Initialization-free generalized Deutsch-Jozsa algorithm

2001 J. Phys. A: Math. Gen. 34 5251

(http://iopscience.iop.org/0305-4470/34/25/307)

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **34** (2001) 5251–5258 PII: S0305-4470(01)20206-6

Initialization-free generalized Deutsch–Jozsa algorithm

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

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

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

Received 7 December 2000, in final form 5 April 2001
Published 15 June 2001
Online at stacks.iop.org/JPhysA/34/5251

Abstract

We generalize the Deutsch–Jozsa algorithm by exploiting summations of the roots of unity. The generalized algorithm distinguishes a wider class of functions promised to be either constant or many to one and onto an evenly spaced range. As previously, the generalized quantum algorithm solves this problem using a single functional evaluation. We also consider the problem of distinguishing constant and evenly balanced functions and present a quantum algorithm for this problem that does not require any initialization of an auxiliary register involved in the process of functional evaluation and after solving the problem recovers the initial state of an auxiliary register.

PACS numbers: 0365L, 0365T

1. Introduction

There has been much research to explore the computational power of a quantum computer. The first example of a problem which can be solved exponentially faster on a quantum computer than on a classical Turing machine was given by Deutsch and Jozsa [1]. They presented a simple promise problem that can efficiently be solved without error on a quantum computer but that requires exhaustive search to solve deterministically without error in a classical setting, even though this problem can efficiently be handled with a classical probabilistic computer, provided an arbitrarily small (one-sided) error probability is tolerated.

The Deutsch–Jozsa (DJ) problem [1] is to determine whether a Boolean function $f : \mathbb{Z}_{2^n} \to \mathbb{Z}_2$ computed by an oracle is either non-constant or non-balanced, where f is said to be balanced if $f(x) = 0$ for exactly half of the input values and $f(x) = 1$ for the remaining half of the input values. Subsequent work by Cleve *et al* [2] has generalized this algorithm to distinguish between constant and evenly balanced functions. The original description of the

n-qubit DJ algorithm [1,2] uses an oracle of the form $|x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes |y + f(x)\rangle$ and requires an n-qubit control register for storing function arguments and a one-qubit auxiliary register for functional evaluation. The implementation of this algorithm has successfully been performed using nuclear magnetic resonance (NMR) [3–6]. The refined DJ algorithm [7], which is a description of the original DJ algorithm using an oracle of the form $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$, removed the necessity of an auxiliary register and has also been implemented by the application of NMR [8,9]. A concurrent construction of the interferometer arms for an oracle computing $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ by one-time evolution of a physical system with arbitrary multi-particle interactions was proposed in [10]. Noting that for the one- and two-qubit DJ problem to distinguish functions promised to be either constant or balanced the qubits do not entangle, a Deutsch-like problem to distinguish between even and odd functions was developed in order to acquire two-qubit entanglement [11, 12]. To avoid difficulties in the application of NMR, variations on the function classes of the DJ problem were introduced and it was shown that by adapting it to one such function class the DJ problem is made solvable without exponential loss of signal [13]. It is known that in the black-box model the exponential quantum speed-up obtained for partial functions by Deutsch and Jozsa [1] and by Simon [14] cannot be obtained for any total function [15]. The generalizations of the DJ problem are related to partial functions and hence we can still obtain exponential quantum speed-up.

These generalizations of the original DJ algorithm have all retained the original core summation of powers of -1 to give either exactly zero or one. While each generalization has expanded the class of functions distinguishable using this core summation, the class of distinguishable functions is still restricted. In this paper, we modify the core summation and thus expand the class of distinguishable functions. We do this by noting that for the single qubit of the original proposal, a balanced function is actually many to one and onto, and the sum of powers of -1 is actually a sum of the square roots of unity. This raises the possibility that exploiting summations of the Mth roots of unity might allow a more useful multi-qubit generalization which distinguishes between functions which are many to one and onto evenly spaced ranges and those which are constant. This paper generalizes the DJ algorithm to distinguish between functions promised to be either constant or many to one and onto evenly spaced ranges more efficiently than is possible using any deterministic classical algorithm.

Most quantum algorithms require initialization setting registers to certain states at start-up. In [16] it was shown that one pure qubit and a supply of maximally mixed qubits are sufficient to implement Shor's quantum factoring algorithm [17] efficiently by the combination of the phase estimation technique [18] (see also [2]) with the semiclassical Fourier transform [19]. Recently, Chi *et al* [20] constructed a quantum algorithm that implements an oracle computing $|x\rangle \mapsto \mathrm{e}^{2\pi \mathrm{i} f(x)/M}|x\rangle$ for $f : \mathbb{Z}_N \to \mathbb{Z}_M$ by making use of an oracle of the form $|x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes |y + f(x)\rangle$ without setting the auxiliary register (the second register) to a definite state before the computation. In this paper we consider a generalization of the DJ problem to distinguish constant and evenly balanced functions [2] and show that when an oracle computing $|x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes |y \oplus f(x)\rangle$ is employed, where $\oplus$ denotes the bitwise addition, the auxiliary register can be of any state at the beginning of the computation. Our initialization-free algorithm accepts any state (pure/mixed or separable/entangled) as an initial state of an auxiliary register and its original state is restored at the end of the computation.

Section 2 introduces a generalization of the DJ algorithm to distinguish between functions which are constant and those which are many to one and have evenly spaced images. Section 3 considers a problem to distinguish constant and evenly balanced functions and is devoted to the construction of an algorithm that can solve this problem and requires no initialization of an auxiliary register.

2. Evenly distributed functions

In this section we formulate a more general version of the DJ problem. We establish that summations of the Mth roots of unity allow distinguishing constant functions from functions which are many to one and onto evenly spaced ranges. We also establish the efficiency of the quantum algorithm compared with classical deterministic algorithms.

A convenient definition of the more general class of functions which are many to one and onto evenly spaced ranges employs $\mathbb{Z}_N$, $\mathbb{Z}_M$ and $\mathbb{Z}_K$ for $K \leqslant M, N$. In more detail, the domain of f, $\mathbb{Z}_N$ is mapped to K evenly spaced elements in the range $\mathbb{Z}_M$ with constant separation ($\mu = M/K$), so each element in the range can be indexed by every element of $j \in \mathbb{Z}_K$ via $\mu j + t \in \mathbb{Z}_M$. Here, t is a possible initial shift less than μ in the $\mathbb{Z}_K$ assignments. Further, each element is mapped exactly $\nu = N/K$ times. When these conditions are satisfied, we refer to the function f as evenly distributed. An example for $N = 4$ and $M = 8$ might be a one-to-one function whose image is $\{1, 3, 5, 7\} = \{2j + 1 : j \in \mathbb{Z}_4\}$, so $K = 4$, the separation is $\mu = M/K = 2$ and the initial shift is $t = 1$. Another example for $N = 6$ and $M = 12$ might be a two-to-one function whose image is $\{2, 6, 10\} = \{4j + 2 : j \in \mathbb{Z}_3\}$, so $K = 3$, the separation is $\mu = M/K = 4$, and the initial shift is $t = 2$.

Our generalization of the DJ problem for evenly distributed functions (GDJ-ED) is to determine whether f is non-constant or not evenly distributed. When f is onto, f is an evenly distributed function if and only if f is a ν-to-one function. Thus if $M = K$ then the GDJ-ED problem is equivalent to determining whether f is non-constant or non-ν-to-one. We remark that ν-to-one functions appear in collision and claw problems [21] under the assumption that f is onto. When K is known we need $\nu + 1$ evaluations of f classically in the worst case in order to solve the GDJ-ED problem. Unless K is known, $\frac{1}{2}N + 1$ evaluations are required in the worst case before determining the answer with certainty. Thus the GDJ-ED problem has the same computational complexity as that of the DJ problem.

The required generalization is obtained by modifying the unitary transformation $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$ to $\hat{\mathcal{U}}_f : |x\rangle \mapsto \omega_M^{f(x)}|x\rangle$ written in terms of the Mth root of unity denoted $\omega_M = \mathrm{e}^{2\pi \mathrm{i}/M}$. For simplicity, we assume that N, M and K are powers of two, that is, $N = 2^n$, $M = 2^m$ and $K = 2^k$ for some positive integers n, m and k. We now write $\mathcal{W}_n$ to denote n-qubit Walsh–Hadamard operator. The general algorithm then becomes

$$\begin{aligned}
\mathcal{W}_n \hat{\mathcal{U}}_f \mathcal{W}_n |0^n\rangle &= \frac{1}{\sqrt{N}} \mathcal{W}_n \hat{\mathcal{U}}_f \sum_{x=0}^{N-1} |x\rangle \\
&= \frac{1}{\sqrt{N}} \mathcal{W}_n \sum_{x=0}^{N-1} \omega_M^{f(x)} |x\rangle \\
&= \sum_{y=0}^{N-1} \left\{ \frac{1}{N} \sum_{x=0}^{N-1} (-1)^{x \cdot y} \omega_M^{f(x)} \right\} |y\rangle \qquad (1)
\end{aligned}$$

where $x \cdot y = \sum_{j=0}^{n-1} x_j y_j$ for $x = \sum_{j=0}^{n-1} x_j 2^j$ and $y = \sum_{j=0}^{n-1} y_j 2^j$ ($x_j, y_j \in \mathbb{Z}_2$). Let S_y be the inner summation in the final state of (1). If f is constant, then we obtain

$$S_y = \frac{1}{N} \omega_M^{f(0)} \sum_{x=0}^{N-1} (-1)^{x \cdot y} = \begin{cases} 0 & \text{if} \quad y \neq 0 \\ \omega_M^{f(0)} & \text{if} \quad y = 0. \end{cases}$$

If f is evenly distributed, then for $y = 0$ we have

$$S_0 = \frac{1}{K} \sum_{j=0}^{K-1} \omega_M^{j\mu + t} = \frac{1}{K} \omega_M^t \sum_{j=0}^{K-1} \omega_K^j = 0.$$

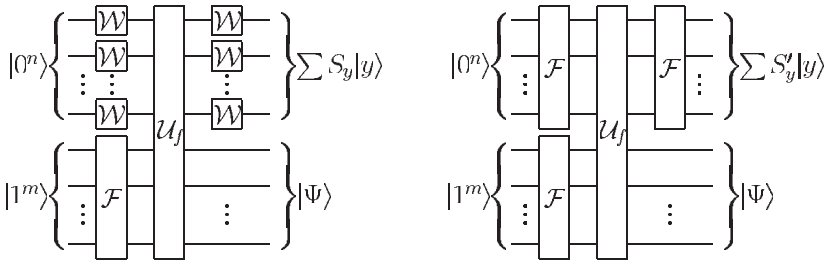

Figure 1. Quantum circuits for the GDJ-ED algorithms in (2) employing (*a*) the n-qubit Walsh–Hadamard operator $\mathcal{W}_n$ and (*b*) the quantum Fourier transform $\mathcal{F}$. Here, ξ is set to 1 and S'_y is the inner summation in the state of (3).

Thus if the outcome of the measurement is $|0^n\rangle$ then f is not evenly distributed and otherwise f is non-constant.

Here, the properties of summations of the roots of unity allow a generalization of the DJ algorithm to distinguish between functions promised to be either constant or evenly distributed. Further, this algorithm requires only a single evaluation of f.

We note that when f is evenly distributed μ can be found by the quantum period-finding algorithm which is the core of the quantum factoring algorithm [17]. Indeed, the application of the quantum Fourier transform to the image of f wipes off the initial shift t and changes its period to $M/\mu = K$, so that with high probability we can determine μ in polynomial time.

When we employ a multi-qubit oracle that performs a functional evaluation by $\mathcal{U}_f$: $|x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes |y + f(x)\rangle$, by slightly modifying the initial state of the auxiliary register in the original DJ algorithm we can solve the GDJ-ED problem. Indeed, we initialize the control register by $|0^n\rangle$ and the auxiliary register by $|\Psi\rangle = \mathcal{F}|-\xi\rangle = (1/\sqrt{M}) \sum_{v=0}^{M-1} \omega_M^{-\xi v} |v\rangle$ for any non-zero $\xi \in \mathbb{Z}_M$ where $\mathcal{F}$ denotes the quantum Fourier transform [22]. We then proceed with the following algorithm. (i) Apply $\mathcal{W}_n \otimes \mathcal{I}$. (ii) Apply $\mathcal{U}_f$. (iii) Apply $\mathcal{W}_n \otimes \mathcal{I}$. The quantum circuit for the GDJ-ED algorithm when $\xi = 1$ is shown in figure 1(*a*). Then the state evolves as follows:

$$\begin{aligned} \left|0^n\right\rangle \otimes |\Psi\rangle &\xrightarrow{\mathcal{W}_n \otimes \mathcal{I}} \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \otimes |\Psi\rangle \\ &\xrightarrow{\mathcal{U}_f} \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \omega_M^{\xi f(x)} |x\rangle \otimes |\Psi\rangle \\ &\xrightarrow{\mathcal{W}_n \otimes \mathcal{I}} \sum_{y=0}^{N-1} \left\{ \frac{1}{N} \sum_{x=0}^{N-1} (-1)^{x \cdot y} \omega_M^{\xi f(x)} \right\} |y\rangle \otimes |\Psi\rangle. \end{aligned} \tag{2}$$

Discarding the auxiliary register we have the same final state as in (1) when $\xi = 1$ and hence algorithm (2) can solve the GDJ-ED problem by a single evaluation of f.

In the overall procedure we can replace $\mathcal{W}_n$ by the quantum Fourier transform. In this case the final state becomes

$$\sum_{y=0}^{N-1} \left(\frac{1}{N} \sum_{x=0}^{N-1} \omega_M^{xy} \omega_M^{\xi f(x)} \right) |y\rangle \otimes |\Psi\rangle \tag{3}$$

and one can easily check that the same result holds with this modified algorithm, whose circuit

is shown in figure 1(*b*). For general positive integers N and M the approximate Fourier transform in [18] can be used.

3. Initialization-free algorithm

In this section we consider a generalized version of the DJ problem formulated by Cleve *et al* [2] to distinguish constant and evenly balanced functions. We construct a quantum algorithm that dispenses with any need for initialization of an auxiliary register and restores the initial state of an auxiliary register after solving this problem.

A function $f : \mathbb{Z}_N \to \mathbb{Z}_M$ is said to be evenly balanced if half the output values of f have parity 0 and half have parity 1. Then the generalized DJ problem for evenly balanced functions (GDJ-EB) is to determine whether f is non-constant or not evenly balanced [2]. Any classical algorithm for the GDJ-EB problem would require $\frac{1}{2}N + 1$ evaluations of f in the worst case before determining the answer with certainty. Thus the GDJ-EB problem has the same computational complexity as that of the GDJ-ED problem.

We remark that the GDJ-EB problem can be solved by directly applying the original DJ algorithm to the composition of the parity function and f. In contrast we construct an initialization-free algorithm that utilizes a quantum oracle evaluating f only. For simplicity, we assume that N and M are powers of two, that is, $N = 2^n$ and $M = 2^m$ for some positive integers n and m. We prepare an n-qubit control register and an m-qubit auxiliary register. Let us assume that the state of the auxiliary register is pure and denote its state by

$$|\Psi\rangle = \sum_{v=0}^{M-1} \alpha_v |v\rangle = \sum_{v=0}^{M-1} \alpha_v \bigotimes_{j=0}^{m-1} \left|v_j\right\rangle$$

where $v = \sum_{j=0}^{m-1} v_j 2^j$ for $v_j \in \mathbb{Z}_2$.

We proceed with the following algorithm. (i) Initialize the control register by $|0^n\rangle$. (ii) Apply $\mathcal{W}_n \otimes \mathcal{I}$. (iii) Apply $\mathcal{U}_f^{\oplus}$. (iv) Apply $\mathcal{I} \otimes \sigma_z^{\otimes m}$. (v) Apply $\mathcal{U}_f^{\oplus}$. (vi) Apply $\mathcal{I} \otimes \sigma_z^{\otimes m}$. (vii) Apply $\mathcal{W}_n \otimes \mathcal{I}$. Here, $\mathcal{U}_f^{\oplus} : |x\rangle \otimes |v\rangle \mapsto |x\rangle \otimes |v \oplus f(x)\rangle$ is the bitwise function-evaluation operator computed by the quantum oracle, $\oplus$ denotes the addition in $\mathbb{Z}_2^m$ and σ_z is the Pauli spin matrix corresponding to the phase-flip operator. Then during steps (i), (ii) and (iii) the state of the control and the auxiliary registers evolves as follows:

$$\begin{aligned}\left|0^n\right\rangle \otimes |\Psi\rangle &\xrightarrow{\mathcal{W}_n \otimes \mathcal{I}} \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \otimes |\Psi\rangle \\ &\xrightarrow{\mathcal{U}_f^{\oplus}} \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \otimes \sum_{v=0}^{M-1} \alpha_v \bigotimes_{j=0}^{m-1} \left|v_j + f(x)_j\right\rangle\end{aligned}$$

where $f(x) = \sum_{j=0}^{m-1} f(x)_j 2^j$ for $f(x)_j \in \mathbb{Z}_2$. After step (iv) the state becomes

$$\begin{aligned}&\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \otimes \sum_{v=0}^{M-1} (-1)^{\sum_{j=0}^{m-1} v_j + f(x)_j} \alpha_v \bigotimes_{j=0}^{m-1} \left|v_j + f(x)_j\right\rangle \\ &\quad = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{p \circ f(x)} |x\rangle \otimes \sum_{v=0}^{M-1} (-1)^{\sum_{j=0}^{m-1} v_j} \alpha_v \bigotimes_{j=0}^{m-1} \left|v_j + f(x)_j\right\rangle\end{aligned}$$

where $p \circ f$ represents the composition of the parity function $p : \mathbb{Z}_2^m \to \mathbb{Z}_2$ and f. After step (v) we have

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{p \circ f(x)} |x\rangle \otimes \sum_{v=0}^{M-1} (-1)^{\sum_{j=0}^{m-1} v_j} \alpha_v \bigotimes_{j=0}^{m-1} \left|v_j\right\rangle$$

and after step (vi) this state changes into

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}(-1)^{p\circ f(x)}|x\rangle\otimes\sum_{v=0}^{M-1}(-1)^{2\sum_{j=0}^{m-1}v_j}\alpha_v\bigotimes_{j=0}^{m-1}\left|v_j\right\rangle=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}(-1)^{p\circ f(x)}|x\rangle\otimes|\Psi\rangle.$$

Finally, after step (vii) we obtain

$$\sum_{y=0}^{N-1}\left\{\frac{1}{N}\sum_{x=0}^{N-1}(-1)^{x\cdot y}(-1)^{p\circ f(x)}\right\}|y\rangle\otimes|\Psi\rangle. \tag{4}$$

Let T_y be the inner summation in the final state of (4). Then when f is constant we have

$$T_y=\frac{(-1)^{p\circ f(0)}}{N}\sum_{x=0}^{N-1}(-1)^{x\cdot y}=\begin{cases}0 & \text{if}\quad y\neq 0\\ (-1)^{p\circ f(0)} & \text{if}\quad y=0\end{cases}$$

and when f is evenly balanced

$$T_0=\frac{1}{N}\sum_{x=0}^{N-1}(-1)^{p\circ f(x)} \tag{5}$$

vanishes. Therefore when f is constant the final state of the control register is $|0^n\rangle$, whereas when f is evenly balanced it is orthogonal to $|0^n\rangle$. Therefore if we measure the control register discarding the auxiliary register, then we can determine whether f is non-constant or not evenly balanced: If the outcome of the measurement is $|0^n\rangle$ then f is not evenly balanced and otherwise f is non-constant.

When N and M are powers of two, this GDJ-EB algorithm can also solve the GDJ-ED problem since an evenly distributed function is evenly balanced. In fact, when f is evenly distributed T_0 in (5) becomes

$$T_0=\frac{1}{K}(-1)^{p(t)}\sum_{j=0}^{K-1}(-1)^{p(j\mu)}=\frac{1}{K}(-1)^{p(t)}\sum_{j=0}^{K-1}(-1)^{p(j)}=0. \tag{6}$$

Here, the second equality in (6) follows from the fact that $j\mu\in\mathbb{Z}_2^m$ is an $m-k$ left shift of a k-bit number $j\in\mathbb{Z}_2^k$ with following zeros where $k=\log_2 K$. For general N and M the class of evenly distributed functions is not contained in the class of evenly balanced functions and the second equality in (6) may not hold. Thus the GDJ-EB algorithm cannot solve the GDJ-ED problem when N and M are not powers of two.

The circuit for the GDJ-EB algorithm is depicted in figure 2. We note that when $|\Psi\rangle=\bigotimes_{j=0}^{m-1}\left(\alpha_j|0\rangle+\beta_j|1\rangle\right)$, the desired phase transform $|x\rangle\mapsto(-1)^{p\circ f(x)}|x\rangle$ is obtained at step (iii) if and only if

$$|\Psi\rangle=\frac{1}{\sqrt{M}}\bigotimes_{j=0}^{m-1}(|0\rangle-|1\rangle) \tag{7}$$

since $\mathcal{U}_f^{\oplus}(|x\rangle\otimes|\Psi\rangle)=(-1)^{p\circ f(x)}|x\rangle\otimes|\Psi\rangle$ is equivalent to $\alpha_j+\beta_j=0$ for all $j=0,1,\ldots,m-1$. In this case steps (iv), (v) and (vi), which correspond to the gates in the dotted box of figure 2, act as the identity operation and hence can be omitted. Thus if we initialize the auxiliary register by the state in (7) then this simplified circuit can solve the GDJ-EB problem by a single functional evaluation with certainty. This simplified algorithm employing the initialization of the auxiliary register was previously constructed by Cleve *et al* [2].

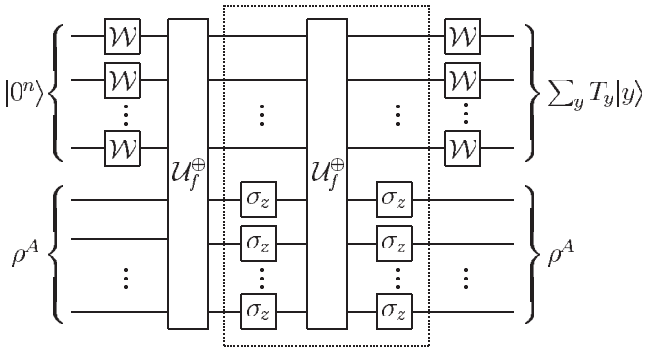

Figure 2. A quantum circuit for the GDJ-EB algorithm: when the auxiliary register is initialized to the state in (7), the gates in the dotted box have no effect on solving the GDJ-EB problem and hence can be omitted.

In the GDJ-EB algorithm we can replace every n-qubit Walsh–Hadamard operator $\mathcal{W}_n$ by the quantum Fourier transform [22] as in the GDJ-ED algorithm. If we do so, then the final state becomes

$$\frac{1}{N}\sum_{x,y=0}^{N-1}\omega_M^{xy}(-1)^{p\circ f(x)}|y\rangle\otimes|\Psi\rangle$$

which still leads to the same conclusion as in (4), and hence we obtain another GDJ-EB algorithm. We remark that for general positive integers N and M the approximate Fourier transform in [18] can be used.

Up to now we have assumed that an auxiliary register is in the pure state. However, this assumption is unnecessary. In fact, any mixed state is allowed. To be more precise, let A be a quantum system to be used as an auxiliary register of which state is described by the density operator ρ^A. We consider a purification of ρ^A. There exists a reference system R such that the compound system AR is in the pure entangled state $|\Psi\rangle_{AR}$ that gives rise to the given reduced state $\rho^A = \mathrm{Tr}_R(\rho^{AR})$. Using the Schmidt decomposition we can rewrite $|\Psi\rangle_{AR}$ as $\sum_{v=0}^{M-1}\alpha_v|v\rangle_A\otimes|\phi_v\rangle_R$. Applying the above algorithm to $|0^n\rangle\otimes|\Psi\rangle_{AR}$ one can see that the final state becomes a separable state

$$\sum_{y=0}^{N-1}\left(\frac{1}{N}\sum_{x=0}^{N-1}(-1)^{x\cdot y}(-1)^{p\circ f(x)}\right)|y\rangle\otimes|\Psi\rangle_{AR}. \tag{8}$$

Thus the GDJ-EB algorithm works for any initial state of the auxiliary register. This implies that we can compose an auxiliary register of any m qubits which are collected from any other registers even though they are still being used in another computational process and are possibly entangled with other qubits. Since the GDJ-EB algorithm recovers the initial state of the joint system AR after extracting the desired relative phase changes in the control register as shown in (8), the qubits in the temporarily composed auxiliary register can be restored to their original positions, and hence can continue the suspended computation.

4. Conclusions

In this paper we dealt with two generalizations of the DJ problem, the GDJ-ED and the GDJ-EB problems, both of which have the same computational complexity as that of the original DJ problem. In section 2 we presented a quantum algorithm that can solve the GDJ-ED

problem with certainty by a single evaluation of a given function. The GDJ-ED algorithm can be constructed from the DJ algorithm by the slight modification of the initial state of the auxiliary register. The initialization-free GDJ-EB algorithm constructed in section 3 requires no knowledge of the initial state of an auxiliary register in advance and, what is more, turns the auxiliary register back to its initial state after solving the GDJ-EB problem. This implies that any register containing useful information to be preserved for another process can temporarily be used as an auxiliary register without degrading its state. We remark that if we employ the initialization-free algorithm in [20], which implements $\hat{\mathcal{U}}_f$ using $\mathcal{U}_f$ and accepts an arbitrary initial state of an auxiliary register, then we can also construct an initialization-free algorithm for the GDJ-ED problem. While our initialization-free algorithm does not demand any *a priori* information on the auxiliary register and its application does not alter the state of the auxiliary register, it requires two evaluations of a given function. This is a trade-off. If initialization is involved then only one functional evaluation is sufficient.

Acknowledgments

This work was supported by the Statistical Research Center for Complex Systems of the Korea Science and Engineering Foundation and the Research Institute of Mathematics. J Kim was supported by the Brain Korea 21 Project and by the National Research Laboratory Project of the Ministry of Science and Technology. D P Chi appreciates the hospitality of Lov K Grover during a visit to DIMACS (Center for Discrete Mathematics and Theoretical Computer Science) and Bell Labs of Lucent Technologies. J Kim would like to thank Michael Gagen for his valuable comment.

References

[1] Deutsch D 1985 *Proc. R. Soc.* A **400** 97–117
Deutsch D and Jozsa R 1992 *Proc. R. Soc.* A **439** 553–8
[2] Cleve R, Ekert A, Macciavello C and Mosca M 1998 *Proc. R. Soc.* A **454** 339–54
[3] Jones J A and Mosca M 1998 *J. Chem. Phys.* **109** 1648–53
[4] Chuang I L, Vandersypen L M K, Zhou X, Leung D W and Lloyd S 1998 *Nature* **393** 143–6
[5] Linden N, Barjat H and Freeman R 1998 *Chem. Phys. Lett.* **296** 61–7
[6] Marx R, Fahmy A F, Myers J M, Bermel W and Glaser S J 2000 *Phys. Rev.* A **62** 012310
[7] Collins D, Kim K W and Holton W C 1998 *Phys. Rev.* A **58** R1633–6
[8] Arvind, Dorai K and Kumar A 1999 *Preprint* quant-ph/9909067
[9] Collins D, Kim K W, Holton W C, Sierzputowska-Gracz H and Stejskal E O 2000 *Phys. Rev.* A **62** 022304
[10] Yamaguchi F, Master C P and Yamamoto Y 2000 *Preprint* quant-ph/0005128
[11] Arvind and Mukunda N 2000 *Preprint* quant-ph/0006069
[12] Dorai K, Arvind and Kumar A 2001 *Phys. Rev.* A **63** 034101
[13] Myers M, Fahmy A F, Glasser S J and Marx R 2001 *Phys. Rev.* A **63** 032302
[14] 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
[15] Beals R, Buhrmann H, Cleve R, Mosca M and de Wolf R 1998 *Preprint* quant-ph/9802049
[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] Chi D P, Kim J and Lee S 2000 *Preprint* quant-ph/0006039
[21] Brassard G, Høyer P and Tapp A 1997 *Preprint* quant-ph/9705002
[22] Coppersmith D 1994 *IBM Research Report* RC19642 (New York: Yorktown Heights)