Exact quantum algorithm to distinguish Boolean functions of different weights

Samuel L Braunstein[1], **Byung-Soo Choi**[1,2], **Subhroshekhar Ghosh**[3] and **Subhamoy Maitra**[4]

[1] Computer Science, University of York, York YO10 5DD, UK

[2] School of Information and Communication, Sungkyunkwan University, Republic of Korea

[3] Indian Statistical Institute, Kolkata 700 108, India

[4] Applied Statistics Unit, Indian Statistical Institute, Kolkata 700 108, India

Abstract

In this work, we exploit the Grover operator for the weight analysis of a Boolean function, specifically to solve the weight-decision problem. The weight w is the fraction of all possible inputs for which the output is 1. The goal of the weight-decision problem is to find the exact weight w from the given two weights w_1 and w_2 satisfying a general weight condition as $w_1 + w_2 = 1$ and $0 < w_1 < w_2 < 1$. First, we propose a limited weight-decision algorithm where the function has another constraint: a weight is in $\left\{w_1 = \sin^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right), w_2 = \cos^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)\right\}$ for integer k. Second, by changing the phases in the last two Grover iterations, we propose a general weight-decision algorithm which is free from the above constraint. Finally, we show that when our algorithm requires $O(k)$ queries to find w with a unit success probability, any classical algorithm requires at least $\Omega(k^2)$ queries for a unit success probability. In addition, we show that our algorithm requires fewer queries to solve this problem compared with the quantum counting algorithm.

PACS number: 03.67.Lx

1. Introduction

In 1985, Deutsch designed a quantum algorithm to determine exactly whether a given Boolean function on one bit is constant or balanced using only one evaluation [1]. Deutsch and Jozsa generalized this algorithm to Boolean functions on n bits, showing an exponential speedup for a quantum machine compared to classical ones [2]. The most important contribution in this field was achieved when Shor discovered polynomial-time quantum algorithms for factoring and computing discrete logarithms, which are exponentially faster than classical algorithms [3]. Because the quantum factoring algorithm shows an exponential speedup, many researchers started to look for other applications. Perhaps the next most significant breakthrough occurred when Grover discovered a quantum database search algorithm which is quadratically faster than classical algorithms [4, 5]. Because the database search is one of the most widely used algorithms in computer applications, the impact was huge, and more researchers have sought other applications.

In this paper we use the Grover operator to efficiently and exactly analyse the weight of a Boolean function. A weight is defined to be the ratio of the number of solutions over all possible inputs of a Boolean function. The weight analysis of Boolean functions is widely used for cryptanalysis [6], coding theory [7], fault-tolerant circuit design [8] and built-in self-testing of circuits [9]. For some restricted classes of Boolean functions, a polynomial-time weight-finding algorithm has been developed, while a *general* weight-finding algorithm requires an exponential query complexity $O(2^n)$ where n is the number of variables [10]. For example, a Deutsch–Jozsa-based algorithm was analysed for cryptanalysis purposes to show an exponential speedup compared to classical approaches [11]. Hence quantum algorithms for the weight analysis of a Boolean function will likely find real-world applications.

Here we shall focus on a special type of weight analysis problem to find an exact weight w from the given set of two weights $\{w_1, w_2 | w_1 + w_2 = 1, 0 < w_1 < w_2 < 1\}$, which is called a weight-decision problem. In other words, we assume that it is already known that a Boolean function f has a weight of either w_1 or w_2 where $w_1 + w_2 = 1$. Our algorithm must determine which of these two possible values is the exact weight of the given Boolean function. Note that in this work we only consider the case $w_1 + w_2 = 1$ because it is a comparatively easy problem, and we hope it will provide hints for the more general case. Meanwhile our approach is a suitable one when w_1 and w_2 are close to 0.5. When w_1 is close to 0 and w_2 is close to 1, even a few classical queries are sufficient to decide the exact weight with high probability. However, when w_1 and w_2 are close to 0.5, the success probability is very low in a classical approach. This case motivates our work. Meanwhile, it should also be emphasized that a sure-success algorithm is mandatory for some applications such as critical decision systems. Hence, we consider only the sure-success algorithm in our work.

Note that the difference between the Grover search and our approach should be emphasized since our approach is heavily based on the Grover search. Given an oracle $f(x)$ with one solution $f(x_{\text{sol}}) = 1$, the Grover search determines x_{sol}, where x is an n-bit input. In other words, we can view the Grover search as an attempt to determine which of 2^n possible oracles has been handed to us. However, regardless of which oracle is present, the initial state of the Grover search can always be written as $|\psi\rangle = \sqrt{\frac{1}{N}}|s\rangle + \sqrt{\frac{N-1}{N}}|ns\rangle$, where $|s\rangle$ and $|ns\rangle$ are normalized solution and non-solution basis states, respectively. The Grover search is simply a rotation of this initial state which is identical for all possible oracles, onto the final state $|s\rangle$, which differs for different oracles. On the other hand, the initial state of our algorithm (in the 2-state representation) depends on the actual weight of the oracle. The initial state may be written as $|\psi\rangle = \sqrt{w_1}|s\rangle + \sqrt{w_2}|ns\rangle$ or $|\psi\rangle = \sqrt{w_2}|s\rangle + \sqrt{w_1}|ns\rangle$ with the weight condition $w_1 + w_2 = 1$. Then using k operations, we attempt to rotate this initial state onto opposite poles of the Bloch sphere in a manner dependent on the actual weight of the oracle.

Specifically, we make the following contributions.

- *Limited weight-decision algorithm.* We show that using k iterations of the Grover operator would allow us to decide exactly w from $\left\{w_1 = \sin^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right), w_2 = \cos^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)\right\}$ for integer k. Note that since it has another constraint on the given weights, we call it a limited weight-decision algorithm.

- *General weight-decision algorithm*. We propose a general weight-decision algorithm, without the above limiting constraint, by exploiting a sure-success database search method. The general weight-decision algorithm uses $k-2$ Grover iterations followed by two further phase-modified Grover operators.
- *Performance analysis*. We show that if our quantum algorithm requires $O(k)$ Grover iterations, where k is an integer value, then any classical algorithm requires $\Omega(k^2)$ queries. Note that $O(m)$ and $\Omega(m)$ mean that the algorithm requires asymptotically at most and at least m queries, respectively. Hence our quantum algorithm achieves at least a quadratic speedup.

  Meanwhile one could solve this weight-decision problem using the quantum counting algorithm, allowing a direct estimate of the weight of the function involved. However, our approach does not rely on the quantum Fourier transform and requires fewer queries.

This paper is organized as follows. Section 2 defines two weight-decision problems and analyses the Grover operator in the Hilbert space and on the Bloch sphere. Section 3 shows what weights can be decided exactly after k Grover iterations and the consequent limitation on the weights. Section 4 shows how to decide exactly the weight w by modifying the phases of the last two Grover iterations when the set of two weights has only the general weight condition. Performance comparisons between our algorithm and the classical, and the quantum counting approach are discussed in section 5. Section 6 concludes the paper with a brief summary and mentions several open problems.

2. Preliminaries

2.1. Definitions

Definition 1 (Weight w of a Boolean function f)**.** *The weight w of a Boolean function f is defined as the ratio of the number of inputs for which outputs are 1 over the number of all possible inputs of f.*

Definition 2 (General weight condition)**.** *The weight of a Boolean function f is one of two weights in $\{w_1, w_2 | w_1 + w_2 = 1, 0 < w_1 < w_2 < 1\}$.*

Definition 3 (General weight-decision problem)**.** *Given a Boolean function f with the general weight condition, decide exactly w of f.*

Definition 4 (Limited weight-decision problem)**.** *Given a Boolean function f with the general weight condition and also the condition that the weights can be written as $\left\{w_1 = \sin^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right), w_2 = \cos^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)\right\}$ for some integer k, decide exactly w of f.*

2.2. Analysis of the Grover operator

2.2.1. The Grover operator. Consider a Boolean function f with a weight $w = \sin^2\frac{\beta_w}{2}$. The uniform superposition of all states is used as the initial state for the algorithm and may be expressed as

$$|\psi_{w,0}\rangle = \sin\frac{\beta_w}{2}|s\rangle + \cos\frac{\beta_w}{2}|ns\rangle, \tag{1}$$

where $|s\rangle$ and $|ns\rangle$ denote the uniform superpositions of solution (i.e., where $f(x) = 1$) and non-solution (i.e., where $f(x) = 0$) basis states, respectively. Now the generalized Grover operator consists of two inversion operators as

$$G = -I_{|\psi_{w,0}\rangle}(\theta) I_{|s\rangle}(\phi), \tag{2}$$

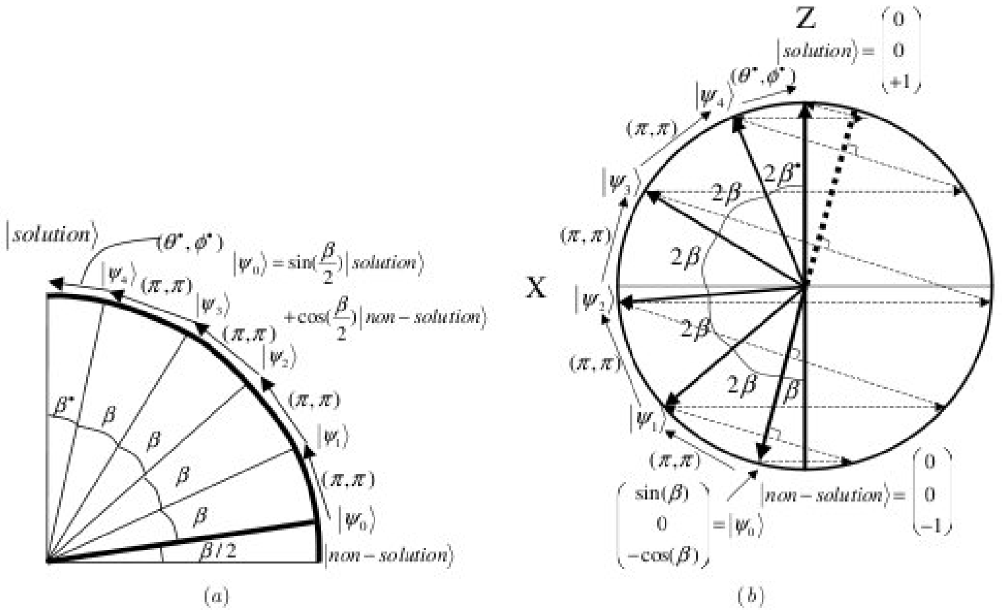

Figure 1. Grover's database search ($\theta = \phi = \pi$). Evolution of the quantum states through (*a*) the Hilbert space and (*b*) the Bloch sphere.

where the inversion operator is defined

$$I_{|\psi\rangle}(\theta) \equiv I - (1 - \mathrm{e}^{\mathrm{i}\theta})|\psi\rangle\langle\psi|. \tag{3}$$

In the standard Grover search algorithm, $\theta = \phi = \pi$. Hence the standard form is

$$\begin{aligned} G &= -I_{|\psi_{w,0}\rangle}(\pi) I_{|s\rangle}(\pi) \\ &= (2|\psi_{w,0}\rangle\langle\psi_{w,0}| - I)(I - 2|s\rangle\langle s|), \end{aligned} \tag{4}$$

where $-I_{|\psi_{w,0}\rangle}(\pi)$ inverts all states about the average, and $I_{|s\rangle}(\pi)$ flips the sign of all solution states. In short, we might summarize the Grover operator as consisting of two inversions—one about the initial uniform-superposition state and the other about the solution states.

After applying the standard Grover operator k times, the initial state becomes

$$|\psi_{w,k}\rangle = \sin(2k+1)\frac{\beta_w}{2}|s\rangle + \cos(2k+1)\frac{\beta_w}{2}|ns\rangle. \tag{5}$$

If we now measure this state in the computational basis, we can find one of the solutions with a success probability of $\sin^2(2k+1)\frac{\beta_w}{2}$. Figure 1(*a*) shows the evolution of the states in the Hilbert space under the Grover operation.

2.2.2. The Bloch sphere. Because the action of the Grover search algorithm takes place within a 2D Hilbert space, we may represent all actions on the surface of the 3D Bloch sphere. The two inversion operators in the 2D Hilbert space can be regarded as a pair of rotation operators on the 3D Block sphere [13] as

$$G = -\mathrm{e}^{\mathrm{i}(\frac{\theta}{2}+\frac{\phi}{2})} R_{|\psi_{w,0}\rangle}(-\theta) R_{|s\rangle}(-\phi), \tag{6}$$

and the initial state becomes a vector

$$\begin{pmatrix} \sin\beta_w \\ 0 \\ -\cos\beta_w \end{pmatrix}. \tag{7}$$

Figure 1(*b*) shows the evolution of the quantum states on the Bloch sphere. Note that all figures hereafter are viewed from the +*Y*-axis of the Bloch sphere for easier comprehension. The vector of the state $|\psi_{w,k}\rangle$ is

$$\begin{pmatrix} \sin[(2k+1)\beta_w] \\ 0 \\ -\cos[(2k+1)\beta_w] \end{pmatrix}. \tag{8}$$

Algorithm 1 Limited weight-decision algorithm

```
Apply $k$ Grover operators to $|\psi_{w,0}\rangle$.
Measure $|\psi_{w,k}\rangle$ in the computational basis.
Let the measured result be $\hat{x}$.
If $k$ is even and $f(\hat{x}) = 1$, $w = w_2$ else $w = w_1$.
If $k$ is odd and $f(\hat{x}) = 1$, $w = w_1$ else $w = w_2$.
```

Because the Bloch sphere representation provides a simple geometric picture, we shall use it to illustrate all the algorithms we devise here.

3. Limited weight-decision algorithm

In this section, we study which weights can be decided exactly after k Grover iterations. Consider a situation where the weight is $w_1 = \sin^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)$, so β_{w_1} should be $\frac{k}{2k+1}\pi$. Note that if the weight is $w_2 = \cos^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)$, we may reformulate it as

$$\cos^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right) = \cos^2\left(\frac{2k+1-k-1}{2k+1}\frac{\pi}{2}\right) = \sin^2\left(\frac{k+1}{2k+1}\frac{\pi}{2}\right), \tag{9}$$

to conclude that β_{w_2} should be $\frac{k+1}{2k+1}\pi$.

After applying k Grover operators, the final states for the two cases $w = w_1$ and $w = w_2$ are

$$|\psi_{w_1,k}\rangle = \begin{pmatrix} \sin(k\pi) \\ 0 \\ -\cos(k\pi) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\cos(k\pi) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ (-1)^{k+1} \end{pmatrix} \tag{10}$$

and

$$|\psi_{w_2,k}\rangle = \begin{pmatrix} \sin[(k+1)\pi] \\ 0 \\ -\cos[(k+1)\pi] \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\cos[(k+1)\pi] \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ (-1)^{k} \end{pmatrix}, \tag{11}$$

respectively.

Note that if k is even, $|\psi_{w_1,k}\rangle = (0, 0, -1)^T$ which denotes the normalized non-solution state; similarly, $|\psi_{w_2,k}\rangle = (0, 0, +1)^T$ which denotes the normalized solution state. As a result, if k is even and the measured value $\hat{x}$ is one of the solutions, then we can conclude that $w = w_2$ otherwise $w = w_1$. On the other hand if k is odd, the two final states are exchanged. Hence if the measured value $\hat{x}$ is one of the solutions, we can conclude that $w = w_1$ otherwise $w = w_2$. Algorithm 1 summarizes this procedure. Note that the case for $k = 1$ was described in [12].

Note that this algorithm can find the exact weight only when $w_1 + w_2 = 1$ and $w_1 = \sin^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)$ and $w_2 = \cos^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)$. Unfortunately the limited weight-decision algorithm has two problems as follows.

- *Irrational weights.* If $w = \sin^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)$, where k is an integer, w cannot be rational, and therefore cannot correspond to a real oracle except for the trivial cases $k = 0, 1$.
- *Not sure success.* As a consequence of $k = 0, 1$ providing the only exact solutions, if we apply the algorithm anyway with the integer number of steps k chosen so that $\sin^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)$ and $\cos^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)$ most closely approximate w_1 and w_2, we will end up with an algorithm which does not perform with sure success (i.e., with unit probability). Though the error probability may be small in many cases, this may still be unsatisfactory. For example, if the difference between the given two weights is very small, the error in the limited weight-decision algorithm will not be negligible. Even worse, if the limited weight-decision algorithm is used repeatedly as a subroutine in a larger quantum program, the error tolerance of the whole procedure will continually diminish.

Notwithstanding these problems, we considered the limited weight-decision algorithm here to motivate the general weight-decision algorithm, for which we will be able to relax the restriction that the weights have the form $\left\{\sin^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right), \cos^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)\right\}$.

4. General weight-decision algorithm

4.1. Motivation

In order to resolve the problems discussed above with the limited weight-decision algorithm, we consider a general weight-decision algorithm with the looser general weight condition. This generalization is potentially worthwhile both theoretically and practically. To achieve this goal we exploit the sure-success approach for the Grover search. Note that since there are many different sure-success approaches, it is likely that there are many different ways to incorporate sure success into our algorithm. However, in this work we focused on what we feel is the simplest approach. Thus, we first review the sure-success database search.

4.2. Sure-success database search

To achieve a sure-success database search many approaches have been developed [13–26]. Typically, a sure-success database search is based on a method changing the two phases, θ and ϕ of $I_{|\psi_{w,0}\rangle}(\theta)$ and $I_{|s\rangle}(\phi)$. In this work, we exploit the approach developed by Brassard *et al* [24]. This procedure is based on the following approach. First, one calculates the required minimum number k of the Grover iterations. Then, from the 1st to the $(k-1)$th operation, the standard Grover operator is applied. However, for the last (kth) operation, a generalized Grover operator is applied by choosing two phases, $\theta, \phi \neq \pi$, for $I_{|\psi_{w,0}\rangle}(\theta)$ and $I_{|s\rangle}(\phi)$, respectively.

Algorithm 2 General weight-decision algorithm

```
1.   $|\psi_{w,0}\rangle = |0\rangle^{\otimes n}|1\rangle, i = 0,$
     If $w_1 \leqslant \sin^2 \frac{\pi}{5}$, $k$ is 2,
     otherwise $k$ satisfies $\sin^2\left(\frac{k-1}{2k-1}\frac{\pi}{2}\right) < w_1 \leqslant \sin^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)$.
     $\frac{k-1}{2k-1}\pi < \beta_{w_1} \leqslant \frac{k}{2k+1}\pi$.
     $\beta_{w_1} + \beta_{w_2} = \pi$.
2.   While$(i < (k-2))$ do
     $\{|\psi_{w,i+1}\rangle = -I_{|\psi_{w,0}\rangle}(\pi)I_{|s\rangle}(\pi)|\psi_{w,i}\rangle, i = i+1\}$
3.   $|\psi_{w,k-1}\rangle = -I_{|\psi_{w,0}\rangle}(-\theta_1)I_{|s\rangle}(\pi)|\psi_{w,k-2}\rangle$
4.   $|\psi_{w,k}\rangle = -I_{|\psi_{w,0}\rangle}(-\theta_2)I_{|s\rangle}(\pi)|\psi_{w,k-1}\rangle$
5.   Measure $|\psi_{w,k}\rangle$ in the computational basis.
     Let the result be $\hat{x}$.
6-1. If $k$ is odd and if $f(\hat{x}) = 1$ then $w = w_1$ else $w = w_2$.
6-2. If $k$ is even and if $f(\hat{x}) = 1$ then $w = w_2$ else $w = w_1$.
```

4.3. *Modified approach for a general weight-decision algorithm*

At the first sight it looks like that Brassard *et al* approach can be directly applied to the general weight-decision problem. However, the current application is somewhat different from the usual database search scenario as follows. The Brassard *et al* method changes only the last operation because its goal is to rotate the state $|\psi_{w,k-1}\rangle$ onto the solution state. However, in the weight-decision scenario, we need to satisfy a more stringent condition: the initial state corresponding to different weights, i.e., $|\psi_{w_1,0}\rangle$ and $|\psi_{w_2,0}\rangle$, should be correctly rotated to the solution (non-solution) state and the non-solution (solution) state, exclusively. From this condition, we can obtain a relation between $\left\{|\psi_{w_1,0}\rangle, |\psi_{w_2,0}\rangle\right\}$ and $\{|s\rangle, |ns\rangle\}$. To satisfy this condition, we propose an approach which changes the last two Grover operations as follows. From the 1st to the $(k-2)$th Grover operation, we use the standard Grover operator with (π, π) phase angles for the $I_{|\psi_{w,0}\rangle}(\theta)$ and $I_{|s\rangle}(\phi)$ operations. However, for the $(k-1)$th and the kth operations, we use slightly modified Grover operators with the phases $(-\theta_1, \pi)$ and $(-\theta_2, \pi)$, respectively. Algorithm 2 summarizes the proposed procedure.

4.3.1. Modification of the phases for the last two Grover operations. We now concentrate on the evolution of the quantum states during the last two steps of our algorithm. Figure 2 shows how we can rotate two initial states, $|\psi_{w_1,0}\rangle$ and $|\psi_{w_2,0}\rangle$, to the states $|ns\rangle$ and $|s\rangle$, respectively. In the figure, two initial states for two different weights are shown differently, but they are actually the same initial state in the algorithm. Likewise, two sets of solutions and non-solutions for different weights are shown as the same state in the figure, but the actual set of solution and non-solutions are different. Note that figure 2 shows a case when only two operations are sufficient to decide w exactly. In the figure, the circle and the diamond denote the two states $|\psi_{w_1,i}\rangle$ and $|\psi_{w_2,i}\rangle$, respectively. If the circle/diamond is filled, the state points towards the positive Y-axis, otherwise the negative Y-axis. Our purpose is to find phase conditions, which can rotate two initial states to the opposite (and hence orthogonal) poles of the Bloch sphere exclusively, yet using the same phase conditions. For example, if $w = w_1(w_2)$ and the method rotates $|\psi_{w_1,0}\rangle(|\psi_{w_2,0}\rangle)$ to $|ns\rangle$, the same method should rotate $|\psi_{w_2,0}\rangle(|\psi_{w_1,0}\rangle)$ to $|s\rangle$. In the initial step, the two initial states $|\psi_{w_1,0}\rangle$ and $|\psi_{w_2,0}\rangle$ are $\left(\sin\beta_{w_1}, 0, -\cos\beta_{w_1}\right)^T$ and $\left(\sin\left(\pi-\beta_{w_1}\right), 0, -\cos\left(\pi-\beta_{w_1}\right)\right)^T = \left(\sin\beta_{w_1}, 0, \cos\beta_{w_1}\right)^T$, respectively. In the first step, the two initial states are rotated to $|A_1\rangle$ and $|B_1\rangle$ states, respectively, by the common rotation operator $R_{|s\rangle}(\pi)$. In the second step, these states are rotated to the states $|A_2\rangle$ and

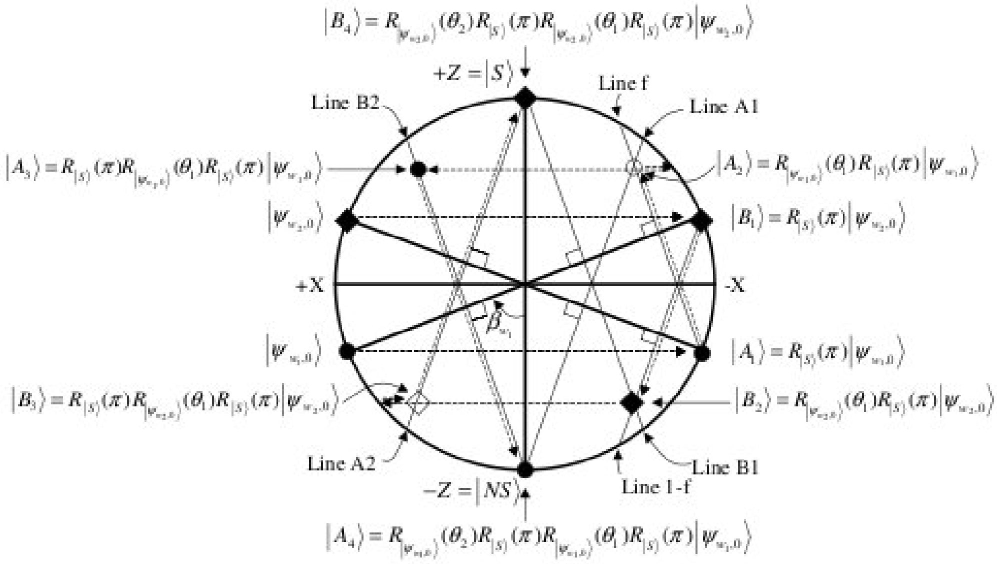
$|B_4\rangle = R_{|\psi_{w_2,k}\rangle}(\theta_2)R_{|S\rangle}(\pi)R_{|\psi_{w_2,k}\rangle}(\theta_1)R_{|S\rangle}(\pi)|\psi_{w_2,0}\rangle$
$+Z = |S\rangle$
Line f
Line B2
Line A1
$|A_3\rangle = R_{|S\rangle}(\pi)R_{|\psi_{w_1,k}\rangle}(\theta_1)R_{|S\rangle}(\pi)|\psi_{w_1,0}\rangle$
$|A_2\rangle = R_{|\psi_{w_1,k}\rangle}(\theta_1)R_{|S\rangle}(\pi)|\psi_{w_1,0}\rangle$
$|\psi_{w_2,0}\rangle$
$|B_1\rangle = R_{|S\rangle}(\pi)|\psi_{w_2,0}\rangle$
+X
-X
β_{w_1}
$|\psi_{w_1,0}\rangle$
$|A_1\rangle = R_{|S\rangle}(\pi)|\psi_{w_1,0}\rangle$
$|B_3\rangle = R_{|S\rangle}(\pi)R_{|\psi_{w_2,k}\rangle}(\theta_1)R_{|S\rangle}(\pi)|\psi_{w_2,0}\rangle$
$|B_2\rangle = R_{|\psi_{w_2,k}\rangle}(\theta_1)R_{|S\rangle}(\pi)|\psi_{w_2,0}\rangle$
Line A2
Line B1
Line 1-f
$-Z = |NS\rangle$
$|A_4\rangle = R_{|\psi_{w_1,k}\rangle}(\theta_2)R_{|S\rangle}(\pi)R_{|\psi_{w_1,k}\rangle}(\theta_1)R_{|S\rangle}(\pi)|\psi_{w_1,0}\rangle$

Figure 2. Last two Grover operations for the general algorithm: The circle and diamond denote the states $|\psi_{w1,k}\rangle$ and $|\psi_{w2,k}\rangle$ respectively.

$|B_2\rangle$, respectively, by the common rotation operator $R_{|\psi_{w,0}\rangle}(\theta_1)$. Here, $|A_1\rangle$ should be rotated to $|A_2\rangle$, which corresponds to the crossing point between the line $A1$ and the line f. The line f is a path, where the point $|A_1\rangle$ can be rotated by the rotation operator $R_{|\psi_{w,0}\rangle}(\theta_1)$. Using the same rule, the same rotation operator rotates $|B_1\rangle$ to $|B_2\rangle$, where $|B_2\rangle$ occurs at the crossing point between the line $B1$ and the line $1 - f$. For the third step, the operator $R_{|S\rangle}(\pi)$ is used. In the final step, the two states are rotated to the opposite poles using the same rotation angle θ_2. Finally, if we measure the final state and the measured value $\hat{x}$ is one of the solutions (non-solutions), we can decide exactly that $w = w_2$ (w_1). In our approach, the key point is to find two crossing points, denoted by the states $|A_2\rangle$ and $|B_2\rangle$, with the required number of operations.

4.3.2. Correctness. In the proposed method, we have to change two phases only for the last two operations, not for any other operations because until the $(k-2)$th operation, there is no crossing point such as $|A_2\rangle$ and $|B_2\rangle$ in figure 2. Therefore, in order to prove the correctness of our approach, we need to show that until the $(k-2)$th operation, there is no such crossing point, but that at the $(k-1)$th operation, there are two such crossing points. To provide a straightforward explanation, we consider the case only when k is odd, $w = w_1$ and $\frac{k-1}{2k-1}\pi < \beta_{w_1} \leqslant \frac{k}{2k+1}\pi$. The other case may be proven in a similar manner.

(i) *No crossing point until the* $(k-2)$*th operation.* Figure 3(a) shows the state $|\psi_{w_1,k-2}\rangle$ after $k-2$ operations of $R_{|\psi_{w_1,0}\rangle}(\pi)R_{|S\rangle}(\pi)$ have been applied. Note that when k is odd, the state $|\psi_{w_1,k-2}\rangle$ should be located in the upper-right part, i.e., in the $(-X, +Z)$ area. Meanwhile, the line $A1$, which is perpendicular to the axis of $|\psi_{w_2,0}\rangle$ and meets the south pole, is given by $Z = -X \tan\beta_{w_1} - 1$. The value of x at the crossing point between the line $A1$ and the circle is $x_{\text{line}} = -\sin 2\beta_{w_1}$, and the value of x for the state $|\psi_{k-2}\rangle$, namely x_{k-2}, is $\sin(2(k-2)+1)\beta_{w_1}$. Therefore, to show that there is no crossing point until the $(k-2)$th operation, we need to prove that x_{line} is always larger than x_{k-2}. The following fact, the proof of which is given in appendix A.1, shows the correctness of this argument.

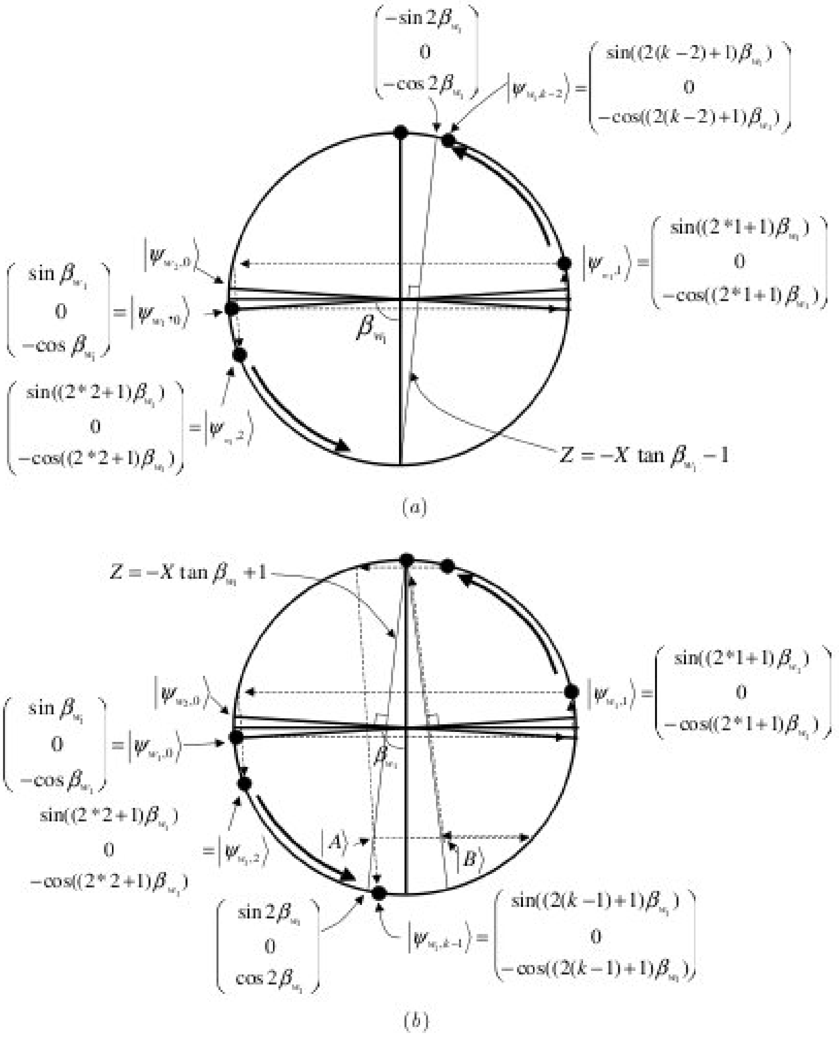
$\begin{pmatrix}-\sin 2\beta_{w_1}\\0\\-\cos 2\beta_{w_1}\end{pmatrix}$
$|\psi_{w_1,k-2}\rangle = \begin{pmatrix}\sin((2(k-2)+1)\beta_{w_1})\\0\\-\cos((2(k-2)+1)\beta_{w_1})\end{pmatrix}$
$|\psi_{w_2,0}\rangle$
$|\psi_{w_1,1}\rangle = \begin{pmatrix}\sin((2*1+1)\beta_{w_1})\\0\\-\cos((2*1+1)\beta_{w_1})\end{pmatrix}$
$\begin{pmatrix}\sin\beta_{w_1}\\0\\-\cos\beta_{w_1}\end{pmatrix} = |\psi_{w_1,0}\rangle$
β_{w_1}
$\begin{pmatrix}\sin((2*2+1)\beta_{w_1})\\0\\-\cos((2*2+1)\beta_{w_1})\end{pmatrix} = |\psi_{w_1,2}\rangle$
$Z = -X\tan\beta_{w_1} - 1$
(a)
$Z = -X\tan\beta_{w_1} + 1$
$|\psi_{w_2,0}\rangle$
$|\psi_{w_1,1}\rangle = \begin{pmatrix}\sin((2*1+1)\beta_{w_1})\\0\\-\cos((2*1+1)\beta_{w_1})\end{pmatrix}$
$\begin{pmatrix}\sin\beta_{w_1}\\0\\-\cos\beta_{w_1}\end{pmatrix} = |\psi_{w_1,0}\rangle$
β_{w_1}
$\begin{matrix}\sin((2*2+1)\beta_{w_1})\\0\\-\cos((2*2+1)\beta_{w_1})\end{matrix} = |\psi_{w_1,2}\rangle$
$|A\rangle$
$|B\rangle$
$\begin{pmatrix}\sin 2\beta_{w_1}\\0\\\cos 2\beta_{w_1}\end{pmatrix}$
$|\psi_{w_1,k-1}\rangle = \begin{pmatrix}\sin((2(k-1)+1)\beta_{w_1})\\0\\-\cos((2(k-1)+1)\beta_{w_1})\end{pmatrix}$
(b)

Figure 3. Correctness of the modification of the last two Grover operations. (*a*) No crossing until the $(k-2)$th operation. (*b*) First crossing at the $(k-1)$th operation.

Fact 1. $-\sin 2\beta_{w_1} > \sin(2(k-2)+1)\beta_{w_1}$, where $\frac{k-1}{2k-1}\pi < \beta_{w_1} \leqslant \frac{k}{2k+1}\pi$.

(ii) *First Crossing Point at the* $(k-1)$th *Operation*. Figure 3(*b*) shows why the first crossing point between the two lines $A1$ and f occurs in the $(k-1)$th operation. To prove this, we need to show that the value of x, x_{line}, of the crossing point between the line $A1$ and the circle is always greater than or equal to the value of x of the state $|\psi_{w_1,k-1}\rangle$, namely x_{k-1}. Note that the value of x_{line} is $\sin 2\beta_{w_1}$, and the value of x_{k-1} is $\sin(2(k-1)+1)\beta_{w_1}$. The following fact, proven in appendix A.2, shows the correctness of this argument.

Fact 2. $\sin 2\beta_{w_1} \geqslant \sin(2(k-1)+1)\beta_{w_1}$, where $\frac{k-1}{2k-1}\pi < \beta_{w_1} \leqslant \frac{k}{2k+1}\pi$.

4.3.3. Phase conditions. Figures 4(*a*) and (*b*) show the evolution of quantum states for the last two operations when k is even and odd, respectively. Note that we consider Boolean

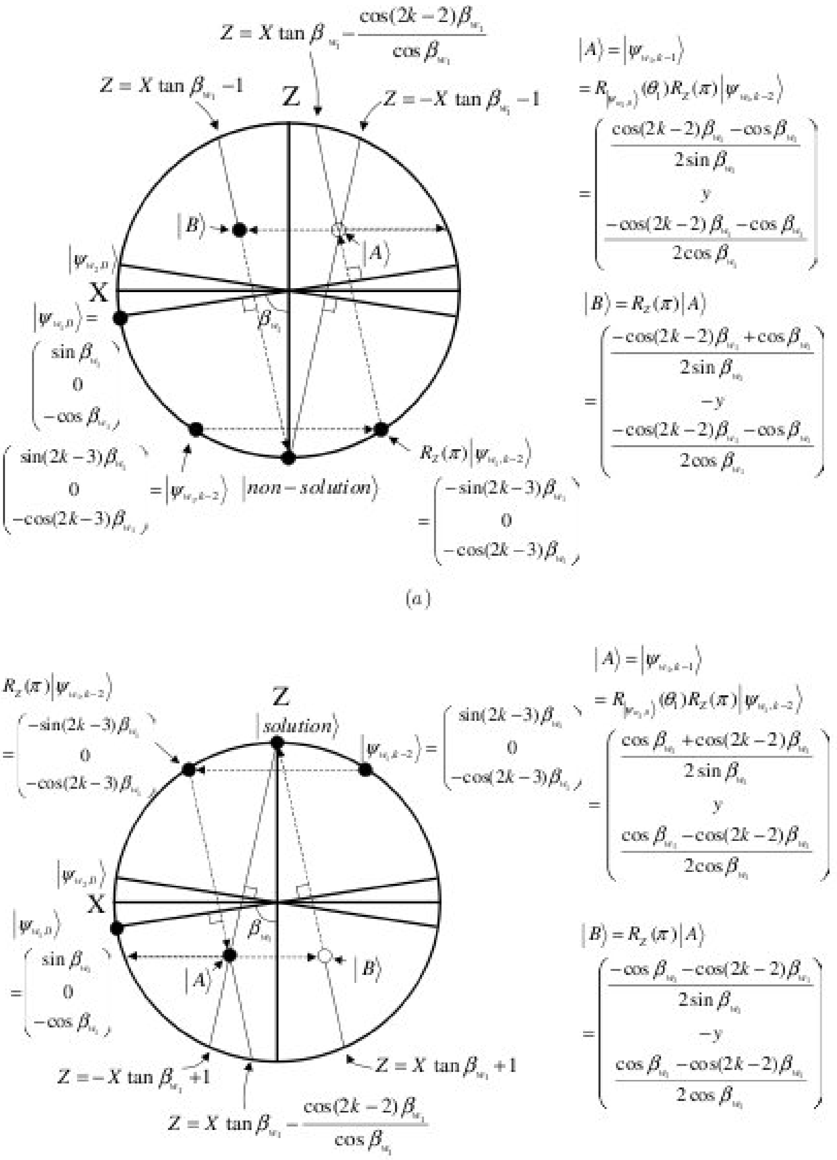
$Z = X \tan\beta_{w_1} - \frac{\cos(2k-2)\beta_{w_1}}{\cos\beta_{w_1}}$
$Z = X \tan\beta_{w_1} - 1$
Z
$Z = -X \tan\beta_{w_1} - 1$
$|A\rangle = |\psi_{w_1,k-1}\rangle = R_{|\psi_{w_1,0}\rangle}(\theta_1) R_Z(\pi) |\psi_{w_1,k-2}\rangle = \begin{pmatrix} \frac{\cos(2k-2)\beta_{w_1} - \cos\beta_{w_1}}{2\sin\beta_{w_1}} \\ y \\ \frac{-\cos(2k-2)\beta_{w_1} - \cos\beta_{w_1}}{2\cos\beta_{w_1}} \end{pmatrix}$
$|B\rangle$
$|A\rangle$
$|\psi_{w_2,0}\rangle$
X
β_{w_1}
$|\psi_{w_1,0}\rangle = \begin{pmatrix} \sin\beta_{w_1} \\ 0 \\ -\cos\beta_{w_1} \end{pmatrix}$
$|B\rangle = R_Z(\pi)|A\rangle = \begin{pmatrix} \frac{-\cos(2k-2)\beta_{w_1} + \cos\beta_{w_1}}{2\sin\beta_{w_1}} \\ -y \\ \frac{-\cos(2k-2)\beta_{w_1} - \cos\beta_{w_1}}{2\cos\beta_{w_1}} \end{pmatrix}$
$\begin{pmatrix} \sin(2k-3)\beta_{w_1} \\ 0 \\ -\cos(2k-3)\beta_{w_1} \end{pmatrix} = |\psi_{w_1,k-2}\rangle$
$|non-solution\rangle$
$R_Z(\pi)|\psi_{w_1,k-2}\rangle = \begin{pmatrix} -\sin(2k-3)\beta_{w_1} \\ 0 \\ -\cos(2k-3)\beta_{w_1} \end{pmatrix}$
(a)
$|A\rangle = |\psi_{w_1,k-1}\rangle = R_{|\psi_{w_1,0}\rangle}(\theta_1) R_Z(\pi) |\psi_{w_1,k-2}\rangle = \begin{pmatrix} \frac{\cos\beta_{w_1} + \cos(2k-2)\beta_{w_1}}{2\sin\beta_{w_1}} \\ y \\ \frac{\cos\beta_{w_1} - \cos(2k-2)\beta_{w_1}}{2\cos\beta_{w_1}} \end{pmatrix}$
$R_Z(\pi)|\psi_{w_1,k-2}\rangle = \begin{pmatrix} -\sin(2k-3)\beta_{w_1} \\ 0 \\ -\cos(2k-3)\beta_{w_1} \end{pmatrix}$
Z
$|solution\rangle$
$|\psi_{w_1,k-2}\rangle = \begin{pmatrix} \sin(2k-3)\beta_{w_1} \\ 0 \\ -\cos(2k-3)\beta_{w_1} \end{pmatrix}$
$|\psi_{w_2,0}\rangle$
X
β_{w_1}
$|\psi_{w_1,0}\rangle = \begin{pmatrix} \sin\beta_{w_1} \\ 0 \\ -\cos\beta_{w_1} \end{pmatrix}$
$|A\rangle$
$|B\rangle$
$|B\rangle = R_Z(\pi)|A\rangle = \begin{pmatrix} \frac{-\cos\beta_{w_1} - \cos(2k-2)\beta_{w_1}}{2\sin\beta_{w_1}} \\ -y \\ \frac{\cos\beta_{w_1} - \cos(2k-2)\beta_{w_1}}{2\cos\beta_{w_1}} \end{pmatrix}$
$Z = -X \tan\beta_{w_1} + 1$
$Z = X \tan\beta_{w_1} + 1$
$Z = X \tan\beta_{w_1} - \frac{\cos(2k-2)\beta_{w_1}}{\cos\beta_{w_1}}$
(b)

Figure 4. Evolution of the states for the last two Grover operations. (*a*) Even *k* and (*b*) Odd *k*.

functions with only the smaller weight case $w = w_1 < w_2$ because the phase conditions are the same for the larger weight w_2. Finally, we find the following phase conditions for θ_1 and θ_2 as shown in appendix B:

$$\cos\theta_1 = \frac{(-1)^k \cos\beta_{w_1} - \cos 2\beta_{w_1} \cos(2k-2)\beta_{w_1}}{\sin 2\beta_{w_1} \sin(2k-2)\beta_{w_1}} \tag{12}$$

$$\cos\theta_2 = \frac{(-1)^k \sin 2\beta_{w_1} (y \sin\theta_2 - (-1)^k \sin\beta_{w_1})}{\cos\beta_{w_1} \cos 2\beta_{w_1} - (-1)^k \cos(2k-2)\beta_{w_1}}. \tag{13}$$

5. Performance comparison

5.1. Classical versus quantum

Let us first consider the complexity of our approach. If w_1 and w_2 are $\sin^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)$ and $\cos^2\left(\frac{k}{2k+1}\frac{\pi}{2}\right)$, respectively, our algorithm can decide an exact weight w with $O(k)$ (in fact, exactly k) Grover operations as shown in algorithm 1 or 2.

Now we consider the complexity of any classical approach. Since the function f is available in the form of an oracle, a classical probabilistic algorithm would work as follows. For k iterations it can present random inputs to the oracle and guess that the function is of weight w_1 (w_2) if the output zero (one) appears more frequently. The approach simply decides based on the majority of outcomes, and the analysis of the majority has been made in [27]. Based on this analysis, we know that the query complexity of such a classical probabilistic algorithm is $\Omega(k^2)$. Finally, then we can see that our approach achieves at least a quadratic speedup.

5.2. Comparison with the quantum counting algorithm

To argue for the efficiency of our algorithm, let us refer to the existing work on quantum counting that exploits the period information of the repeated Grover iterations [24, 28, 29]. From this period information, one can guess the number of solutions. Hence, like Shor's factoring algorithm, this task is achieved using the quantum Fourier transform.

Because we propose here an algorithm to decide exactly the weight among two given weights, it is meaningful to compare the query complexity between our approach and a method based on quantum counting with the same promise.

5.2.1. Quantum counting. The counting problem is the task of finding the number of solutions of a given Boolean function. Now repeated Grover operations show (quasi) periodic patterns with the iteration numbers. Hence we can count the number of solutions using the quantum Fourier transform as shown in algorithm 3 [24, 28, 29]. In the algorithm, $\mathbf{G}_f = \mathbf{Q}(\mathbf{W}, f, -1, -1)$ denotes the Grover operator with the notation of [4, 5], where $\mathbf{W}$ denotes the Walsh–Hadamard transform on n qubits that maps $|0\rangle$ to $2^{-n/2}\sum_{i=0}^{2^n-1}|\mathrm{i}\rangle$, f is the given Boolean function, the first and second -1 denote $\theta = \pi$ and $\phi = \pi$, respectively. Meanwhile, an integer P determines the time taken by the algorithm, and consequently, the precision of the estimation.

Algorithm 3 Quantum counting [24, 28, 29]

```
Let  $\mathbf{C}_f : |x\rangle \otimes |\Psi\rangle \mapsto |x\rangle \otimes (\mathbf{G}_f)^x|\Psi\rangle$
Let  $\mathbf{f}_P : |k\rangle \mapsto \frac{1}{\sqrt{P}}\sum_{l=0}^{P-1} e^{2\pi\imath kl/P}|l\rangle$
1.   $|\Psi_0\rangle = \mathbf{W} \otimes \mathbf{W}|0\rangle|0\rangle$
2.   $|\Psi_1\rangle = \mathbf{C}_f|\Psi_0\rangle$
3.   $|\Psi_2\rangle = |\Psi_1\rangle$after the second register is measured (optional)
4.   $|\Psi_3\rangle = \mathbf{f}_P \otimes \mathbf{I}|\Psi_2\rangle$
5.   $\hat{x}$ = measured value of $|\Psi_3\rangle$
     (if $\hat{x} > P/2$ then $\hat{x} = (P - \hat{x})$)
6.   output:   $N\sin^2(\hat{x}\pi/P)$(and $\hat{x}$ if needed)
```

5.2.2. Comparison of query complexity. Now let us analyse the quantum counting method for the weight-decision problem. First, we must understand the relation between P and $\hat{x}$, and the given weight. If we are asked to check whether or not the given weight is correct, we should know the exact value of P and the expected value of $\hat{x}$. For example, if the given weight is $\sin^2(k\pi/(4k+2))$, then we can confirm the given weight by using $P = (4k+2)$ and by checking whether or not $\hat{x}$ is k. Likewise, if the given weight is $\cos^2(k\pi/(4k+2)) = \sin^2((k+1)\pi/(4k+2))$, then we can verify the given weight by using $P = (4k+2)$ and by checking whether or not $\hat{x}$ is $k+1$. Therefore, for our problem such as the limited weight-decision problem, the quantum counting method can be exploited by assuming $P = (4k+2)$ and by checking $\hat{x}$. More explicitly, after P queries, if $\hat{x}$ is k, we can conclude that the exact weight is w_1 and if $\hat{x}$ is $k+1$, the exact weight is w_2. In summary, when we try to use quantum counting for the weight-decision problem, it takes around $(4k+2)$ queries, but our method requires only k queries, and hence our method requires four times fewer queries than the quantum counting method.

As an aside we note that the quantum counting method is based on the quantum Fourier transform, which might be difficult to implement depending on what kind of quantum computer is being used.

In summary, compared with the quantum counting method, our method requires four times fewer queries, and does not need the quantum Fourier transform.

6. Conclusion and open problems

We investigated an application of the Grover operator for the weight analysis problem of a Boolean function, specifically the weight-decision problem, which aims to find a correct weight from the two weights satisfying $w_1 + w_2 = 1$. First, we showed what weight can be decided exactly after k Grover operators. Unfortunately, this algorithm, the limited weight-decision algorithm, is not a sure-success one for the general weight-decision problem. To overcome this problem, we modified the limited weight-decision algorithm by using a certain number of standard Grover operators from the 1st to the $(k-2)$th steps, followed by two phase-modified Grover operators for the final two steps. We note that our proposed quantum algorithm, the general weight-decision algorithm, achieves at least a quadratic speedup compared to the best classical algorithm. As well, the proposed algorithm requires four times fewer queries than the quantum counting algorithm.

In this work, we considered only a restricted type of weight-decision problem with a condition $w \in \{w_1, w_2 | 0 < w_1 < w_2 < 1, w_1 + w_2 = 1\}$ because it is significantly easier compared to a more general condition. Hence we might hope to extend our approach to a more general case such as when the sum of two weights is not unity. Unfortunately, at present, we have no idea how to generalize our proposed algorithm for this more general case. When $w_1 + w_2 = 1$, there is a symmetry between the two quantum states associated with the larger and the smaller weights. It is precisely for this reason that we need only solve phase conditions for the final two modified Grover operations. However, for a more general case, there would be no such symmetry, and hence finding of an exact algorithm would be more difficult. Meanwhile, as an intermediate step towards this more general case, we could consider a slightly generalized problem for deciding $w \in \left\{w_1, w_2 | 0 < w_1 < \frac{1}{2}, \frac{1}{2} < w_2 < 1, \frac{1}{2} < w_1 + w_2 < \frac{3}{2}\right\}$; for example, $w_1 = \frac{1}{3}$ and $w_2 = \frac{3}{4}$. Finally, more generally, we might hope to find an algorithm to decide $w \in \{w_1, w_2 | 0 < w_1 < w_2 < 1, 0 < w_1 + w_2 < 2\}$; for example, $w_1 = \frac{2}{3}$ and $w_2 = \frac{3}{4}$. In any case, even without such an efficient algorithm one could always rely on the method of quantum counting to determine the weights in these cases for a modest overhead.

Acknowledgments

The authors would very much like to thank the anonymous reviewers for improving the quality of the presentation. SLB currently holds a Royal Society–Wolfson Research Merit Award. BSC was partially supported by IT Scholarship Program supervised by IITA (Institute for Information Technology Advancement) and MIC (Ministry of Information and Communication), and currently by the Post Brain Korea 21 (Ministry of Education and Human Resources Development, Republic of Korea). This work is also partially supported by BK21 under Professor Jun Dong Cho, an advisor.

Appendix A. Proof of correctness

A.1. No crossing point until the $(k-2)th$ operation

Proof of fact 1 is as follows.

Proof. From the value of β_{w_1}, we can get the value of $-\sin 2\beta_{w_1}$ as $-\sin\left(\frac{\pi}{2k-1}\right) < -\sin 2\beta_{w_1} \leqslant -\sin\left(\frac{\pi}{2k+1}\right)$. Meanwhile, $(k-2)\pi + \frac{\pi}{2k-1} < (2k-3)\beta_{w_1} \leqslant (k-2)\pi + \frac{2\pi}{2k+1}$. Further, because k is odd in this case we have $\sin\left[(k-2)\pi + \frac{\pi}{2k-1}\right] > \sin(2k-3)\beta_{w_1} \geqslant \sin\left[(k-2)\pi + \frac{2\pi}{2k+1}\right]$. Finally, $-\sin\left(\frac{\pi}{2k-1}\right) > \sin(2k-3)\beta_{w_1} \geqslant -\sin\left(\frac{2\pi}{2k+1}\right)$. Therefore, $-\sin 2\beta_{w_1} > \sin(2k-3)\beta_{w_1}$. □

A.2. First crossing point at the $(k-1)th$ operation

Proof of fact 2 is as follows.

Proof. From fact 1, we can bound the value of $\sin 2\beta_{w_1}$ as $\sin\left(\frac{\pi}{2k-1}\right) > \sin 2\beta_{w_1} \geqslant \sin\left(\frac{\pi}{2k+1}\right)$. Meanwhile, $(k-1)\pi < (2k-1)\beta_{w_1} \leqslant (k-1)\pi + \frac{\pi}{2k+1}$. Further, because k is odd in this case we have $\sin(k-1)\pi < \sin(2k-1)\beta_{w_1} \leqslant \sin\left[(k-1)\pi + \frac{\pi}{2k+1}\right]$. Finally, $0 < \sin(2k-1)\beta_{w_1} \leqslant \sin\left(\frac{\pi}{2k+1}\right)$. Therefore, $\sin 2\beta_{w_1} \geqslant \sin(2k-1)\beta_{w_1}$. □

Appendix B. Phase conditions

At first, to rotate $|\psi_{w_1,k-2}\rangle$ to the first crossing point $|A\rangle$, θ_1 should satisfy

$$R_{|\psi_{w_1,0}\rangle}(\theta_1)\begin{pmatrix}-\sin(2k-3)\beta_{w_1}\\ 0\\ -\cos(2k-3)\beta_{w_1}\end{pmatrix} = \begin{pmatrix}\dfrac{\cos(2k-2)\beta_{w_1} - (-1)^k\cos\beta_{w_1}}{2\sin\beta_{w_1}}\\ y\\ \dfrac{-\cos(2k-2)\beta_{w_1} - (-1)^k\cos\beta_{w_1}}{2\cos\beta_{w_1}}\end{pmatrix}. \tag{B.1}$$

As a result, θ_1 should satisfy

$$\cos\theta_1 = \frac{(-1)^k\cos\beta_{w_1} - \cos 2\beta_{w_1}\cos(2k-2)\beta_{w_1}}{\sin 2\beta_{w_1}\sin(2k-2)\beta_{w_1}}. \tag{B.2}$$

Further, the value of y for the state $|A\rangle$ is $\sin\theta_1\sin(2k-2)\beta_{w_1}$. A similar approach allows us to find the phase condition for θ_2 as

$$R_{|\psi_{w_1,0}\rangle}(\theta_2)\begin{pmatrix}\dfrac{-\cos(2k-2)\beta_{w_1}+(-1)^k\cos\beta_{w_1}}{2\sin\beta_{w_1}}\\ -y\\ \dfrac{-\cos(2k-2)\beta_{w_1}-(-1)^k\cos\beta_{w_1}}{2\cos\beta_{w_1}}\end{pmatrix}=\begin{pmatrix}0\\0\\-(-1)^k\end{pmatrix}. \qquad \text{(B.3)}$$

Finally, then we can see that θ_2 satisfies the following equation

$$\cos\theta_2=\frac{(-1)^k\sin 2\beta_{w_1}\left(y\sin\theta_2-(-1)^k\sin\beta_{w_1}\right)}{\cos\beta_{w_1}\cos 2\beta_{w_1}-(-1)^k\cos(2k-2)\beta_{w_1}}. \qquad \text{(B.4)}$$

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] Shor P W 1997 *SIAM J. Comput.* **26** 1484–509

[4] Grover L K 1996 *Proc. 28th Ann. ACM Symposium on the Theory of Computing (Philadelphia, PA)* pp 212–9

[5] Grover L K 1997 *Phys. Rev. Lett.* **79** 325–8

[6] Filiol E and Fontaine C 1998 *Proceedings of Advances in Cryptology—EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques (Lecture Notes in Computer Science* vol 1403*)* pp 475–88

[7] MacWilliams F J and Sloane N J A 1996 *The Theory of Error-Correting Codes* (Amsterdam: North-Holland)

[8] Chakrabarty K and Hayes J P 1996 *J. Electron. Test., Theory Appl.* **8** 71–86

[9] Chakrabarty K and Hayes J P 1995 *IEEE Trans. VLSI Syst.* **3** 72–83

[10] Filiol E 1999 *Proceedings of IMA Conference on Cryptography and Coding (Lecture Notes in Computer Science* vol 1746*)* (Berlin: Springer) pp 70–80

[11] Maitra S and Mukhopadhyay P 2005 *Int. J. Quantum Inf.* **3** 359–70

[12] Green F and Pruim R 2001 *Inf. Process. Lett.* **80** 257–60

[13] Long G L, Tu C C, Li Y S, Zhang W L and Yan Y 2000 *J. Phys. A: Math. Gen.* **34** 861–6

[14] Long G L 2001 *Phys. Rev.* A **64** 022307

[15] Høyer P 2000 *Phys. Rev.* A **62** 052304

[16] Li D and Li X 2001 *Phys. Lett.* A **287** 304

[17] Galindo A and Martin-Delgado M A 2000 *Phys. Rev.* A **62** 062303

[18] Li C-M, Hwang C-C, Hsieh J-Y and Wang K-S 2002 *Phys. Rev.* A **65** 034305

[19] Sun Y, Long G-L and Li X 2002 *Phys. Lett.* A **294** 143

[20] Hsieh J-Y and Li C-M 2002 *Phys. Rev.* A **65** 052322

[21] Chi D P and Kim J 1999 *Chaos Solitons Fractals* **10** 1689–93
Chi D P and Kim J 1999 *Proceedings of Quantum Computing and Quantum Communications (First NASA International Conference, selected papers, QCQC'98) (Lecture Notes in Computer Science* vol 1509*)* (Berlin: Springer) pp 148–51

[22] Long G L, Li Y S, Zhang W L and Niu L 1999 *Phys. Lett.* A **262** 27

[23] Li C-M, Hwang C-C, Hsieh J-Y and Wang K-S 2002 *Phys. Rev.* A **65** 034305

[24] Brassard G, Høyer P, Mosca M and Tapp A 2002 *Quantum Computation and Information, Contemporary Mathematics* vol 305 ed S J Lomonaco Jr and H E Brandt (Providence, RI: AMS) pp 53–74

[25] Mosca M 2001 *Theor. Comput. Sci.* **264** 139–53

[26] Nayak A and Wu F 1999 *Proceedings of Symposium on Theory of Computing* pp 384–93

[27] Alonso L, Reingold E M and Schott R 1993 *Inf. Process. Lett.* **47** 253–5

[28] Boyer M, Brassard G, Høyer P and Tapp A 1998 *Fortschr. Phys.* **46** 493–505

[29] Brassard G, Høyer P and Tapp A 1998 *Proceedings of the 25th International Colloquium on Automata, Languages and Programming (Lecture Notes In Computer Science* vol 1443*)* pp 820–31