Quantum functional oracles

Quantum functional oracles

Jinsoo Kim[1], Soojoon Lee[2] and Dong Pyo Chi[2]

[1] School of Electrical Engineering and Computer Science, Seoul National University,
Seoul 151-744, Korea
[2] School of Mathematical Sciences, Seoul National University, Seoul 151-742, Korea

E-mail: jkim@ee.snu.ac.kr, level@math.snu.ac.kr and dpchi@math.snu.ac.kr

Abstract
The limitation on the size of quantum computers makes it important to reuse
qubits for auxiliary registers even though they are entangled with others and are
occupied by other computational processes. We construct a quantum algorithm
that performs the functional phase rotation, which is the generalized form of
the conventional conditional phase transforms, using the functional evaluation
oracle. The constructed algorithm works without any *a priori* knowledge of
the state of an auxiliary register at the beginning and it recovers the initial state
of an auxiliary register at the end. This provides ample scope to choose qubits
for auxiliary registers at will.

PACS numbers: 03.67.Lx, 03.65.Ta

1. Introduction

One of the most powerful features of quantum computation is quantum parallelism. The
superposition principle of quantum mechanics makes it feasible to prepare a quantum register
in a coherent superposition of all possible inputs. Through a query to a quantum oracle
evaluating a given function, we can encode the information on the function into the basis
states or the phases. It may well be possible to distill from this state a certain global property
of the function, thus exploiting quantum parallelism. For a given function f there are two
ways to encode the information on f into qubits. One way is to use a functional evaluation
oracle computing $|x, y\rangle \mapsto |x, y + f(x)\rangle$, in which the information on f is encoded into the
basis states. The first quantum register (the control register) contains input states which we
wish to interfere, and the second quantum register (the auxiliary register) is usually used to
draw relative phase changes in the first register. The other way is to make use of the functional
phase rotation $|x\rangle \mapsto e^{i\theta f(x)}|x\rangle$ for an arbitrary real constant θ, in which the information on f
is encoded into the phases. On occasions, the functional evaluation oracle is merely used to
induce the effect of the functional phase rotation on the control register. In this case, we can

adopt the functional phase-rotation oracle instead of the functional evaluation oracle and we can avoid the necessity of an auxiliary register.

Deutsch and Jozsa [1, 2] have presented a simple promise problem to determine whether a Boolean function $f : \mathbb{Z}_N \to \mathbb{Z}_2$ is either constant or balanced. They have shown that this can be solved efficiently without error on a quantum computer while it requires an exhaustive search to solve deterministically without error in a classical setting. In subsequent work, Cleve *et al* [3] have generalized their algorithm to distinguish between constant and evenly-balanced functions. The key of these algorithms is the π-rotation of phases controlled by the query result of the functional evaluation oracle computing $|x, y\rangle \mapsto |x, y + f(x)\rangle$. The refined Deutsch–Jozsa algorithm [4], which is the description of the original Deutsch–Jozsa algorithm by an oracle of the form $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$, removed the necessity of the auxiliary register. Recently, Chi *et al* [5] generalized these algorithms to distinguish between constant and evenly-distributed functions by exploiting summations of the roots of unity. They employed an oracle computing $|x\rangle \mapsto e^{2\pi i f(x)/M}|x\rangle$ for a given function $f : \mathbb{Z}_N \to \mathbb{Z}_M$. Grover has constructed a quantum algorithm that can find a particular item in expected time $O(\sqrt{N})$ when an unstructured list of N items is given [6–8]. This algorithm relies on the conditional phase transform $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ where f is the Boolean function computed by an oracle. Chi and Kim [9] have generalized this search algorithm and have shown that a quantum computer can search a database by a single query when the number of solutions is equal to, or more than, a quarter. Their algorithm makes use of the conditional γ-phase transform $|x\rangle \mapsto e^{i\gamma f(x)}|x\rangle$ where γ is a real constant. By combining the ZQP algorithm of Simon [10] and Grover's quantum search algorithm, Brassard and Høyer [11] have shown that Simon's problem can be solved on a quantum computer in worst-case polynomial time; thus, it is in the QP class.

For a function $f : \mathbb{Z}_N \to \mathbb{Z}_M$ the conditional phase transform can be generalized to the functional phase rotation $R_{\xi, f} : |x\rangle \mapsto \omega^{\xi f(x)}|x\rangle$, where $\omega = e^{2\pi i/M}$ is a primitive Mth root of unity and ξ is an arbitrary constant in $\mathbb{Z}_M$. In this paper, we construct a quantum algorithm that implements the functional phase rotation $R_{\xi, f}$ using the functional evaluation operator $U_f : |x, y\rangle \mapsto |x, y + f(x)\rangle$. While most quantum algorithms employing the functional evaluation operator require the initialization of an auxiliary register to start with, our algorithm requires no initialization of an auxiliary register.

There have been a few results on the initialization problem in quantum computation. Biron *et al* [12, 13] have found that Grover's search algorithm [6] is robust against modest noise in the amplitude initialization procedure. Also, Carlini and Hosoya [14] have obtained a similar result for the generalized search algorithm constructed by Chi and Kim [9] and for the counting algorithm presented by Brassard *et al* [15]. In [16] it has been shown that one pure qubit and a supply of maximally mixed qubits are sufficient to implement efficiently the quantum factoring algorithm of Shor [17] by the conjunction of the phase estimation technique [18] (see also [3]) and the semiclassical Fourier transform [19]. Recently, Chi *et al* [5] constructed a quantum algorithm that can distinguish between constant and evenly-balanced functions without the initialization of an auxiliary register.

If quantum algorithms can avoid initializing auxiliary registers at the beginning and moreover preserve the initial states of auxiliary registers at the end, we can reuse qubits for auxiliary registers even though they are entangled with others and are occupied by other computational processes. We do this by constructing a quantum algorithm that performs the functional phase rotation and retrieves the initial state of the auxiliary register after its application. Hence the auxiliary register in the constructed algorithm can consist of any qubits collected from any other registers. We also demonstrate that to realize the initialization-free functional phase rotation at least two operations dependent on the given function are necessary.

Thus, the presented algorithm is optimal in that it involves only two functional evaluations. Of course, if any kind of initialization is involved, a single functional evaluation is sufficient.

The rest of this paper is organized as follows. In section 2 we investigate the initialization-free functional phase rotation restricted on an auxiliary register and we construct a quantum algorithm for this localized operation. This facilitates the construction of the initialization-free functional phase rotation from the functional evaluation operator, which is presented in section 3. Finally, we draw a conclusion in section 4.

2. Localized operation

In order for an algorithm performing the functional phase rotation to leave the auxiliary register intact, the overall procedure should operate on an auxiliary register as the identity operator. In this section, we look at the phase changes in the control register resulting from the initialization-free functional phase rotation as those in the auxiliary register and we investigate their effect on the auxiliary register. We restrict the initialization-free functional phase rotation on the auxiliary register and then, localizing the functional evaluation operator on the auxiliary register, we construct a quantum algorithm that performs the localized initialization-free functional phase rotation.

Let $f : \mathbb{Z}_N \to \mathbb{Z}_M$ be given. The unitary operation, which realizes the functional phase rotation $R_{\xi,f} : |x\rangle \mapsto \omega^{\xi f(x)}|x\rangle$ on the control register and leaves the auxiliary register untouched, can be written as $R_{\xi,f} \otimes I : |x, y\rangle \mapsto \omega^{\xi f(x)}|x, y\rangle$ where ξ is an arbitrary constant in $\mathbb{Z}_M$ and $\omega = e^{2\pi i/M}$ is a primitive Mth root of unity. The phase changes in the control register caused by $R_{\xi,f}$ can be considered as the phase changes in the auxiliary register dependent on the states of the control register. If we define a unitary operator $J_{\xi,z} : |y\rangle \mapsto \omega^{\xi z}|y\rangle$ for $y \in \mathbb{Z}_M$, then we can rewrite $R_{\xi,f} \otimes I$ as

$$R_{\xi,f} \otimes I : \sum_{x,y} \alpha_{xy}|x\rangle \otimes |y\rangle \mapsto \sum_{x,y} \alpha_{xy}|x\rangle \otimes J_{\xi,f(x)}|y\rangle. \tag{1}$$

Though $J_{\xi,z}$ alone gives rise to global phase change, which has no effect on quantum states, since the global phase changes on the auxiliary register are controlled by the basis states of the control register as seen in (1), we can obtain the required relative phase changes. We note that $J_{\xi,z}$ has one eigenvalue $\omega^{\xi z}$ and the corresponding eigenspace is the whole Hilbert space. Similarly, we can restrict the functional evaluation operator $U_f : |x, y\rangle \mapsto |x, y + f(x)\rangle$ on the auxiliary register and this localization induces a translation $T_z : |y\rangle \mapsto |y + z\rangle$ where $z = f(x)$ is dependent on the state of the control register. Hence, we can describe U_f in terms of the translation operator on the auxiliary register as follows

$$U_f : \sum_{x,y} \alpha_{xy}|x\rangle \otimes |y\rangle \mapsto \sum_{x,y} \alpha_{xy}|x\rangle \otimes T_{f(x)}|y\rangle. \tag{2}$$

Due to the expressions (1) and (2) the construction of $J_{\xi,z}$ from T_z implies that of $R_{\xi,f}$ from U_f and we can concentrate on the local operations on the auxiliary register.

In particular, when f is the identity map I, we denote $R_{\xi,f}$ by R_ξ. In this case, the functional phase rotation $R_\xi : |y\rangle \mapsto \omega^{\xi y}|y\rangle$ can easily be implemented by $\text{QFT}^\dagger T_\xi^\dagger \text{QFT}$ where QFT represents the quantum Fourier transform. This quantum circuit is depicted in figure 1. Without loss of generality, we may assume that N and M are powers of two, i.e. $N = 2^n$ and $M = 2^m$ for some non-negative integers n and m. We remark that, for general positive integers N and M, the approximate Fourier transform in [18] can be used.

Using R_ξ we can construct a quantum algorithm to implement $J_{\xi,z}$ with two $T_{\pm z}$. To begin with, we prepare a quantum register and denote its state by $|\Phi\rangle = \sum_{y=0}^{M-1} \alpha_y|y\rangle$.
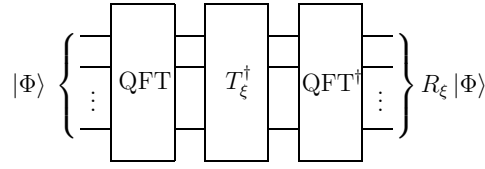
Figure 1. Quantum circuit to implement $R_\xi : |y\rangle \mapsto \omega^{\xi y}|y\rangle$.
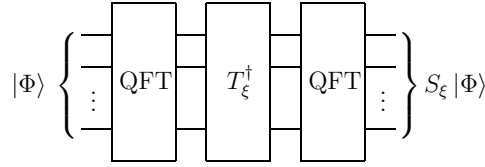

Figure 2. Quantum circuit to implement $S_\xi : |y\rangle \mapsto \omega^{\xi y}|-y\rangle$.

Without initializing the register we proceed with the following algorithm: (i) apply T_z; (ii) apply $R_\xi = \text{QFT}^\dagger T_\xi^\dagger \text{QFT}$; (iii) apply $T_z^\dagger = T_{-z}$; (iv) apply $R_\xi^\dagger = \text{QFT}^\dagger T_\xi \text{QFT}$. Then the state of the register evolves as follows

$$
\begin{aligned}
|\Phi\rangle &\xrightarrow{T_z} \sum_{y=0}^{M-1} \alpha_y |y+z\rangle \\
&\xrightarrow{R_\xi} \sum_{y=0}^{M-1} \omega^{\xi(y+z)} \alpha_y |y+z\rangle \\
&\xrightarrow{T_z^\dagger} \sum_{y=0}^{M-1} \omega^{\xi(y+z)} \alpha_y |y\rangle \\
&\xrightarrow{R_\xi^\dagger} \omega^{\xi z} |\Phi\rangle.
\end{aligned}
\tag{3}
$$

Thus we have $R_\xi^\dagger T_z^\dagger R_\xi T_z |\Phi\rangle = \omega^{\xi z}|\Phi\rangle$ and hence we attain

$$
J_{\xi,z} \equiv \omega^{\xi z} I = R_\xi^\dagger T_z^\dagger R_\xi T_z.
\tag{4}
$$

The algorithm implementing $J_{\xi,z}$ is not unique. All cyclic permutations of the steps in the above algorithm give an identical effect. For example, we may start at step (ii), perform successive steps, and end at step (i). In fact, if we use the notation $[A, B] = ABA^{-1}B^{-1}$ for invertible operators A and B, then from equation (4) we can deduce

$$
J_{\xi,z} = [R_\xi^\dagger, T_z^\dagger] = [T_z, R_\xi^\dagger] = [R_\xi, T_z] = [T_z^\dagger, R_\xi]
\tag{5}
$$

with its inverse $J_{-\xi,z} = [R_\xi, T_z^\dagger] = [T_z, R_{\xi,I}] = [R_\xi^\dagger, T_z] = [T_z^\dagger, R_\xi^\dagger]$. Moreover, instead of R_ξ we can use a unitary operator $S_\xi : |y\rangle \mapsto \omega^{\xi y}|-y\rangle$. Indeed, we can easily check the relation

$$
J_{\xi,z} = S_\xi T_z S_\xi T_z.
\tag{6}
$$

We remark that $S_\xi^\dagger = S_\xi$. The quantum circuit implementing S_ξ by $\text{QFT}T_\xi^\dagger \text{QFT}$ is shown in figure 2.

Even though there are many ways to implement $J_{\xi,z}$, as shown in equations (5) and (6), the number of $T_{\pm z}$ involved in each implementation is equal to two and cannot be reduced.

To see this, we suppose that there is a quantum algorithm implementing $J_{\xi,z}$ and that the only way to encode the information on z is through $T_{\pm z}$. Then the dependence on z requires at least one $T_{\pm z}$ at a certain step. Hence, the overall unitary operation performed by the algorithm can be written by $V_2 T_{\pm z} V_1 = \omega^{\xi z} I$ for some unitary operators V_1 and V_2. Since $V_1 V_2 = \omega^{\xi z} T_{\pm z}^{\dagger}$, $V_1 V_2$ depends on z. Thus, in another step, the information on z should be used once more and so the overall procedure demands at least two translations by $\pm z$. This observation is crucial in showing that at least two queries to the functional evaluation oracle are necessary for any initialization-free algorithms performing the functional phase rotation. This is discussed in the next section.

3. Functional phase rotation

The algorithm for the localized initialization-free functional phase rotation derived in the previous section is extended in this section. We construct an initialization-free quantum algorithm that implements the functional phase rotation using two evaluations of a given function such that the initial state of an auxiliary register remains as it is.

Owing to the relations (1) and (2), the algorithms (5) and (6) for the localized operator $J_{\xi,z}$ on the auxiliary register suggest quantum algorithms that perform the functional phase rotation and preserve the initial state of the auxiliary register. We describe the extension of (4). First of all, we prepare two quantum registers and let $|\Phi\rangle = \sum_{x=0}^{N-1} \alpha_x |x\rangle$ and $|\Psi\rangle = \sum_{y=0}^{M-1} \beta_y |y\rangle$ be the respective states of the control and the auxiliary registers. Here the state of the auxiliary register is temporarily assumed to be pure and no initialization is required during the preparation of the registers. We proceed with the following algorithm: (i) apply U_f; (ii) apply $I \otimes R_{\xi}$; (iii) apply $U_f^{\dagger} = U_{-f}$; (iv) apply $I \otimes R_{\xi}^{\dagger} = I \otimes R_{-\xi}$. This procedure makes the state of the registers evolve as follows

$$
\begin{aligned}
|\Phi\rangle \otimes |\Psi\rangle \xrightarrow{U_f} &\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \alpha_x \beta_y |x\rangle \otimes |y + f(x)\rangle \\
\xrightarrow{I \otimes R_{\xi}} &\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \alpha_x \beta_y \omega^{\xi(y+f(x))} |x\rangle \otimes |y + f(x)\rangle \\
\xrightarrow{U_f^{\dagger}} &\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \alpha_x \beta_y \omega^{\xi(y+f(x))} |x\rangle \otimes |y\rangle \\
\xrightarrow{I \otimes R_{\xi}^{\dagger}} &\left(\sum_{x=0}^{N-1} \omega^{\xi f(x)} \alpha_x |x\rangle \right) \otimes |\Psi\rangle.
\end{aligned}
\tag{7}
$$

Discarding the auxiliary register at the final stage we obtain the functional phase rotation $R_{\xi,f} : |x\rangle \mapsto \omega^{\xi f(x)} |x\rangle$. The quantum circuit for the algorithm (7) is depicted in figure 3. The other extensions of equations (5) and (6) are analogous. For instance, the quantum circuit for $R_{\xi,f}$ developed from equation (6) is shown in figure 4.

In the procedure (7) we have assumed that the auxiliary register is in a pure state. However, this is not an essential requirement. In fact, any mixed state is allowed. To be more precise, we let A be the quantum system of the auxiliary register and we describe its state by the density operator ρ^A. Then a reference system R exists such that the compound system AR is in a pure entangled state $|\Psi^{AR}\rangle$ that gives rise to the given reduced state $\rho^A = \mathrm{Tr}_R(\rho^{AR})$, where $\rho^{AR} = |\Psi^{AR}\rangle\langle\Psi^{AR}|$ is the purification of ρ^A. Using the Schmidt decomposition, we can rewrite $|\Psi^{AR}\rangle$ as $\sum_{y=0}^{M-1} \alpha_y |y\rangle_A \otimes |\Psi_y\rangle_R$. We remark that the states $|\Psi_y\rangle_R$ may not form
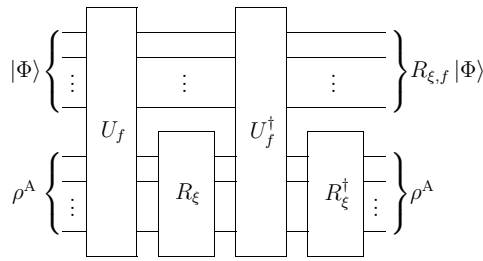
Figure 3. Quantum circuit for the initialization-free functional phase rotation in (7): the state of the auxiliary register is described by a density operator ρ^A.
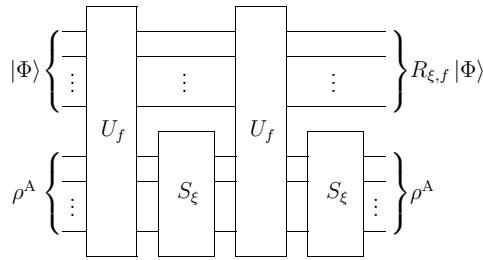

Figure 4. Quantum circuit for the initialization-free functional phase rotation extended from equation (6): the state of the auxiliary register is described by a density operator ρ^A.

the standard basis for the subsystem R but just an orthonormal basis, while the states $|y\rangle_A$ form the standard basis for the subsystem A. Now applying the algorithm (7) to $|\Phi\rangle \otimes |\Psi^{AR}\rangle$ we can see that the final state becomes $(R_{\xi,f}|\Phi\rangle) \otimes |\Psi^{AR}\rangle$. Therefore, the algorithm (7) still works even when the state of the auxiliary register is mixed. This implies that we can compose the auxiliary register of any qubits which are collected out of any other registers, even though they are still occupied by other computational processes and possibly entangled with other qubits. The algorithm (7) recovers the initial state of the joint system AR after extracting the desired relative phase changes. Thus, the qubits in the temporarily composed register can be restored to their positions to continue the suspended computation.

The algorithm (7) requires two functional evaluations, i.e., U_f and $U_f^\dagger$ (two U_f for the extension of (6)). This is because it excludes initialization. From equation (2) it is clear that U_f causes translations in the auxiliary register, and from the argument at the end of section 2 we know that any quantum algorithm for $J_{\xi,z}$ needs at least two translations. Hence, any quantum algorithm that implements the functional phase rotation and leaves the auxiliary register untouched requires at least two evaluations of a given function. On the other hand, if the auxiliary register is initializable only one functional evaluation is sufficient. In fact, since QFT$|-\xi\rangle$ is an eigenvector of T_z and the associated eigenvalue is $\omega^{\xi z}$, if we let $|\Psi\rangle = \text{QFT} \, T_{-\xi}|0\rangle$ then U_f maps $|x\rangle \otimes |\Psi\rangle$ to $\omega^{\xi f(x)}|x\rangle \otimes |\Psi\rangle$. The special case for $\xi = 1$ has been studied in [3, 20]. If the bipartite system AR is separable, that is, the quantum mutual entropy $S(A : R)$ equals zero, and ρ^A is known, then a certain frame change on the subsystem A has an effect on the initialization of the auxiliary register and we can make the auxiliary register regain its early state after performing the functional phase rotation.

For a more general function $f : \mathbb{Z}_N \to [0, 1) \subset \mathbb{R}$, the value $f(x)$ can be approximated by its m-bit binary expansion $(0.a_1 a_2 \ldots a_m)_2 = \sum_{i=1}^m a_i 2^{-i}$ for $a_i \in \mathbb{Z}_2$. We define the m-bit approximation $\tilde{f} : \mathbb{Z}_N \to \mathbb{Z}_M$ of f by $\tilde{f}(x) = \sum_{i=1}^m a_i 2^{m-i}$ where $M = 2^m$ [3, 20]. Then, $R_{\xi,\tilde{f}}$ approximates the operation $|x\rangle \mapsto e^{2\pi i \xi f(x)}|x\rangle$. This approximate functional

phase rotation can be applied to the construction of the conditional γ-phase transform in the generalized quantum search algorithm [9].

4. Conclusions

We have considered a general form of the conventional conditional phase transforms, which we call the functional phase rotation. From the functional evaluation oracle, we have constructed a quantum oracle that performs the functional phase rotation and does not require any kind of initialization of the auxiliary register involved in the functional evaluation. This implies that, when the functional evaluation oracle is used simply to induce the functional phase rotation, there is no need to initialize the auxiliary register. The presented algorithm can also recover the initial state of the auxiliary register, which makes it possible to compose an auxiliary register of any qubits even though they are entangled with others and occupied by other computational processes. While a single call of the functional evaluation oracle is sufficient to realize the functional phase rotation when initialization is involved, the initialization-free algorithm queries the functional evaluation oracle twice.

References

[1] Deutsch D 1985 *Proc. R. Soc.* A **400** 97–117
[2] Deutsch D and Jozsa R 1992 *Proc. R. Soc.* A **439** 553–8
[3] Cleve R, Ekert A, Macciavello C and Mosca M 1998 *Proc. R. Soc.* A **454** 339–54
[4] Collins D, Kim K W and Holton W C 1998 *Phys. Rev.* A **58** R1633–6
[5] Chi D P, Kim J and Lee S 2001 *J. Phys. A: Math. Gen.* **34** 5251–8
[6] Grover L K 1996 *Proc. 28th Annual ACM Symp. on Theory of Computing* (Philadelphia, PA: ACM) pp 212–9
   Grover L K 1997 *Phys. Rev. Lett.* **79** 325–8
[7] Boyer M, Brassard G, Høyer P and Tapp A 1996 *Proc. 4th Workshop on Physics and Computation* (Los Alamitos, CA: IEEE Computer Society Press) pp 36–43
[8] Bennette C H, Bernstein E, Brassard G and Vazirani U 1997 *SIAM J. Comput.* **26** 1510–23
[9] Chi D P and Kim J 1999 *Lecture Notes in Computer Science* vol 1509 (New York: Springer) pp 148–51
   Chi D P and Kim J 1999 *Chaos Solitons and Fractals* **10** 1689–93
[10] Simon D R 1994 *FOCS: Proc. 35th IEEE Symp. on the Foundations of Computer Science* (Piscataway, NJ: IEEE Computer Society Press) pp 116–23
   Simon D R 1997 *SIAM J. Comput.* **26** 1474–83
[11] Brassard G and Høyer P 1997 *Preprint* quant-ph/9704027
[12] Biron D, Biham O, Biham E, Grassl M and Lidar D A 1999 *Lecture Notes in Computer Science* vol 1509 (New York: Springer) pp 140–7
[13] Biham E, Biham O, Biron D, Grassl M and Lidar D A 1999 *Phys. Rev.* A **60** 2742–5
[14] Carlini A and Hosoya A 1999 *Preprint* quant-ph/9909089
[15] Brassard G, Høyer P and Tapp A 1998 *Preprint* quant-ph/9805082
[16] Parker S and Plenio M B 2000 *Phys. Rev. Lett.* **85** 3049–52
[17] Shor P W 1994 *FOCS: Proc. 35th IEEE Symp. on the Foundations of Computer Science* (Piscataway, NJ: IEEE Computer Society Press) pp 124–34
   Shor P W 1997 *SIAM J. Comput.* **26** 1484–509
[18] Kitaev A Y 1995 *Preprint* quant-ph/9511026
[19] Griffiths R B and Niu C-S 1996 *Phys. Rev. Lett.* **76** 3228–31
[20] Cleve R, Ekert A, Henderson L, Macchiavello C and Mosca M 1999 *Preprint* quant-ph/9903061