Fourier Oracle and Quantum Fourier Transform of Order N in Qubit System

Haikal Khubbi Saputra, Muhammad Taufiqi, Heru Sukamto, Bintoro Anang Subagyo, Agus Purwanto, Lila Yuwana*

Department of Physics, Institut Teknologi Sepuluh Nopember, Jl. Teknik Kimia, Surabaya, 60111, Jawa Timur, Indonesia.

*Corresponding author(s). E-mail(s): lila@physics.its.ac.id;

Abstract

The Quantum Fourier Transform (QFT) in a qubit system is sufficiently flexible for representing a group of order $\boldsymbol{N}$. This study addresses the problem that the dimensionality of qudit used strictly determines the number of group elements the Fourier series can represent. We modify the alternative representation of the Fourier basis as query and phase kickback in the application to solve the Hidden Subgroup Problem. We review algorithms that led the idea into the construction of Fourier Oracle and expanded QFT. We provide an application example of the algorithm for factoring problems using the Qiskit library. The gate decomposition analysis shows that the Fourier Oracle circuit requires $\boldsymbol{O(\mathrm{poly}(\log N))}$ and the expanded QFT requires $\boldsymbol{O(\mathrm{poly}(N))}$ time steps. The expanded QFT allows a full-quantum algorithm yet risks the polynomial-time efficiency.

Keywords: Quantum Fourier Transform, Fourier Oracle, Hidden Subgroup Problem, Quantum Circuit

1 Introduction

The quantum Fourier transform (QFT) is notable in quantum computing. The early algorithm showed the exponential advantage of quantum in the application of a simple case of the QFT. That was Deutsch's result on the QFT of order 2 [1]. On the same approach, Simon presented a problem that was a special case of hidden subgroup problem (HSP) [2]. Simon's inspiration led Shor to extend the case into discrete logarithm and factoring problems, which are crucial problems in cryptography, Shor had achieved the polynomial-time QFT [3]. Coppersmith provided the quantum circuit of QFT in the qubit system, representing the cyclic group of order 2^n [4]. Nowadays, the QFT circuit has become popular through multiple utilization as an algorithm's subroutine, for example, phase estimation [5–10], variational eigensolver [7, 11], signal processing [12–14], and weighted sum [15]. The application of the above algorithm captures the field of quantum cryptography [2, 3, 8], quantum chemistry [6], and especially quantum neural network [12, 15–23]. The instance property of QFT is the control of qubits' phase. Quantum neural network algorithms have prioritized phase-wise over bit-wise computation. The reason is that continuity of phase is the computation resource machine learning has sought. Certain variation models of quantum neural network focuses on the topic of continuos-variable [19–22, 24].

The famous Shor algorithm has evoked numerous studies about modular arithmetic operations in quantum computing. The HSP for commutative group is an immediate result of divisibility in modular arithmetic. Ruiz-Perez observed that most formulations of quantum circuits for modular arithmetic used a logical bit-wise approach [15]. The influence of classical logic circuit familiarity had an impact on these. However, a qubit is not strictly logical but generally rotational. Various studies demonstrated interesting methods to perform arithmetic operations via the phase of qubit [15, 25–29]. This phase-wise modular circuit is explorable further to modify the black box oracle.

The problem in using phase-wise operation is the inflexibility of the group representation with the dimensionality of the qudit system used. The Shor algorithm has succeeded due to its approach of group representation of order N with the QFT matrix. Unfortunately, dimensionality problems arise as soon as the quantum circuit represents the matrix. The matrix size is $N \times N$ with N as a product of two prime numbers, suppose p and q. Circuit representation breaks down the unitary matrix into its tensor product $p \times p \otimes q \times q$. This circuit is constructible with a combination of p and q dimensional qudit [9, 15, 24, 29, 30]. However, this is not applicable with unknown p and q, which is the purpose of the computation itself. Shor's answer to this question was a semi-classical quantum algorithm. The generalization of the Shor algorithm, the Quantum Phase Estimation (QPE), can perform good approximation. This makes the Shor algorithm in any qudit system reliable. The preference for the qubit system remains unchanged even though the qudit system has significant benefits. The underlying reasons cover harder-to-implement universal qudit gates, a lot of interaction inside and outside the channel, characterization of qudit gate, controllability due to selection rule, and less intuitive Bloch sphere of a qudit [24, 31].

This work will show an alternative solution to the dimensionality problem by creating a circuit of a group representation of exact order N as a Fourier basis in a qubit system. The method relates to the modification of two stages in the QPE algorithm. We modify the black box oracle stage that commonly contains bit-wise modular operation to the phase-wise modular operation. The modification relates to a query representation on a Fourier basis, which is thus named the Fourier Oracle. We also modify the phase kickback stage that commonly represents the group of order 2^n by expanding it to the group of order N, which is thus named the expanded QFT. The immediate result has hinted that post-measurement is unnecessary. The Fourier Oracle circuit is polynomial-time. However, the expanded QFT circuit has failed to hold the polynomial-time efficiency. We confirm that modification of the QPE algorithm is applicable to solve the HSP by demonstrating the result of the Qiskit simulation. Potential future works may extend to investigating new quantum circuits.

We begin the topic of discussion by reviewing prior successful algorithms and continue to the novelty of research. The section 2 discusses Deutsch-Jozsa and Quantum Phase Estimation algorithms for solving the HSP. This section also explores the circuit representation and the operation of the two algorithms mentioned. The section 3 will explain modified Quantum Phase Estimation that involves the group representation of order N of the Fourier basis. The section 4 will demonstrate an example of the algorithm's application on a factoring problem. The section 5 contains the conclusion of the paper.

2 Algorithms for Hidden Subgroup Problem

Kitaev identified a pattern between early quantum algorithms that aim to solve the same group theory problem, finding the "hidden" function that hides the subgroup. The term Kitaev used was the Abelian Stabilizer Problem since the group theory problem covers Abelian groups. Jozsa advocated Kitaev's formulation into Abelian and non-Abelian groups. The term Abelian Stabilizer Problem eventually became a Hidden Subgroup Problem. Further studies in this problem are notably remarkable in quantum computing [5, 8, 10, 32–35]. This section will review Deutsch-Jozsa 2.1 and Quantum Phase Estimation 2.2 algorithms that solve HSP with additional comments to introduce phase-wise computation.

The HSP is formally defined as follows. Let $f : G \to X$ be a constant function from a generated group G into the coset X and consider subgroup K such that $f(kg) = f(g)$ for all $k, g \in G$. Suppose black box perform unitary transformation U_f operation such that $U_f |\psi(g_0)\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle |f(g)\rangle$. Find a function f that hides the subgroup K.

Subgroup K does exist if only if $\forall g_1, g_2 \in K$, $g_1 g_2^{-1} \in K$. The method to examine every element group into each other is the subgroup test. In classical computing, a subgroup test will execute $g_1 g_2^{-1}$ independently such that one query will take N times. To test all queries possible, we will need $N \times N$ times to ensure there is no subgroup for a given arbitrary group G. In quantum computing, the subgroup test of one query can be performed simultaneously with all group elements through quantum parallelism. Therefore, the test needs a one-time evaluation instead of N times to ensure a query does not generate a coset corresponding to any subgroup. The test is also applicable to multiple queries. That has a strong correlation with the prime number representation as a query. We will demonstrate the additional trick to employ multiple queries in section 4.

2.1 Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm solved one of the simple problems in HSP, that is Deutsch's Problem. Deutsch's problem statement is to determine whether the given function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is either *balanced* or *constant*. The function is *balanced* if only if the function maps to exactly half zero and one. The function is *constant* if only if the function maps all elements to zero or one.

The choice of the given function f determines whether either subgroup does or doesn't exist. Consider a coset $X = gK$, with element group $g \in G$ and a subgroup K. In the case of a balanced function, the subgroup does exist in the form $K = \mathbb{Z}_2$. However, in the case of a balanced function, the subgroup does exist in the form $K = \mathbb{Z}_1$ which is a trivial subgroup, thus the subgroup doesn't exist.

For the given task to distinguish a function f, the feasible strategy is a query evaluation of the function one by one randomly. If the function has ever returned a different value, then f must be balanced. If the number of queries has reached half of the given n and there are no different number has ever occurred, then we can infer that the function f is constant. For the worst case, the number of queries required is $2^{n-1} + 1$ since the cyclic group $\mathbb{Z}_2^n$ has 2^n element. That is the solution for classical computing. For quantum computing, the ability to evaluate many states allows it to conclude the solution in one query.

The Deutsch-Jozsa algorithm covers three stages of computation, that is creating superposition, black box oracle, and phase kickback. These are the generalization names over past years of studies. The basic technique in creating superposition is to employ the Hadamard gate because it distributes amplitude probabilities equally. The operator of n-Hadamard gates is a well-known Walsh-Hadamard operator:

$$H^{\otimes n} : |j\rangle \to \frac{1}{2^{n/2}} \sum_{l=0}^{2^n-1} (-1)^{l\cdot j} |l\rangle . \tag{1}$$

The Hadamard gate is a simple case of QFT. The phase of the respective qubit represents the group $\mathbb{Z}_2$ in corresponding to the input qubit. The algorithm starts with $n+1$ qubit preparation followed by Walsh-Hadamard. We provide the circuit representation in Fig. 1.

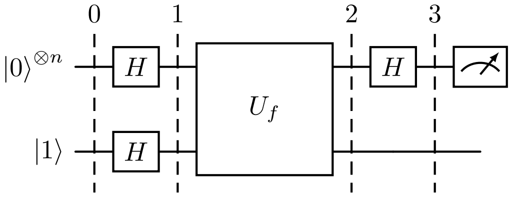

Fig. 1 Deutsch-Jozsa algorithm circuit

$$(H^{\otimes n} \otimes H) : |\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle \to |\psi_1\rangle = \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle \frac{1}{2^{1/2}} \sum_{k=0}^{1} (-1)^{k\cdot 1} |k\rangle . \tag{2}$$

The first register superposition represents group elements as queries. Quantum parallelism then executes the given queries simultaneously inside the black box.

$$U_f : |\psi_1\rangle \to |\psi_2\rangle = \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} (-1)^{f(j)} |j\rangle \frac{1}{2^{1/2}} \sum_{k=0}^{1} (-1)^{k} |k\rangle . \tag{3}$$

The state $|\psi_2\rangle$ form above is the result of phase-wise operation. This state is derivable from the first proposed bit-wise operation. Consider the result of bit-wise operation state $|\psi_2\rangle$ is $\frac{1}{2^{n/2}}\sum_{j=0}^{2^n-1}|j\rangle\,\frac{1}{2^{1/2}}\sum_{k=0}^{1}(-1)^k\,|k\oplus f(j)\rangle$. Since phase and qubit terms in the second register have the term k and correspond to the same modulo 2 operation, we can denote $\frac{1}{2^{n/2}}\sum_{j=0}^{2^n-1}|j\rangle\,\frac{1}{2^{1/2}}\sum_{k=0}^{1}(-1)^{k\oplus f(j)}\,|k\rangle$. Finally, we can move the $f(j)$ phase to the left and obtain the Eq. 3. We now can implement the next stage, the phase kickback, to return the qubit solution of the black box.

$$(H^{\otimes n}\otimes I):|\psi_2\rangle\rightarrow|\psi_3\rangle=\frac{1}{2^n}\sum_{j=0}^{2^n-1}\sum_{l=0}^{2^n-1}(-1)^{f(j)\oplus l\cdot j}\,|l\rangle\,\frac{1}{2^{1/2}}\sum_{k=0}^{1}(-1)^k\,|k\rangle\,. \qquad (4)$$

The indication of a constant function, $f(j)=f(j')$, is the solution maps to zero qubits. Observe for $l=0$, the phase term remains $f(j)$. The solution has to be positive for $f(j)=f(j')=0$ and negative for $f(j)=f(j')=1$. The summation j and l alternates the positive and negative terms, thus annihilating other qubit solutions. The amplitude probability gathers to zero as needed. For the case of a balanced function, we also observe the zero qubit solution $l=0$. Since the balanced function maps to one and zero, this annihilates the amplitude of the zero qubit solution instead. By measuring the first register, the quantum algorithm achieves the conclusion of the given problem with a one-time query evaluation that represented as a black box.

We note that the alternating phase of summation can determine the vanishment of the qubit solution's amplitude probability. This insight is useful for further approaches to phase-wise algorithm development. The phase operation in the Deutsch-Jozsa algorithm is the group multiplication of order 2, which is equivalent to the modulo 2 addition. This operation is obtainable with Hadamard and Z gates. The phase operation in general may extend to the modulo N addition, which is obtainable with the QFT and phase gates.

The second register, which is the least significant qubit, plays a passive role in the algorithm since the measurement on the second register isn't direct. The second or

n-th additional register can benefit to be extra computational space such as modular arithmetic in the Shor algorithm.

2.2 Quantum Phase Estimation

QPE is a generalization of the Shor algorithm. The Shor algorithm is capable of solving specific cases of Abelian HSP, however, the QPE is capable of solving the general case of Abelian HSP. We may adopt the formal problem definition of HSP in the beginning section 2 for a given arbitrary group G in this subsection.

In analogy to the computing stage from the previous subsection, the QPE and Shor algorithms use the black box for controlled unitary operation. The variable j represents a qubit of the first register and controls the operation in the second register. The density operator below describes the general controlled operation U_j.

The density operator of the controlled gate, in general, is a direct sum of the identity I gate and arbitrary unitary the U gate. We will denote the U gate with some index respective to its transformation. We note that the zero preimage of its transformation maps into the I gate. To give an intuitive feeling, we will denote several I gates as U_0.

Consider the multiplication of two-qubit gates with different qubit controls yet operate on the same qubit. The gate is different from a multi-controlled gate. For example, suppose a three-qubit channel and two arbitrarily controlled gates such that both operate on a third qubit, yet the first gate is controlled by the second qubit and the second gate is controlled by the first qubit. We provide the circuit in Fig.2.

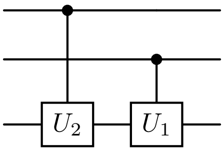

Fig. 2 Multiplication of two-qubit gates with different qubit controls

The multiplication of the density operator will be:

$$\begin{aligned}\rho_1\rho_2 &= (I\otimes|0\rangle\langle0|\otimes U_0 + I\otimes|1\rangle\langle1|\otimes U_1)(|0\rangle\langle0|\otimes I\otimes U_0 + |1\rangle\langle1|\otimes I\otimes U_2)\\ &= |00\rangle\langle00|\otimes U_0 + |01\rangle\langle01|\otimes U_1 + |10\rangle\langle10|\otimes U_2 + |11\rangle\langle11|\otimes U_1U_2. \qquad (5)\end{aligned}$$

This property also conserves in larger gate by induction.

$$\begin{aligned}\prod_{l=0}^{n-1}\rho_l &= \prod_{l=0}^{n-1}(I^{\otimes n-1-l}\otimes|0\rangle\langle0|\otimes I^{\otimes l}\otimes U_0 + I^{\otimes n-1-l}\otimes|1\rangle\langle1|\otimes I^{\otimes l}\otimes U_{2^l})\\ &= \sum_{j_0=0}^{1}\cdots\sum_{j_{n-1}=0}^{1}|j_{n-1}\dots j_0\rangle\langle j_{n-1}\dots j_0|\otimes U_{j_0 2^0}\dots U_{j_{n-1}2^{n-1}}\\ &= \sum_{j=0}^{2^n-1}|j\rangle\langle j|\otimes U_j. \qquad (6)\end{aligned}$$

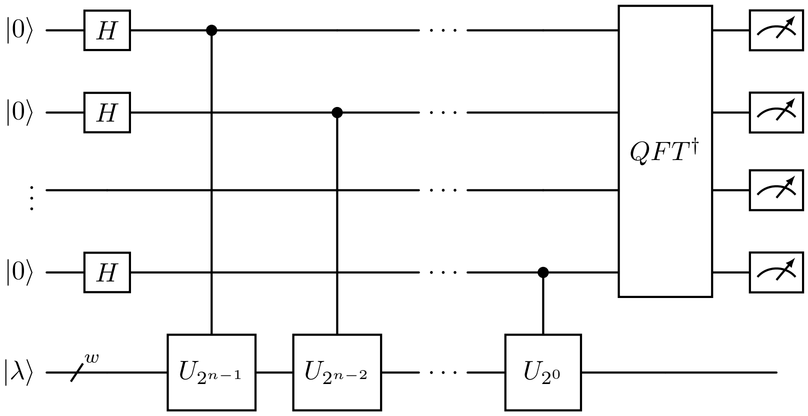

Fig. 3 Quantum Phase Estimation Circuit

Let λ to be arbitary state. The second stage of the circuit will be

$$C_{U_j} : |\psi_1'\rangle \to |\psi_2\rangle = \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} |j\rangle\, U_j \,|\lambda\rangle\,. \tag{7}$$

The Fig. 3 provided a circuit representation of full QPE. Observe that the first stage doesn't change much but the Hadamard gate in the second register. The second stage changed as explained above. We note that the arbitrarily controlled unitary operation frequently appears as the bit-wise modular arithmetic operation, which we may modify to the phase-wise modular operation in the next section. Notice that in the third stage, the QFT appears frequently as the group representation of order 2^n, on which we may modify this to the group representation of exact order N in the next section. Since we have obtained the explanation of QPE's circuit representation, we may exclude the explanation of the post-measurement protocol. We mark that the protocol isn't necessary after the QFT modification.

3 Modified Quantum Phase Estimation

The modified QPE reconstructs the second and third stages of computation. The subsection 3.1 provides construction of the Fourier Oracle. The subsection 3.2 begins with a description of the conventional QFT and continues to the expanded QFT.

3.1 Fourier Oracle

The preparation of the query in classical computing is to apply the NOT-gates to the bits given. Similarly, a query preparation in quantum computing uses a X gate in analogy to the NOT-gate. However, quantum computing has more than one way to perform this. Suppose n-Hadamard gates operate on n-qubits in the first step of computation instead of X gate. Then, let us apply $n \times n$ black box into a given circuit, and n-Hadamard gates afterward. This is similar to the special case of Deutsch-Jozsa in the section 2.1. Query q is preparable by applying the Z gate in place of q's non-zero element inside the black box since $X = HZH$. One may argue additional Hadamards make it less efficient. However, we are pointing out that we successfully prepared the query in another way. This trick of preparing a query later is known as a black box oracle. The choice of applying X gate is bit oracle and the choice of applying Z gate is phase oracle. Application varies combination of different types of gates and placement of black box. Demonstrably, the Z gate is just one case of much non-unique phase representation. We will describe another way of implementing phase oracle called Fourier oracle[1].

Fourier oracle circuit is the direct use of controlled operation explained in section 2.2. We discussed the multiplication of two-qubit gates with different qubit controls. Now consider one qubit control multi-operation.

One qubit control multi-operation is the multiplication of several two-qubit gates. For example, suppose a three-qubit channel and let two arbitrarily controlled gates such that both are controlled by the first qubit, yet the first gate operates on the third qubit and the second gate operates on the second qubit. We provide the circuit in Fig. 4.

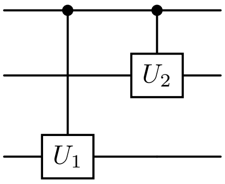

Fig. 4 One Qubit Control Multi-Operation

The multiplication of the density operator will be:

$$\begin{aligned}\rho_2\rho_1 &= (|0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes U_2 \otimes I)(|0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes I \otimes U_1) \\ &= |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes U_2 \otimes U_1. \end{aligned} \tag{8}$$

[1] The Fourier oracle term of this work has a different focus compared with the used term in [36]

This property conserves in larger gate by induction.

$$\begin{aligned}\prod_{l=0}^{n-1}\rho_{n-1-l} &= \prod_{l=0}^{n-1}(\left|0\right\rangle\left\langle 0\right|\otimes I^{\otimes l}\otimes I\otimes I^{\otimes n-1-l}+\left|1\right\rangle\left\langle 1\right|\otimes I^{\otimes l}\otimes U_{2^{n-1-l}}\otimes I^{\otimes n-1-l}) \\ &= \left|0\right\rangle\left\langle 0\right|\otimes I^{\otimes n}+\left|1\right\rangle\left\langle 1\right|\otimes(\bigotimes_{l=0}^{n-1}U_{2^{n-1-l}}). \qquad (9)\end{aligned}$$

Consider using a phase gate in place of an arbitrary unitary gate U_j. Phase gate is 2×2 qubit gate that alters basis one with an arbitrary phase and leaves basis zero unchanged $P:\left|j\right\rangle\rightarrow e^{i(\phi\cdot j)}\left|j\right\rangle$. Here we able to redenote ϕ's range from $[0,2\pi]$ into $[0,N-1]$ such that $\phi=2\pi i\frac{j'}{N}$. Notice that $-j'$ in this operation is congruence with $N-j'$, thus still closed in range $[0,N-1]$. Moreover, we can modify this j' into a product of three variable $j'=xjk$. We provide the circuit in Fig. 5. We suppose the first register has the same size as the second register. Also, the initial computing stage has been replaced with $2n$-Hadamard gates.

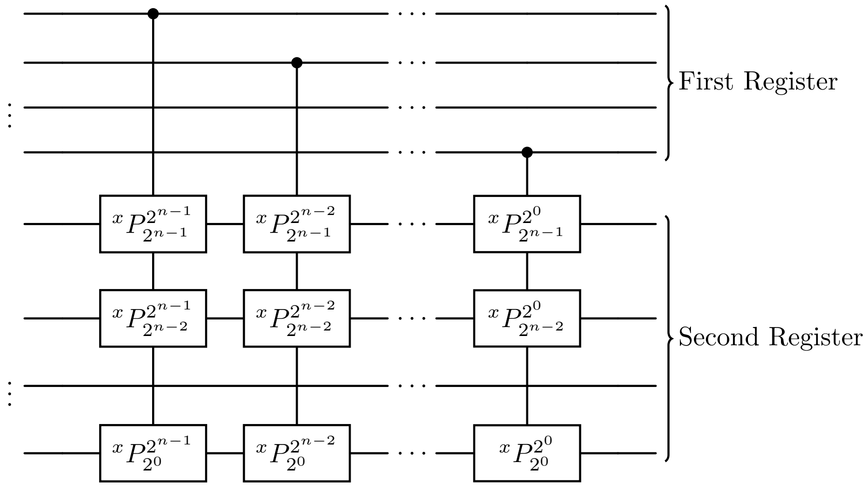
First Register
Second Register
${}^{x}P^{2^{n-1}}_{2^{n-1}}$
${}^{x}P^{2^{n-2}}_{2^{n-1}}$
${}^{x}P^{2^{0}}_{2^{n-1}}$
${}^{x}P^{2^{n-1}}_{2^{n-2}}$
${}^{x}P^{2^{n-2}}_{2^{n-2}}$
${}^{x}P^{2^{0}}_{2^{n-2}}$
${}^{x}P^{2^{n-1}}_{2^{0}}$
${}^{x}P^{2^{n-2}}_{2^{0}}$
${}^{x}P^{2^{0}}_{2^{0}}$

Fig. 5 The Fourier Oracle Circuit

The multiplication of the density operator will be:

$$\begin{aligned}\rho_x &= \sum_{j=0}^{2^n-1} |j\rangle\langle j| \otimes (\bigotimes_{l=0}^{n-1} {}^{x}P_j^{2^{n-1-l}}) \\ &= \textstyle\sum_{j=0}^{2^n-1} |j\rangle\langle j| \otimes \sum_{k_{n-1-l}=0}^{1} e^{2\pi i (k_{n-1-l} 2^{n-1-l} \frac{xj}{N})} |k_{n-1-l}\rangle\langle k_{n-1-l}| \otimes \\ &\qquad\qquad \cdots \otimes \textstyle\sum_{k_0=0}^{1} e^{2\pi i (k_0 2^0 \frac{xj}{N})} |k_0\rangle\langle k_0| \\ &= \sum_{j=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{2\pi i (x\frac{jk}{N})} |jk\rangle\langle jk| \,. \end{aligned} \tag{10}$$

Observe that matrix size is $2^{2n} \times 2^{2n}$ such that combination of $n \times n$ two-qubit controlled-phase gates. The matrix is diagonal such that it is correlated with the phase gate. It embeds phases corresponding to the product of the first register and the second register.

$$C_p^x : |\psi_1\rangle = \frac{1}{2^n}\sum_{j=0}^{2^n-1}\sum_{k=0}^{2^n-1} |j\rangle|k\rangle \to |\psi_2\rangle = \frac{1}{2^n}\sum_{j=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{2\pi i (x\frac{jk}{N})} |j\rangle|k\rangle \,. \tag{11}$$

The variable x in this stage is the query. Superposition simultaneously multiplicates x with every element group in terms of phase. In this stage the group multiplication is complete and the subgroup may appear as a phase. Phase kickback leads this phase solution into a qubit solution.

3.2 Quantum Fourier Transform

The term phase kickback is identical to the inverse QFT operation that inverses the expansion of the qubits. Consider the non-inverse QFT transformation:

$$QFT_{2^n} : |j\rangle \to |\psi_j\rangle = \frac{1}{2^{n/2}}\sum_{l=0}^{2^n-1} e^{2\pi i (lj/2^n)} |l\rangle$$

$$
\begin{aligned}
&= \frac{1}{2^{n/2}} \bigotimes_{k=1}^{n} \sum_{l_k=0}^{1} e^{2\pi i j (l_k 2^{-k})} |l_k\rangle \\
&= \frac{1}{2^{n/2}} \bigotimes_{k=1}^{n} (|0\rangle + e^{2\pi i j 2^{-k}} |1\rangle).
\end{aligned} \tag{12}
$$

The preimage $|j\rangle$ can be expanded to be $|j_1 \dots j_n\rangle$, with the individual $|j_k\rangle$. The Hadamard of the given n contributes the most significant qubit, since $-1 = 2^n/2 = 2^{n-1}$. The phase of the Hadamard is revisable to the corresponding index 2^{-k}. Notice that the variable k above denotes from $k = 1$, not $k = 0$. This variable is consistent with the most significant qubit. Let phase gate to be $P_{2^k} : |j_k\rangle \rightarrow \exp(2\pi i j_k 2^{-k}) |j_k\rangle$. This gives the polynomial for the other qubits other than the most significant. For example, observe the simple case of the QFT of order 2^3 circuit in Fig. 6. The 0 in the phase term is an additional term to denote the polynomial is lesser than zero. Notice that the circuit is the reversed version of the transformation in Eq. 12. The permutation transformation $\pi : |j_k\rangle \rightarrow |j_{n-k}\rangle$ is applicable to swap the sequence of qubits, with different k starts from 0 not 1.

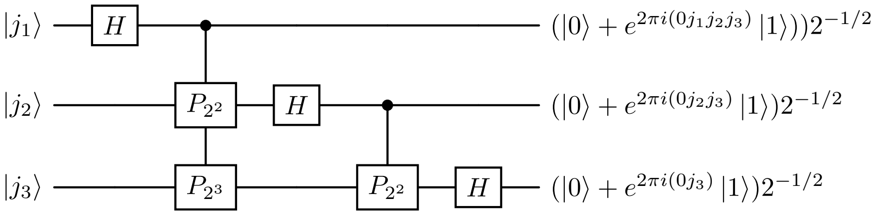
$|j_1\rangle$
$|j_2\rangle$
$|j_3\rangle$
H
P_{2^2}
H
P_{2^3}
P_{2^2}
H
$(|0\rangle + e^{2\pi i(0j_1j_2j_3)} |1\rangle))2^{-1/2}$
$(|0\rangle + e^{2\pi i(0j_2j_3)} |1\rangle)2^{-1/2}$
$(|0\rangle + e^{2\pi i(0j_3)} |1\rangle)2^{-1/2}$

Fig. 6 Simple Quantum Fourier Transform Circuit.

The above transformation is constructible with the combination of Hadamard, controlled phase, and permutation gates. Suppose the density operator denotes the Hadamard placement $\rho_H(k) = H \otimes I^{\otimes k}$. For the controlled phase gate, we gather the one qubit control multioperation from section 3.1. The density operator for controlled phase gate is $\rho_P(k) = |0\rangle\langle 0| \otimes I^{\otimes k} + |1\rangle\langle 1| \otimes (\bigotimes_{k=0}^{n-1} P_{2^k})$. The density operator of permutation gate is $\rho_\pi = \bigotimes_{l=0}^{n-1} \sum_{j_l=0}^{1} |j_{n-1-l}\rangle\langle j_l|$. The above notation denotes the construction of the QFT of order 2^n circuit which is the original work of Coppersmith [4, 10, 30].

$$\rho_{QFT_{2^n}} = \rho_\pi \prod_{k=0}^{n-1} I^{\otimes n-1-k} \otimes \rho_P(k)\rho_H(k). \tag{13}$$

The QFT in quantum circuits conserves dimensionality. In a qubit system, the dimension of the matrix is $2^n \times 2^n$. Meanwhile, in the qudit system, the dimension of the matrix is $d^n \times d^n$. The $N \times N$ QFT and identity matrix are subspaces of the upper nearest $2^n \times 2^n$ matrix. Therefore, the direct sum of $M_{N\times N} \oplus I_{2^n - d}$ is the appropriate unitary matrix for QFT of the order N in the qubit system. The following transformation describes the inverse QFT of order N may evolve qubit $|j\rangle$ if j is less than N or remain the qubit unchanged if j is greater than equal to N. We may call this expanded QFT for discussion.

$$QFT_N^\dagger : \begin{cases} |j\rangle \to \frac{1}{N^{1/2}} \sum_{l=0}^{N-1} e^{-2\pi i (l \frac{j}{N})} |l\rangle \,, & j < N; \\ |j\rangle \to |j\rangle \,, & j \geq N. \end{cases} \tag{14}$$

We will provide a summary of second-stage transformations in the Eq. 11. Let $|\psi_2\rangle$ be separated into the appropriate N part.

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{2^n} \sum_{j=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi i (x \frac{jk}{N})} |j\rangle |k\rangle \\ &= \frac{1}{2^n} \sum_{j=0}^{N-1} \sum_{k=0}^{2^n-1} e^{2\pi i (x \frac{jk}{N})} |j\rangle |k\rangle + \frac{1}{2^n} \sum_{j=N}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi i (x \frac{jk}{N})} |j\rangle |k\rangle \,. \end{aligned} \tag{15}$$

Let the inverse QFT operate in the first register,

$$QFT_N^\dagger \otimes I^{\otimes n} : |\psi_2\rangle \to |\psi_3\rangle \,, \tag{16}$$

with the $|\psi_3\rangle$ is

$$
\begin{aligned}
|\psi_3\rangle &= \frac{1}{2^n}\sum_{j=0}^{N-1}\sum_{k=0}^{2^n-1} e^{2\pi i(x\frac{jk}{N})}\frac{1}{N^{1/2}}\sum_{l=0}^{N-1} e^{-2\pi i(l\frac{j}{N})}\,|l\rangle\,|k\rangle + \frac{1}{2^n}\sum_{j=N}^{2^n-1}\sum_{k=0}^{2^n-1} e^{2\pi i(x\frac{jk}{N})}\,|j\rangle\,|k\rangle \\
&= \frac{1}{2^n N^{1/2}}\sum_{j=0}^{N-1}\sum_{k=0}^{2^n-1}\sum_{l=0}^{N-1} e^{2\pi i(j\frac{(xk-l)}{N})}\,|l\rangle\,|k\rangle + \frac{1}{2^n}\sum_{j=N}^{2^n-1}\sum_{k=0}^{2^n-1} e^{2\pi i(x\frac{jk}{N})}\,|j\rangle\,|k\rangle \\
&= \frac{N^{1/2}}{2^n}\sum_{k=0}^{2^n-1}|xk \mod N\rangle\,|k\rangle + \frac{1}{2^n}\sum_{j=N}^{2^n-1}\sum_{k=0}^{2^n-1} e^{2\pi i(x\frac{jk}{N})}\,|j\rangle\,|k\rangle\,. \qquad (17)
\end{aligned}
$$

Notice that for case $l - xk \neq 0$, the phase will be N-th roots of unity; therefore, its summation is zero. For case $l - xk = 0$, the summation is not zero, and the amplitude probability of the qubit solution rises. Query x determines the set of qubit solutions and its distribution of amplitude probabilities. If query x such that $p \nmid x$, the set of qubit solutions generates a group of p and q such that improper subgroup of N, therefore query is wrong, and the set of qubit solutions is long. If query x such that $p \mid x$, the set of qubit solutions generates a group of p such that the proper subgroup of N, therefore query is correct, and the set of qubit solutions is short. Notice amplitude probabilities in the second term such that $j = N \leq j \leq 2^n - 1$. These probabilities are garbage probabilities that we learn nothing from obtaining them. The expanded QFT transformation isn't decomposable with a non-expanded QFT transformation approach. The polynomial of j is not complete 2^n. The non-zero terms of matrix are $N^2 + 2^{\lceil {}^2\log N \rceil} - N$. Some 2×2 submatrix decompositions remain at least one row or column of all zeros. Thus, the determinant is zero, and the matrix is not invertible. An alternative solution for the problem here is universal algorithms for gate decomposition of the unitary matrix.

The idea of the universal gate decomposition algorithm is to nullify the lower triangular side of a matrix with controlled operation [10, 37, 38]. Upper triangular will follow to be zeros since the unitary matrix is orthonormal. Let $J = \{j_1, j_2, ..., j_N\}$ be a set of matrix index's permutation such that its entries correlate to the permuted form of controlled operation. For example, let the QFT of the order 3 matrix expand to be 4×4 matrix and let $J = \{j_1, j_2, j_3\} = \{0, 2, 1\}$. Suppose $\mu = \mu_1\mu_2\mu_3$ be determinant of matrix U. We may denote matrix entries consistent with its indexes.

$$U = \begin{pmatrix} a_{00} & a_{01} & a_{02} & 0 \\ a_{10} & a_{11} & a_{12} & 0 \\ a_{20} & a_{21} & a_{22} & 0 \\ 0 & 0 & 0 & a_{33} \end{pmatrix}. \tag{18}$$

First step, the goal is $J = (j_3, j_1) = (1, 0) = 0$. Let $u_1 = \sqrt{|a_{20}|^2 + |a_{10}|^2}$. Overline denotes complex conjugate.

$$U_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \overline{\mu}_1 \frac{\overline{a}_{20}}{u_1} & -\overline{\mu}_1 \frac{\overline{a}_{10}}{u_1} & 0 \\ 0 & \frac{\overline{a}_{10}}{u_1} & \frac{\overline{a}_{20}}{u_1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, U_1 U = \begin{pmatrix} a_{00} & a_{01} & a_{02} & 0 \\ 0 & a'_{11} & a'_{12} & 0 \\ u_1 & a'_{21} & a'_{22} & 0 \\ 0 & 0 & 0 & a_{33} \end{pmatrix}. \tag{19}$$

Second step, the goal is $J = (j_2, j_1) = (2, 0) = 0$. Moreover, lastly, the third step's goal is $J = (j_3, j_2) = (1, 2) = 0$.

$$\begin{gathered} U_2 = \begin{pmatrix} \overline{a}_{00} & u_1 & 0 & 0 \\ -\overline{\mu}_2 u_1 & \overline{\mu}_2 \overline{a}_{00} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, U_2 U_1 U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & a'_{11} & a'_{12} & 0 \\ 0 & a''_{21} & a''_{22} & 0 \\ 0 & 0 & 0 & a_{33} \end{pmatrix}; \\ U_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \overline{\mu}_3 \overline{a}''_{22} & -\overline{\mu}_3 \overline{a}'_{12} & 0 \\ 0 & \overline{a}'_{12} & \overline{a}''_{22} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad U_3 U_2 U_1 U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \end{gathered} \tag{20}$$

Moving the unitary gates into the right-hand side, we have accomplished a set of permuted controlled operations.

$$U = U_1^\dagger U_2^\dagger U_3^\dagger.$$

Since the matrix's vector is symmetry, if the diagonal column vector is equal to one, then another entry is equal to zero, either the column or row vector. Every last step nullifying column is equal to one since it is a sum of the absolute value of all vector entries. For example, in the zeroth column, the last step is $|a_{00}|^2 + |a_{10}|^2 + |a_{20}|^2 = 1$. Consequently, the other zeroth rows must be zero.

The non-zero terms of the lower triangular expanded QFT matrix are part of the QFT submatrix. Therefore, the algorithm may decompose $N(N-1)/2$ quantum gates at worst. Compare to the non-expanded QFT, it has only required $n(n-1)/2$ quantum gates such that $n = \lceil {}^2\log N \rceil$. This comparison is a huge difference since the algorithm results in exponential time. The polynomial time result of non-expanded QFT doesn't hold in the expanded QFT.

We have explained how to apply the Fourier oracle and the expanded QFT circuit. This preparation is enough to solve the HSP problem.

4 Application on Factoring

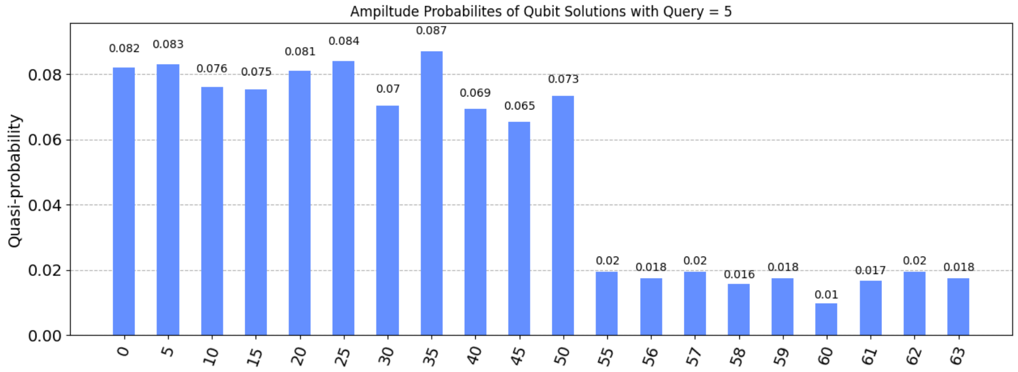

Fig. 7 Qiskit Simulation Histogram of Qubit Solutions with Query = 5

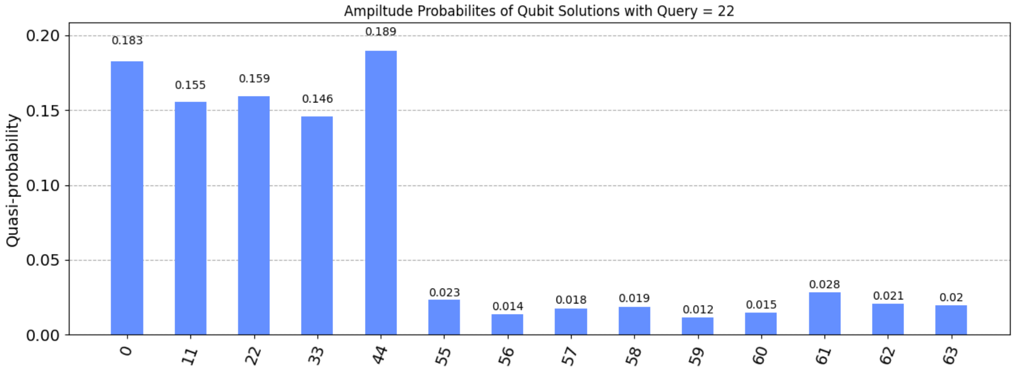

Fig. 8 Qiskit Simulation Histogram of Qubit Solutions with Query = 22

The factoring problem is one example of Abelian HSP and an influential problem in cryptography. The factoring problem is equivalent to the period-finding and order-finding problem [10]. The problem is to determine the prime factors of the given composite number N. We note that this problem is different from Deutsch-Jozsa since there's no case of trivial subgroup. The subgroup does exist for sure with the given composite number N. The foundation of cryptography uses big composite numbers. However, we provide an example of small composite numbers to give more intuition about how the algorithm works. We use IBM Quantum Computing software, Qiskit, with the simulator type "qasm_simulator" in a classical computer. Finally, we will demonstrate a qiskit simulation result of the Modified QPE algorithm to solve the factoring problem.

Suppose the given composite number is 55, which prime factors are 5 and 11. These prime factors are the group and subgroup generators. Now, consider the number between 0 and 54. These numbers are the complete group elements. The numbers of multiple 5 are 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50. These numbers are the subgroup generated by the generator 5. The numbers of multiple 11 are 11, 22, 33, and 44. These numbers are the subgroup generated by the generator 11. We can also refer to the multiple numbers of 5 and 11 as the correct query space. The simulation outcomes

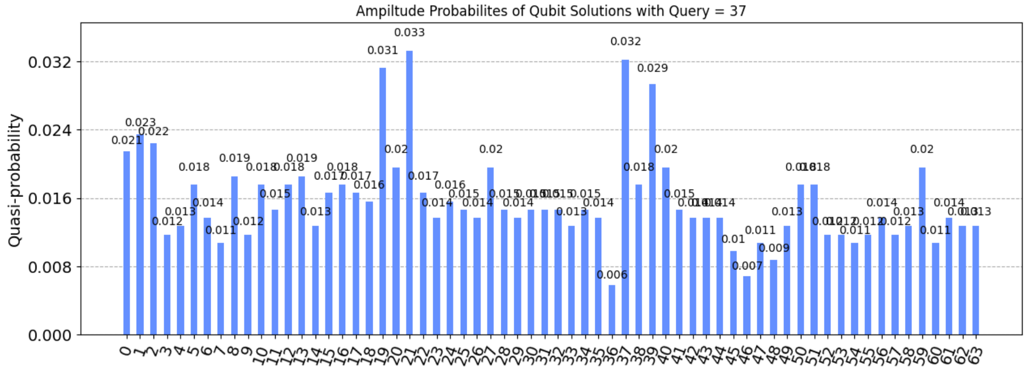

Fig. 9 Qiskit Simulation Histogram of Qubit Solutions with Query = 37

in Fig. 7 and Fig. 8 contain the information that the query is correct. The algorithm will return the full set of multiple 5s if only we apply the query that is also a multiple of 5. The same thing works on the multiple numbers of 11. The other query number, except the 0, will return a full set from 0 to 63. The simulation outcomes in Fig. 9 contain the information that the query is wrong. Observe that numbers from 55 to 63 always appear in Fig. 7, Fig. 8, and Fig. 9, these are garbage probabilities that we learn nothing from obtaining it.

The multiple principle is applicable to employ numerous queries. We can observe that the key to obtaining the correct query is the prime number. The prime numbers under 55 are 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, and 53. We may consider only the first one over three fractions of the above prime list numbers. The remainder primes are 2, 3, 5, 7, 11, and 13. Despite this being a simple procedure, it is quite helpful in eliminating a large list of potential prime factors. We may examine a query with a composite number higher than 55. Suppose we choose the query 60, the amplitude probabilities of Fig. 10 and Fig. 7 are different although the 60 modulo 55 is 5. The number 60 has prime factors of 2, 3, and 5, thus letting the outcomes of the algorithm also change. This behavior is a bit different than the normal QFT matrix.

We also note that since the query representation is a global multiplier of the phase, the greater query doesn't require the greater qubits.

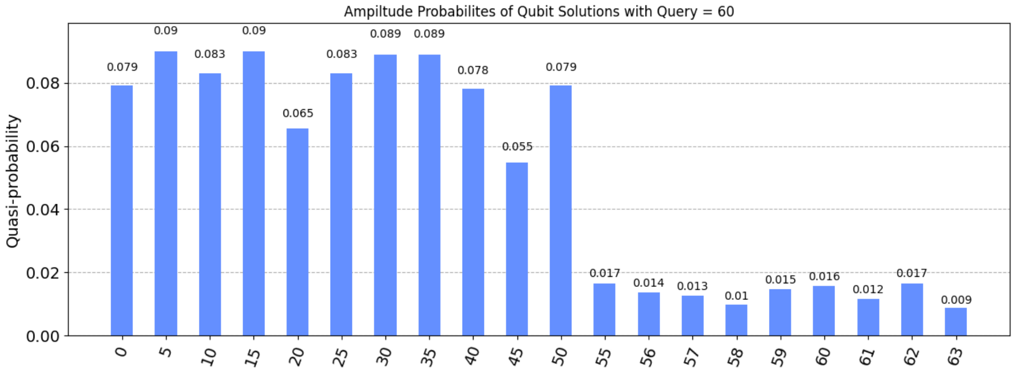

Fig. 10 Qiskit Simulation Histogram of Qubit Solutions with Query = 60

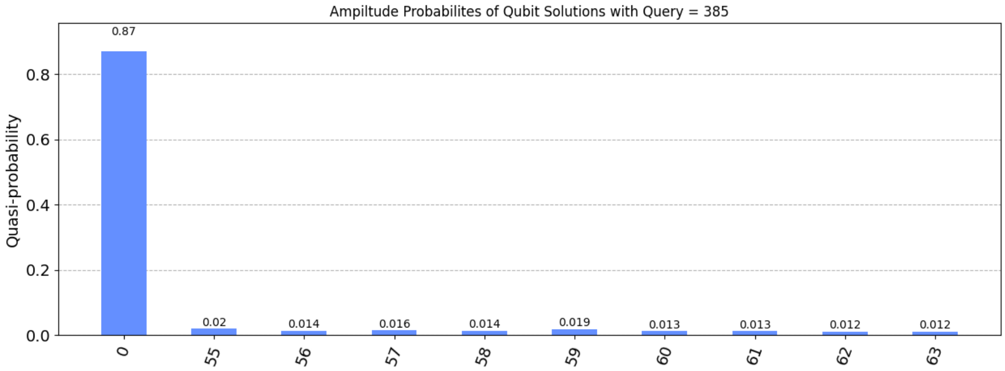

Fig. 11 Qiskit Simulation Histogram of Qubit Solutions with Query = 385

The trick to employ multiple queries is to cluster prime numbers as composite numbers. There are three possible outcomes of the queries. The first is the query doesn't share the prime factor as represented in Fig. 9. The second is the query does share only one prime factor as represented in Fig. 7, Fig 8, and Fig. 10. The third is the query does share both prime factors as represented in the Fig. 11. If the chosen query appears to be the first case, we may move to the other query. If the chosen query appears to be the third case, we may break down the prime factor of it. The algorithm reaches the conclusion if the chosen query appears to be the second case. We note that the whole protocol of factoring with the modified QPE algorithm may involve measurements as a subroutine. However, the modified QPE algorithm doesn't include the "approximation" post-measurement protocol as in the Shor algorithm. The simulation shows that the modified QPE performs the full-quantum algorithm successfully.

5 Conclusion

In this work, we show that the proposed Fourier oracle and expanded QFT have successfully achieved the group representation of Fourier basis of order N. The construction of the Fourier oracle circuit makes use of controlled phase gates similar to the non-expanded QFT with $O(\mathbf{poly}(\log N))$. The universal gate decomposition generates the expanded QFT with $O(\mathbf{poly}(N))$ obtained. The qiskit simulation shows that the algorithm is capable of solving one of the HSP problems, that is the factoring problem. Future work may contain polynomial decomposition of the expanded QFT, another arithmetic operation of Fourier oracle, use of them as a solution on specific HSP, and application on quantum signal processing.

References

[1] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

[2] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[3] Don Coppersmith. An approximate fourier transform useful in quantum factoring. *arXiv preprint quant-ph/0201067*, 2002.

[4] Daniel R Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483, 1997.

[5] Adetokunbo Adedoyin, John Ambrosiano, Petr Anisimov, William Casper, Gopinath Chennupati, Carleton Coffrin, Hristo Djidjev, David Gunter, Satish Karra, Nathan Lemons, et al. Quantum algorithm implementations for beginners. *arXiv preprint arXiv:1804.03719*, 2018.

[6] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews*, 120(22):12685–12717, 2020.

[7] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H Booth, et al. The variational quantum eigensolver: a review of methods and best practices. *Physics Reports*, 986:1–128, 2022.

[8] Chien-Hung Cho, Chih-Yu Chen, Kuo-Chin Chen, Tsung-Wei Huang, Ming-Chien Hsu, Ning-Ping Cao, Bei Zeng, Seng-Ghee Tan, and Ching-Ray Chang. Quantum computation: Algorithms and applications. *Chinese Journal of Physics*, 72:248–269, 2021.

[9] Ye Cao, Shi-Guo Peng, Chao Zheng, and Gui-Lu Long. Quantum fourier transform and phase estimation in qudit system. *Communications in Theoretical Physics*, 55(5):790, 2011.

[10] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[11] Xinglan Zhang and Feng Zhang. Variational quantum computation integer factorization algorithm. *International Journal of Theoretical Physics*, 62(11):245, 2023.

[12] Kosuke Mitarai, Masahiro Kitagawa, and Keisuke Fujii. Quantum analog-digital conversion. *Physical Review A*, 99(1):012301, 2019.

[13] Haiting Yin, Dayong Lu, and Rui Zhang. Quantum windowed fourier transform and its application to quantum signal processing. *International Journal of Theoretical Physics*, 60:3896–3918, 2021.

[14] Ola Al-Ta'ani1a, Ali Mohammad Alqudah, and Manal Al-Bzoor. Implementation and analysis of quantum fourier transform in image processing. *Jordan J. Electrical Eng*, 5:11–26, 2019.

[15] Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. Quantum arithmetic with the quantum fourier transform. *Quantum Information Processing*, 16:1–14, 2017.

[16] Marco AS Trindade, Vinícius NA Lula-Rocha, and S Floquet. Clifford algebras, quantum neural networks and generalized quantum fourier transform. *Advances in Applied Clifford Algebras*, 33(3):38, 2023.

[17] Mingchao Guo, Hailing Liu, Yongmei Li, Wenmin Li, Fei Gao, Sujuan Qin, and Qiaoyan Wen. Quantum algorithms for anomaly detection using amplitude estimation. *Physica A: Statistical Mechanics and its Applications*, 604:127936, 2022.

[18] Yao Zhang and Qiang Ni. Recent advances in quantum machine learning. *Quantum Engineering*, 2(1):e34, 2020.

[19] Nathan Killoran, Thomas R Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. Continuous-variable quantum neural networks. *Physical Review Research*, 1(3):033063, 2019.

[20] James Stokes, Saibal De, Shravan Veerapaneni, and Giuseppe Carleo. Continuous-variable neural network quantum states and the quantum rotor model. *Quantum Machine Intelligence*, 5(1):12, 2023.

[21] Jasvith Raj Basani and Aranya B Bhattacherjee. Continuous-variable deep quantum neural networks for flexible learning of structured classical information. *arXiv*

preprint arXiv:2006.10927, 2020.

[22] Ulrik L Andersen, Jonas S Neergaard-Nielsen, Peter Van Loock, and Akira Furusawa. Hybrid discrete-and continuous-variable quantum information. *Nature Physics*, 11(9):713–719, 2015.

[23] Do Ngoc Diep. Some quantum neural networks. *International Journal of Theoretical Physics*, 59(4):1179–1187, 2020.

[24] Yuchen Wang, Zixuan Hu, Barry C Sanders, and Sabre Kais. Qudits and high-dimensional quantum computing. *Frontiers in Physics*, 8:589504, 2020.

[25] AnQi Zhang, XueMei Wang, and ShengMei Zhao. The multiplier based on quantum fourier transform. *CCF Transactions on High Performance Computing*, 2: 221–227, 2020.

[26] Joseph L Pachuau, Arnab Roy, and Anish Kumar Saha. Integer numeric multiplication using quantum fourier transform. *Quantum Studies: Mathematics and Foundations*, 9(1):155–164, 2022.

[27] Selçuk Çakmak, Murat Kurt, and Azmi Gençten. Quantum fourier transform-based arithmetic logic unit on a quantum processor. *Annalen der Physik*, page 2300457, 2023.

[28] Junhong Nie, Qinlin Zhu, Meng Li, and Xiaoming Sun. Quantum circuit design for integer multiplication based on schönhage-strassen algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.

[29] Archimedes Pavlidis and Emmanuel Floratos. Quantum-fourier-transform-based quantum arithmetic with qudits. *Physical Review A*, 103(3):032417, 2021.

[30] Daan Camps, Roel Van Beeumen, and Chao Yang. Quantum fourier transform revisited. *Numerical Linear Algebra with Applications*, 28(1):e2331, 2021.

[31] Hengyun Zhou, Haoyang Gao, Nathaniel T Leitao, Oksana Makarova, Iris Cong, Alexander M Douglas, Leigh S Martin, and Mikhail D Lukin. Robust hamiltonian engineering for interacting qudit systems. *arXiv preprint arXiv:2305.09757*, 2023.

[32] A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995.

[33] Richard Jozsa. Quantum algorithms and the fourier transform. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):323–337, 1998.

[34] Katalin Friedl, Gábor Ivanyos, Frédéric Magniez, Miklos Santha, and Pranab Sen. Hidden translation and translating coset in quantum computing. *SIAM Journal on Computing*, 43(1):1–24, 2014.

[35] Jingwen Suo, Licheng Wang, Sijia Yang, Wenjie Zheng, and Jiankang Zhang. Quantum algorithms for typical hard problems: a perspective of cryptanalysis. *Quantum Information Processing*, 19:1–26, 2020.

[36] Emmanuel Amiot, Thomas Noll, Moreno Andreatta, and Carlos Agon. Fourier oracles for computer-aided improvisation. In *ICMC 2006*, pages 1–1, 2006.

[37] Chi-Kwong Li, Rebecca Roberts, and Xiaoyan Yin. Decomposition of unitary matrices and quantum gates. *International Journal of Quantum Information*, 11 (01):1350015, 2013.

[38] Mikio Nakahara and Tetsuo Ohmi. *Quantum computing: from linear algebra to physical realizations*. CRC press, 2008.