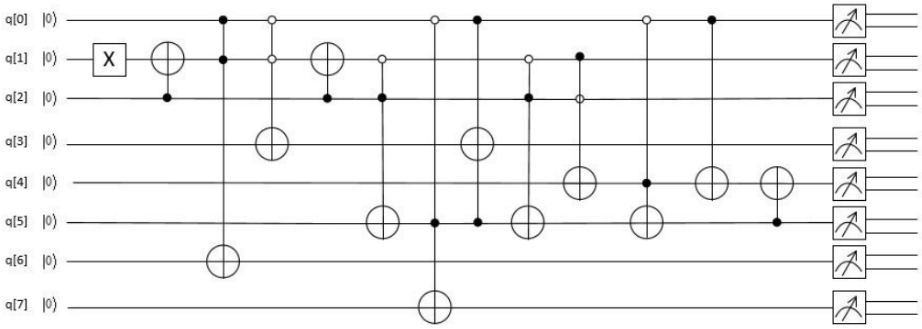


Figure 12: Implementation circuit for computing $2^2 \bmod 21 = 4$, here, the basis vector of 2 is $q[2]q[1]q[0] = |010\rangle$

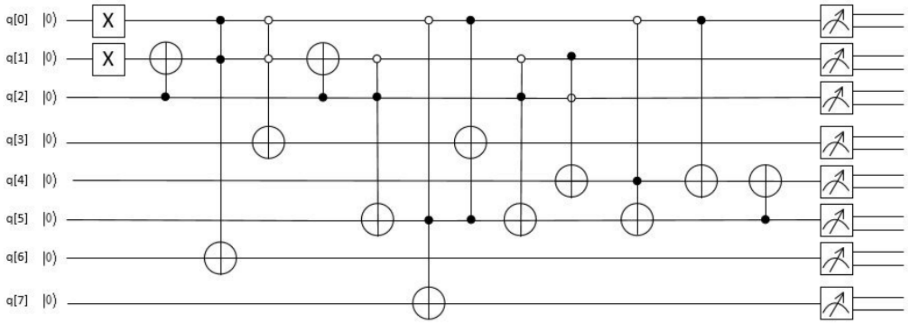


Figure 13: Implementation circuit for computing $2^3 \bmod 21 = 8$, here, the basis vector of 3 is $q[2]q[1]q[0] = |011\rangle$

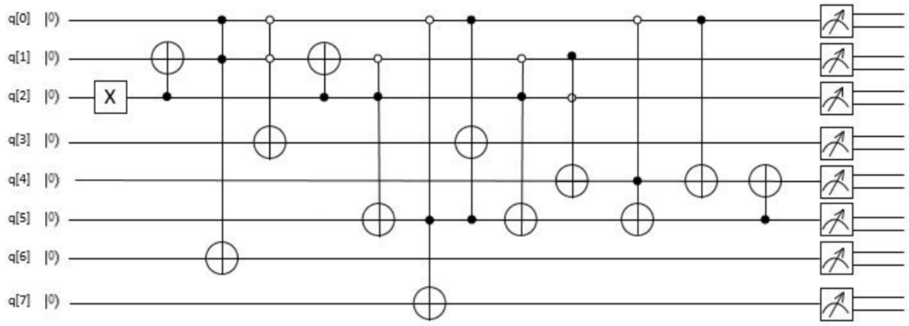


Figure 14: Implementation circuit for computing $2^4 \bmod 21 = 16$, here, the basis vector of 4 is $q[2]q[1]q[0] = |100\rangle$

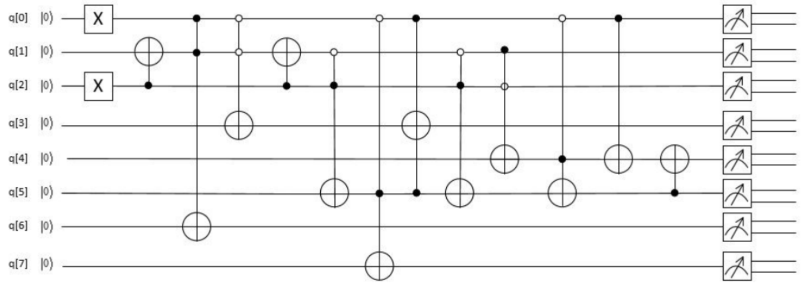


Figure 15: Implementation circuit for computing $2^5 \bmod 21 = 11$, here, the basis vector of 5 is $q[2]q[1]q[0] = |101\rangle$

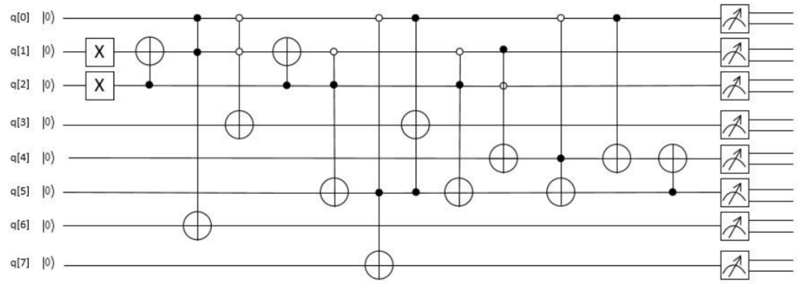


Figure 16: Implementation circuit for computing $2^6 \bmod 21 = 1$, here, the basis vector of 6 is $q[2]q[1]q[0] = |110\rangle$

The result of running the circuit on python with QISKit can be shown in Fig. 17 which corresponds to Eq. (14). The exact result to get $|1\rangle$ is 1. Due to noise interference, equipment performance and experimental results will be affected. Then, we can find that the results of ibmqx5 are stable around 0.92 (ibmqx4 has only 5 qubits which cannot realize the circuit).

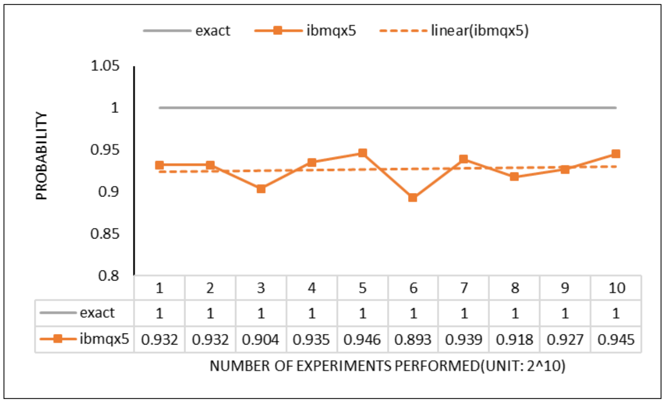


Figure 17: Experimental results of QOFA where $N = 21, a = 2$

4 Conclusions

In this paper, three representative algorithms (i.e., Deutsch-Jozsa, Grover and Shor algorithms) are studied and we design their implementation circuits based on the theoretical research and program the corresponding programs on python with QISKit which realize the remote connection to the real quantum devices (i.e., ibmqx4 and ibmqx5) on IBM Q. These experimental results show the feasibility of these algorithms and serve to assess the functionality and fidelity of these devices. From the results, we can find that the stability and computing performance of ibmqx5 are higher than ibmqx4. Different from the web page experiment mode, this kind of programming call mode, which uses the tool kit API to connect to the devices, 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. Besides, we can customize the composite gates we want on python to achieve more functionality, not limited to the set of gates provided by these platforms. We will continue to study the quantum algorithms, and try to improve the quantum algorithms and design the corresponding circuits to verify the feasibility of the algorithms on the real quantum computer.

Acknowledgement: The authors would like to express heartfelt gratitude to the anonymous reviewers and editor for their comments that improved the quality of this paper. And the support of all the members of the quantum research group of NUIST is especially acknowledged, their professional discussions and advice have helped us a lot.

Funding Statement: This work was supported by the Natural Science Foundation of Jiangsu Province under Grant BK20171458, and in part by the Natural Science Foundation of China under Grant Nos. 61672290 and 61802002, the Natural Science Foundation of Jiangsu Higher Education Institutions of China under Grant No. 19KJB520028, Jiangsu Graduate Scientific Research Innovation Program under Grant No. KYCX20_0978, and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Alibaba (2018): Alibaba's superconducting quantum computer and processor. http://quantumcomputer.ac.cn/.

Bennett, C. H.; Brassard, G. (1984): Quantum cryptography: public key distribution and coin tossing. *Processing of IEEE International Conference on Computers, Systems and Signal Processing*, vol. 175, pp. 8.

Chen, X. B.; Sun, Y. R.; Xu, G. H.; Jia, Y.; Qu, Z. et al. (2017): Controlled bidirectional remote preparation of three-qubit state. *Quantum Information Processing*, vol. 16, no. 10, pp. 244.

Chen, X. B.; Tang, X.; Xu, G.; Dou, Z.; Chen, Y. L. et al. (2018): Cryptanalysis of secret sharing with a single d-level quantum system. *Quantum Information Processing*, vol. 17, no. 9, pp. 225.

Childs, A. M.; Kothari, R.; Somma, R. D. (2015): Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, vol. 46, no. 6, pp. 1920-1950.

Deutsch, D.; Jozsa, R. (1992): Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London, Series A (Mathematical and Physical Sciences)*, vol. 439, pp. 553-558.

D-Wave (2018): D-Wave leap. https://cloud.dwavesys.com/leap.

Ekert Artur, K. (1991): Quantum cryptography based on Bell's theorem. *Physical Review Letters*, vol. 67, no. 6, pp. 661-663.

Gangopadhyay, S.; Manabputra; Behera, B. K.; Panigrahi, P. K. (2018): Generalization and demonstration of an entanglement-based Deutsch-Jozsa-like algorithm using a 5-qubit quantum computer. *Quantum Information Processing*, vol. 17, no. 7, pp. 160.

Grosshans, F.; Lawson, T.; Morain, F.; Smith, B. (2015): Factoring safe semiprimes with a single quantum query. arXiv:1511.04385v3.

Grover, L. K. (1996): A fast quantum mechanical algorithm for database search. *Proceedings of the 28th Annual ACM Symposium Theory of Computing*, pp. 212-219.

Huang, W.; Su, Q.; Liu, B.; He, Y. H.; Fan, F. et al. (2017): Efficient multiparty quantum key agreement with collective detection. *Scientific Reports*, vol. 7, no. 1, pp. 15264.

IBM (2018): IBM quantum computing platform. https://www.research.ibm.com/ibm-q/.

Igor, L.; Saeedi, M. (2012): Constant-optimized quantum circuits for modular multiplication and exponentiation. *Quantum Information Processing*, vol. 12, no. 5, pp. 361-394.

Kaliski, B. (2005): Euler's totient function. In: van Tilborg H.C.A. (eds.), *Encyclopedia of Cryptography and Security*, pp. 206-206.

Liu, W. J.; Chen, H. W.; Li, Z. Q.; Liu, Z. H. (2008): Efficient quantum secure direct communication with authentication. *Chinese Physics Letters*, vol. 25, no. 7, pp. 2354-2357.

Liu, W. J.; Chen, H. W.; Ma, T. H.; Li, Z. Q.; Liu, Z. H. et al. (2009): An efficient deterministic secure quantum communication scheme based on cluster states and identity authentication. *Chinese Physics B*, vol. 18, no. 10, pp. 4105-4109.

Liu, W. J.; Chen, Z. F.; Ji, S.; Wang, H. B.; Zhang, J. (2017): Multi-party semiquantum key agreement with delegating quantum computation. *International Journal of Theoretical Physics*, vol. 56, no. 10, pp. 3164-3174.

Liu, W. J.; Chen, Z. F.; Liu, C.; Zheng, Y. (2015): Improved deterministic N-To-One joint remote preparation of an arbitrary qubit via EPR pairs. *International Journal of Theoretical Physics*, vol. 54, no. 2, pp. 472-483.

Liu, W. J.; Gao, P. P.; Wang, Y. X.; Yu, W. B.; Zhang, M. J. (2019): A unitary weights based one-iteration quantum perceptron algorithm for non-ideal training sets. *IEEE Access*, vol. 7, pp. 36854-36865.

Liu, W. J.; Gao, P. P.; Yu, W. B.; Qu, Z. G.; Yang, C. N. (2018): Quantum relief algorithm. *Quantum Information Processing*, vol. 17, no. 10, pp. 280.

Liu, W. J.; Li, C. T.; Zheng, Y.; Xu, Y.; Xu, Y. S. (2019): Quantum Privacy-preserving price e-negotiation. *International Journal of Theoretical Physics*, vol. 58, no. 10, pp. 3259-3270.

Liu, W. J.; Liu, C.; Chen, H. W.; Li, Z. Q.; Liu, Z. H. (2014): Cryptanalysis and improvement of quantum private comparison protocol based on bell entangled states. *Communications in Theoretical Physics*, vol. 62, no. 2, pp. 210-214.

Liu, W. J.; Liu, C.; Liu, Z. H.; Liu, J. F.; Geng, H. T. (2014): Same initial states attack in yang et al.'s quantum private comparison protocol and the improvement. *International Journal of Theoretical Physics*, vol. 53, no. 1, pp. 271-276.

Liu, W. J.; Liu, C.; Wang, H. B.; Jia, T. T. (2013): Quantum private comparison: a review. *IETE Technical Review*, vol. 30, no. 5, pp. 439-445.

Liu, W. J.; Liu, C.; Wang, H. B.; Liu, J. F.; Wang, F. et al. (2014): Secure quantum private comparison of equality based on asymmetric W State. *International Journal of Theoretical Physics*, vol. 53, no. 6, pp. 1804-1813.

Liu, W. J.; Wang, F.; Ji, S.; Qu, Z. G.; Wang, X. J. (2014): Attacks and improvement of quantum sealed-bid auction with EPR pairs. *Communications in Theoretical Physics*, vol. 61, no. 6, pp. 686-690.

Liu, W. J.; Wang, H. B.; Yuan, G. L.; Xu, Y.; Chen, Z. Y. et al. (2016): Multiparty quantum sealed-bid auction using single photons as message carrier. *Quantum Information Processing*, vol. 15, no. 2, pp. 869-879.

Liu, W. J.; Xu, Y.; Yang, C. N.; Gao, P. P.; Yu, W. B. (2018): An efficient and secure arbitrary n-party quantum key agreement protocol using bell states. *International Journal of Theoretical Physics*, vol. 57, no. 1, pp. 195-207.

Liu, W.; Chen, J.; Wang, Y.; Gao, P.; Lei, Z. et al. (2020): Quantum-based feature selection for multiclassification problem in complex systems with edge computing. *Complexity*, vol. 2020, no. 1, 8216874.

Liu, W.; Xu, Y.; Wang, H.; Lei, Z. (2019): Quantum searchable encryption for cloud data based on full-blind quantum computation. *IEEE Access*, vol. 7, no. 1, pp. 186284-186295.

Liu, W.; Xu, Y.; Zhang, M.; Chen, J.; Yang, C. (2019): A novel quantum visual secret sharing scheme. *IEEE Access*, vol. 7, no., pp. 114374-114384.

Liu, Z. H.; Chen, H. W.; Xu, J.; Liu, W. J.; Li, Z. Q. (2012): High-dimensional deterministic multiparty quantum secret sharing without unitary operations. *Quantum Information Processing*, vol. 11, no. 6, pp. 1785-1795.

Liu, W. J.; Gao, P. P.; Liu, Z. H.; Chen, H. W.; Zhang, M. J. (2019): A quantum-based database query scheme for privacy preservation in cloud environment. *Security and Communication Networks*, vol. 2019, no. 14, pp. 1-14.

Lloyd, S.; Mohseni, M.; Rebentrost, P. (2013): Quantum algorithms for supervised and unsupervised machine learning. arXiv:1307.0411v2.

Miller, G. L. (1976): Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, vol. 13, no. 3, pp. 300-317.

Nielsen, M. A.; Chuang, I. L. (2002): *Quantum Computation and Quantum Information: 10th Anniversary Edition*, pp. 20-30. Cambridge University Press, New York, USA.

Pan, Z. Q.; Yu, W. J.; Yi, X. K.; Khan, A.; Yuan, F. et al. (2019): Recent progress on generative adversarial networks (GANs): a survey. *IEEE Access*, vol. 7, pp. 36322-36333.

QISKit (2018): Quantum information software Kit. https://qiskit.org/.

Qu, Z. G.; Chen, S. Y.; Ji, S.; Ma, S. Y.; Wang, X. J. (2018): Anti-noise bidirectional quantum steganography protocol with large payload. *International Journal of Theoretical Physics*, vol. 57, no. 6, pp. 1-25.

Qu, Z. G.; Cheng, Z. W.; Liu, W. J.; Wang, X. J. (2019): A novel quantum image steganography algorithm based on exploiting modification direction. *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 7981-8001.

Qu, Z. G.; Wu, S. Y.; Wang, M. M.; Sun, L.; Wang, X. J. (2017): Effect of quantum noise on deterministic remote state preparation of an arbitrary two-particle state via various quantum entangled channels. *Quantum Information Processing*, vol. 16, no. 306, pp. 1-25.

Roy, S.; Behera, B. K.; Pan, A. K.; Panigrahi, P. K. (2018): Experimental realization of quantum violation of entropic noncontextual inequality in four dimension using IBM quantum computer. arXiv:1710.10717v4.

Shor, P. W. (1994): Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124-134.

Smith, R. S.; Curtis, M. J.; Zeng, W. J. (2017): A practical quantum instruction set architecture. arXiv:1608.03355v2.

Smolin, J. A.; Smith, G.; Vargo, A. (2013): Oversimplifying quantum factoring. *Nature*, vol. 499, no. 7457, pp. 163-165.

Srinivasan, K.; Behera, B. K.; Panigrahi, P. K. (2017): Solving linear systems of equations by gaussian elimination method using Grover's search algorithm: an IBM quantum experience. arXiv:1801.00778v1.

Vandersypen, L. M. K.; Steffen, M.; Breyta, G.; Yannoni, C. S.; Sherwood M. H. et al. (2001): Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, vol. 414, no. 6866, pp. 883.

Xin, T.; Huang, S. L.; Lu, S. R.; Li, K. R.; Luo, Z. H. et al. (2018): NMRcloudQ: a quantum cloud experience on a nuclear magnetic resonance quantum computer. *Chinese Science Bulletin*, vol. 63, no. 1, pp. 17-23.

Xu, G.; Chen, X. B.; Li, J. (2015): Network coding for quantum cooperative multicast. *Quantum Information Processing*, vol. 14, no. 11, pp. 4297.