Presented in WCC 2013, Bergen, Norway, 15–19 April, 2013

Quantum algorithm to check Resiliency of a Boolean function (Extended Abstract)

Kaushik Chakraborty · Subhamoy Maitra

Abstract In this paper, for the first time, we present quantum algorithms to check the order of resiliency of a Boolean function. We first show that the Deutsch-Jozsa algorithm can be directly used for this purpose. We also point out how the quadratic improvement in query complexity over the Deutsch-Jozsa algorithm can be obtained using the well known Grover's algorithm. While the worst case quantum query complexity to check the resiliency order is exponential, we can cleverly devise a strategy so that the number of measurements are polynomial in number of input variables of the Boolean function. We also point out a subset of n-variable Boolean functions for which the algorithm works in polynomial many steps, i.e., we achieve exponential speed-up over best known classical algorithms.
Keywords: Boolean Functions, Deutsch-Jozsa Algorithm, Grover Algorithm, Measurement, Resiliency.

1 Introduction

After the introduction of Deutsch-Jozsa algorithm [4] in quantum paradigm, several works have been presented in literature to describe strategies that can distinguish Boolean functions of different weights (for example, see [1] and the references therein). Such problems are actually related to studying Walsh spectrum of Boolean functions. From cryptologic viewpoint, the concept of balancedness can be generalized with the idea of resiliency and we will consider this problem here. We will study how efficiently one can check whether a Boolean function is m-resilient or not.

Before proceeding further, let us introduce basics of Boolean functions. A Boolean function on n variables may be viewed as a mapping from $\{0,1\}^n$ into $\{0,1\}$. We will denote the set of n-variable Boolean functions as $\mathcal{B}_n$. It is easy

Kaushik Chakraborty
Indian Statistical Institute, Kolkata 700 108, India
E-mail: kaushik.chakraborty9@gmail.com

Subhamoy Maitra
Applied Statistics Unit, Indian Statistical Institute, Kolkata 700 108, India
E-mail: subho@isical.ac.in

to note that $|\mathcal{B}_n| = 2^{2^n}$. Let us denote the addition operator over $GF(2)$ by $\oplus$. Let $x = (x_1, \ldots, x_n)$ and $\omega = (\omega_1, \ldots, \omega_n)$ both belonging to $\{0,1\}^n$ and the inner product $x \cdot \omega = x_1\omega_1 \oplus \cdots \oplus x_n\omega_n$. Let $f(x)$ be a Boolean function on n variables. Then the *Walsh transform* of $f(x)$ is an integer valued function over $\{0,1\}^n$ which is defined as $W_f(\omega) = \sum_{x\in\{0,1\}^n} (-1)^{f(x)\oplus x\cdot\omega}$. The fastest known classical algorithm to calculate all the Walsh spectrum values of $f \in \mathcal{B}_n$, i.e., $W_f(\omega)$ at each of the 2^n points ω, is of $O(n2^n)$ time complexity. To calculate the Walsh spectrum value at a specific point requires $O(2^n)$ time too in classical domain.

For a binary string *str*, the number of 1's in the string is called (Hamming) weight of *str* and denoted as $wt(str)$. In truth table representation, a Boolean function $f \in \mathcal{B}_n$ can be viewed as a binary string of length 2^n, which is the output column of the truth table. If $wt(f) = 2^{n-1}$, then f is called a balanced function. In terms of Walsh spectrum, $f \in \mathcal{B}_n$ is balanced if and only if $W_f(0,0,\ldots,0) = 0$. Following [7], a function $f \in \mathcal{B}_n$ is m-resilient iff its Walsh transform satisfies $W_f(\omega) = 0$, for $0 \leq wt(\omega) \leq m$. It is easy to note that a balanced function is actually a 0-resilient function. Thus, informally speaking, the problems related to resiliency will be the generalization of the problems related to balancedness.

Before proceeding further, let us briefly discuss certain algorithms in classical as well as quantum domain. We consider that the Boolean function $f \in \mathcal{B}_n$ is available as an (classical or its quantum counterpart) oracle, i.e., the corresponding output can be obtained efficiently given the input. Given the promise that f is either constant or balanced, to check which one it is, we have Deutsch-Jozsa [4] algorithm to solve it in constant time. Though there is no polynomial time algorithm to solve this problem in classical domain, probabilistic polynomial time algorithms are indeed available to solve this problem efficiently. Given that $W_f(\omega) = 0$ or $\pm 2^n$, the question of "which one it is" can be solved exactly in a similar manner, by considering the function $f(x) \oplus \omega \cdot x$ instead of $f(x)$.

It is well known that checking resiliency of an n-variable Boolean function requires exponential time in n in classical domain. In this paper we try to analyse the solution of this problem in quantum paradigm. We note that the traditional Deutsch-Jozsa [4] can be used for this purpose. Further, we try to devise strategies with better efficiency than this using Grover algorithm [5]. It should be noted that Grover algorithm has earlier been used in weight decision problems for Boolean functions [1] and we note that a more involved application of this algorithm can also be exploited in the resiliency checking problem. The most important contribution of our work is that, though the worst case query complexity[1] of our algorithm may be exponential, we need only polynomial many measurements for this purpose. We also identify a sub class of Boolean functions for which our quantum algorithms work with polynomial many queries in n. The best known classical algorithm for this sub class requires exponential many steps.

2 Algorithm to check Resiliency

Given f is either constant or balanced, if the corresponding quantum implementation U_f is available, Deutsch-Jozsa [4] provided a quantum algorithm that decides

[1] For quantum algorithms, we write "query complexity" instead of "time complexity" as we need to query some oracles, e.g., $U_f, \mathcal{O}_g$ as described in Section 2.

in constant many queries which one it is. The overall idea of the algorithm can be summarized as in Figure 1.

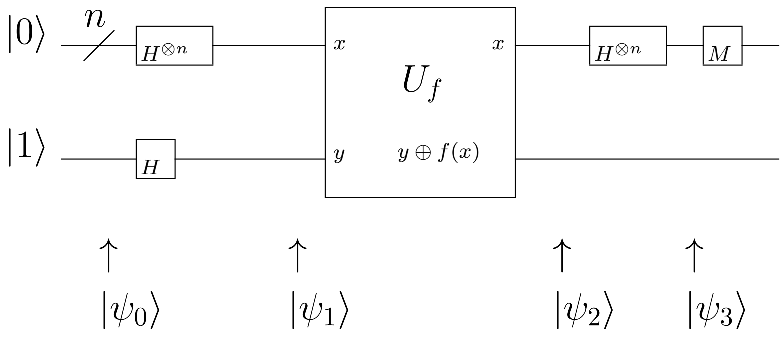

Fig. 1 Quantum circuit to implement Deutsch-Jozsa Algorithm

The step by step description of the Deutsch-Jozsa [4] algorithm can be written as follows.

```
Input: A Boolean function $f$ on $n$ variables is available in the form of the unitary
        transformation $U_f$
Output: $n$-bit pattern
Take an $(n+1)$ qubit state $|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle$;
Apply Hadamard Transform $H^{\otimes(n+1)}$ on $|\psi_0\rangle$ to get $|\psi_1\rangle = \sum_{x\in\{0,1\}^n} \frac{|x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$;
Apply $U_f$ on $|\psi_1\rangle$ to get $|\psi_2\rangle = \sum_{x\in\{0,1\}^n} \frac{(-1)^{f(x)}|x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$;
Apply Hadamard Transform on the first $n$ qubits of $|\psi_2\rangle$ to obtain
$|\psi_3\rangle = \sum_{z\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} \frac{(-1)^{x\cdot z\oplus f(x)}|z\rangle}{2^n} \left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$;
Measurement at $M$: measure the first $n$ qubits of $|\psi_3\rangle$;
After measurement, all zero state ($n$-bit all zero pattern) implies that the function is
constant, else it is balanced;
```

Algorithm 1: The Deutsch-Jozsa algorithm [4].

Let us now describe our interpretation of Deutsch-Jozsa algorithm in terms of Walsh spectrum values. We denote the operator for Deutsch-Jozsa algorithm as $\mathcal{D}_f = H^{\otimes n} U_f H^{\otimes n}$, where the Boolean function f is available as an oracle U_f. For brevity, we abuse the notation and do not write the auxiliary qubit, i.e., $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ and the corresponding output in this case[2]. Now one can observe that $\mathcal{D}_f|0\rangle^{\otimes n} = \sum_{z\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} \frac{(-1)^{x\cdot z\oplus f(x)}|z\rangle}{2^n} = \sum_{z\in\{0,1\}^n} \frac{W_f(z)}{2^n}|z\rangle$, i.e., the associated probability with a state $|z\rangle$ is $\frac{W_f^2(z)}{2^{2n}}$. In this regard, we have the following technical result as pointed out in [6].

[2] We go for similar abuse of notation for the phase inversion oracle later.

Proposition 1 *Given an n-variable Boolean function f, $\mathcal{D}_f|0\rangle^{\otimes n}$ produces a superposition of all states $z \in \{0,1\}^n$ with the amplitude $\frac{W_f(z)}{2^n}$ corresponding to each state $|z\rangle$.*

Consider that we are interested to know whether $f \in \mathcal{B}_n$ is m-resilient. Let $S_m = \{x \in \{0,1\}^n | wt(x) \leq m\}$ and $\overline{S}_m = \{x \in \{0,1\}^n | wt(x) > m\}$. Consider the n-qubit state $|\Psi\rangle = \sum_{s \in S_m} \frac{W_f(s)}{2^n}|s\rangle + \sum_{s \in \overline{S}_m} \frac{W_f(s)}{2^n}|s\rangle$. For brevity, let us represent $|\Psi\rangle = a|X\rangle + b|Y\rangle$. That is, $a^2 = \sum\limits_{s \in S_m} \frac{W_f^2(s)}{2^{2n}}$ and $b^2 = \sum\limits_{s \in \overline{S}_m} \frac{W_f^2(s)}{2^{2n}}$.

Using the Deutsch-Jozsa algorithm, we obtain $\sum_{z \in \{0,1\}^n} \frac{W_f(z)}{2^n}|z\rangle$ (before the measurement) and $a^2 = \sum\limits_{s \in S_m} \frac{W_f^2(s)}{2^{2n}}$. That is, some state $s \in S_m$ will appear after the measurement with probability a^2. Hence, in expected $O(\frac{1}{a^2})$ iterations, one can observe some $s \in S_m$ after the measurement and output that f is not m-resilient. If f is indeed m-resilient, then $a^2 = 0$ and thus any state $s \in S_m$ will never appear at the output. One may note that the minimum absolute value of Walsh spectrum is 2 and thus, we can have a situation that f is not m-resilient, but a^2 is $O(\frac{1}{2^{2n}})$. In such a case, the algorithm will require exponential many queries to provide the correct result. Thus the resiliency checking algorithm is as follows.

```
Input: A Boolean function $f$ on $n$ variables is available in the form of the unitary
        transformation $U_f$, order of resiliency $m$ and the number of iteration $r$
Output: $n$-bit pattern
$S_m = \{x \in \{0,1\}^n | wt(x) \leq m\}$;
for $r$ many times do
    Apply Deutsch-Jozsa algorithm and take the $n$-bit output $u$;
    if $u \in S_m$ then
        Report that the function is not $m$-resilient (NO) and terminate;
    end
end
Report that the function is $m$-resilient (YES);
```

Algorithm 2: Resiliency checking using the Deutsch-Jozsa algorithm [4].

Theorem 1 *Let c be a predefined constant. Algorithm 2 correctly answers NO, but answers YES with success probability greater than or equal to c, in r many steps, where r is $O(\frac{1}{a^2})$ and $a^2 = \sum\limits_{s \in S_m} \frac{W_f^2(s)}{2^{2n}} > 0$.*

Proof According to Algorithm 2, one can observe that for each iteration, the success probability is a^2. At i-th step, the success probability will be $1 - (1 - a^2)^i$. So, at $i = r$ the success probability will become $1 - (1 - a^2)^r = c$. Now solving this equation we get r is $O(\frac{1}{a^2})$. □

Remark 1 Algorithm 2 is written in such a manner that if a function is indeed m-resilient, then $a = 0$ and thus the algorithm will say YES after executing r many steps. However, it is known that for nonzero Walsh spectrum values, the minimum is ±2 and thus, $a^2 \geq \frac{4}{2^{2n}}$. Hence, after repeating the algorithm r, i.e., $O(2^{2n})$ many times, if we don't observe any binary string $u \in S_m$ after measurements, then we can conclude that the Boolean function f is m-resilient with success probability greater than some predefined constant c. This provides the worst case scenario.

2.1 Improvement using Grover Algorithm

Grover algorithm [5] provides a quadratic speed-up compared to repeated use of Deutsch-Jozsa algorithm and that is the motivation we try out here. Instead of equal superposition $|\psi\rangle = H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{2^{\frac{n}{2}}} \sum_{x \in \{0,1\}^n} |x\rangle$ in Grover algorithm, we will use the state of the form $|\Psi\rangle = \mathcal{D}_f(|0\rangle^{\otimes n}) = \sum_{x \in \{0,1\}^n} \frac{W_f(x)}{2^n}|x\rangle$.

Consider that any n-qubit state is represented in the computational basis. We want to amplify the amplitude at the points in S_m. This we achieve in a similar manner as in Grover algorithm.

The Grover algorithm requires inversion of phase. Towards this, we will use $g(x) \in \mathcal{B}_n$, different from $f(x)$. The corresponding operator $\mathcal{O}_g$ inverts the phase of the states $|x\rangle$ where $x \in S_m$. That is, we need to change phase for the points having weight less than or equal to m. This can be achieved by choosing the n-variable Boolean function $g(x)$ such that $g(x) = 1$, when $wt(x) \leq m$, and $g(x) = 0$, otherwise. Thus g is a symmetric function. A symmetric Boolean function can be efficiently implemented, as described in [3]. The circuit complexity of an n-variable symmetric Boolean function is $4.5n + o(n)$. It is known that given a classical circuit g, a quantum circuit of comparable efficiency can be implemented. Thus, we will consider that for a symmetric function g, the quantum circuit $\mathcal{O}_g$ can be efficiently implemented using $O(n)$ circuit complexity.

Now let us consider the operator $G_t = [(2|\Psi\rangle\langle\Psi| - I)\mathcal{O}_g]^t$ on $|\Psi\rangle$ to get $|\Psi_t\rangle$. The idea presented in the following result is similar to amplitude amplification for constructing Dicke states as presented in [2]. However, we present the proof for better understanding.

Theorem 2 *Let* $|\Psi\rangle = \sum_{s \in S_m} \frac{W_f(s)}{2^n}|s\rangle + \sum_{s \in \overline{S}_m} \frac{W_f(s)}{2^n}|s\rangle = a|X\rangle + b|Y\rangle$, *where* $a = \sin\theta$, $b = \cos\theta$. *The application of* $[(2|\Psi\rangle\langle\Psi| - I)\mathcal{O}_g]^t$ *operator on* $|\Psi\rangle$ *produces* $|\Psi_t\rangle$, *in which the probability amplitude of* $|X\rangle$ *is* $\sin(2t+1)\theta$.

Proof For $t = 1$, one can check that

$$|\Psi_1\rangle = [(2|\Psi\rangle\langle\Psi| - I)\mathcal{O}_g]|\Psi\rangle = [(2|\Psi\rangle\langle\Psi|)\mathcal{O}_g]|\Psi\rangle - \mathcal{O}_g|\Psi\rangle.$$

Now substituting the values of a, b we get that $|\Psi_1\rangle = \sin 3\theta|X\rangle + \cos 3\theta|Y\rangle$.

Now we will use induction. Let the application of $[(2|\Psi\rangle\langle\Psi| - I)\mathcal{O}_g]^t$ operator on $|\Psi\rangle$ updates the probability amplitude of $|X\rangle$ as $\sin(2t\theta + \theta)$, for $t = k$. From the assumption we have $[(2|\Psi\rangle\langle\Psi| - I)\mathcal{O}_g]^k|\Psi\rangle = \sin(\theta + 2k\theta)|X\rangle + \cos(\theta + 2k\theta)|Y\rangle$. Now for $t = k+1$, it can be checked that

$$[(2|\Psi\rangle\langle\Psi| - I)\mathcal{O}_g]^{(k+1)}|\Psi\rangle = \sin(\theta + 2(k+1)\theta)|X\rangle + \cos(\theta + 2(k+1)\theta)|Y\rangle.$$

Thus, the proof. □

After the Deutsch-Jozsa algorithm we obtain $\sum_{z\in\{0,1\}^n} \frac{W_f(z)}{2^n}|z\rangle$ (before the measurement) with $a^2 = \sum\limits_{s\in S_m} \frac{W_f^2(s)}{2^{2n}}$ and $b^2 = \sum\limits_{s\in \overline{S}_m} \frac{W_f^2(s)}{2^{2n}}$. Thus, we have $\sin\theta = a$. For large n, one can approximate it as $\theta = a$ and hence we need t iterations of Grover like strategy such that $(2t+1)\theta \geq \sin^{-1} c$, where c is a predefined constant. Thus, here we need an expected $O(\frac{1}{a})$ iterations, compared to $O(\frac{1}{a^2})$ iterations using the Deutsch-Jozsa algorithm only.

```
Input: A Boolean function $f$ on $n$ variables is available in the form of the unitary
       transformation $U_f$, order of resiliency $m$ and the number of iteration $r$ and a
       series of positive integers $t_i$, $1 \leq i \leq r$ related to number of Grover iteration
Output: $n$-bit pattern
$S_m = \{x \in \{0,1\}^n | wt(x) \leq m\}$;
for $i = 1$ to $r$ do
    Apply Deutsch-Jozsa algorithm till the step before measurement to obtain
    $|\Psi\rangle = \sum_{s\in S_m} \frac{W_f(s)}{2^n}|s\rangle + \sum_{s\in \overline{S}_m} \frac{W_f(s)}{2^n}|s\rangle$;
    By applying Grover iteration, obtain $|\Psi_{t_i}\rangle = [(2|\Psi\rangle\langle\Psi| - I)\mathcal{O}_g]^{t_i}|\Psi\rangle$;
    Measure $|\Psi_{t_i}\rangle$ in computational basis to obtain $n$-bit string $u$;
    if $u \in S_m$ then
        Report that the function is not $m$-resilient (NO) and terminate;
    end
end
Report that the function is $m$-resilient (YES);
```

Algorithm 3: Resiliency checking using the Grover algorithm [4].

One important issue here is that any estimate of a may not be known and thus, estimating t_r could be challenging. Given that t_r can be estimated, after application of Grover's algorithm, we will obtain a state $\sum\limits_{s\in S_m} a'_s|s\rangle + \sum\limits_{s\in \overline{S}_m} b'_s|s\rangle = a'|X\rangle + b'|Y\rangle$, where $(a')^2$ is very close to 1. Using this (Grover algorithm followed by Deutsch-Jozsa algorithm), we get a quadratic speed-up over just using Deutsch-Jozsa algorithm.

It is natural to use Grover algorithm for amplitude amplification and thus obtaining quadratic speed-up. However, in the known applications (e.g., search), the number of target states for which the amplitude is increased are not large. That guarantees the efficient implementation of the phase reversal circuit. In this case, the situation is different as we need to amplify the amplitude at $\sum_{i=0}^{m} \binom{n}{i}$ many points of weight $\leq m$ and this could be exponential. Thus, it is an important question whether the phase reversal can be implemented efficiently. In this case, this can be achieved as the phase reversal can be implemented with symmetric functions, the implementation of which is efficient [3].

2.2 Deciding the numbers of Grover iteration

Now let us explicitly describe how one can decide the values of t_i for $1 \leq i \leq r$. As given in Algorithm 3, we have $|\Psi\rangle = \sum_{s\in S_m} \frac{W_f(s)}{2^n}|s\rangle + \sum_{s\in \overline{S}_m} \frac{W_f(s)}{2^n}|s\rangle = a|X\rangle + b|Y\rangle$, where $a = \sin\theta$, $b = \cos\theta$. Our motivation is to observe some state

$s \in S_m$, if $\sum_{s\in S_m} \frac{W_f(s)}{2^n} > 0$, i.e., if $a > 0$. We will apply Grover algorithm to obtain $|\Psi_{t_i}\rangle = [(2|\Psi\rangle\langle\Psi| - I)\mathcal{O}_g]^{t_i}|\Psi\rangle = \sin\theta_i|X\rangle + \cos\theta_i|Y\rangle$ such that $\sin\theta_i$ is greater than or equal to some predefined constant, say $\sin\theta_c = c$.

Note that $\theta_1 = \theta$, and we need $t_1 = 0$, i.e., in this case, we do not apply the Grover algorithm at all and the situation is similar to Algorithm 2 where only Deutsch-Jozsa algorithm will be used. As we do not know the value of θ, we need to try for different values of t_i, $1 \le i \le r$ such that in one of those cases, $\theta_c \le \theta_i \le \pi - \theta_c$.

We divide the region $[0, \frac{\pi}{2}]$ in $r+1$ many parts, $\alpha_{r+1}, \alpha_r, \alpha_{r-1}, \ldots, \alpha_1$ (in ascending order). There must exist some $i \in [1, r]$ such that $\alpha_{i+1} \le \theta < \alpha_i$. In the i-th step, we assume that $\alpha_{i+1} \le \theta \le \alpha_i$. Thus, in this step we require the minimum t_i such that $\theta_c = (2t_i+1)\alpha_{i+1} \le (2t_i+1)\theta \le (2t_i+1)\alpha_i = \pi - \theta_c$. Thus we need

$$(2t_i + 1)\alpha_i = \pi - \theta_c, \tag{1}$$

$$(2t_i + 1)\alpha_{i+1} = \theta_c. \tag{2}$$

Similarly, we have

$$(2t_{i-1} + 1)\alpha_{i-1} = \pi - \theta_c, \tag{3}$$

$$(2t_{i-1} + 1)\alpha_i = \theta_c. \tag{4}$$

Thus, from (1), (4), we get,

$$\frac{2t_i + 1}{2t_{i-1} + 1} = \frac{\pi - \theta_c}{\theta_c}. \tag{5}$$

Taking the initial condition $t_1 = 0$ and by solving the above recurrence relation, we get,

$$(2t_i + 1) = \frac{(\pi - \theta_c)^{(i-1)}}{\theta_c^{(i-1)}} \tag{6}$$

Now the question is how many times we have to continue this process or what should be the value of r. To answer this question we have to consider in the worst case given the value of $\sin\theta$. Let $\sin\theta = a$. From Theorem 2 we know that to ensure $\sin\theta$ to be close to 1, the maximum value among t_i's, i.e., t_r according to our technique, should be taken as $O(\frac{1}{a})$. So, $(2t_r + 1) \approx \frac{1}{a}$ and we can write $r \approx \log_{\frac{\pi-\theta_c}{\theta_c}}(\frac{1}{a})$, i.e., r is $O(\log\frac{1}{a})$. Thus, we have the following result.

Theorem 3 *Let c be a predefined constant. Algorithm 3 correctly answers NO, but answers YES with success probability greater than or equal to c, in r, i.e., $O(\log\frac{1}{a})$ many steps and the number of times the Grover operator is executed is $O(\frac{1}{a})$ where $a^2 = \sum\limits_{s\in S_m} \frac{W_f^2(s)}{2^{2n}}$.*

Proof How we estimate r is explained above. In Algorithm 3, at the i-th step we apply the operator $[(2|\psi\rangle\langle\psi| - I)\mathcal{O}_g]$, t_i times. Here i varies from 1 to r. So, the total number of times the Grover operator is applied is $T = \sum\limits_{i=1}^{r} t_i$. From (6), we can substitute the value of t_i and get $T = \frac{1}{2}\sum\limits_{i=1}^{r}(\frac{(\pi-\theta_c)^{(i-1)}}{\theta_c^{(i-1)}} - 1)$. By solving this equation and also substituting the value of r we get,

$$T \approx \frac{1}{2}[\frac{1/a - 1}{(\pi - \theta_c)/\theta_c - 1} - \frac{1}{2}\{\log_{\frac{\pi-\theta_c}{\theta_c}}(\frac{1}{a})(\log_{\frac{\pi-\theta_c}{\theta_c}}(\frac{1}{a}) + 1)\}]. \tag{7}$$

So, the number of times the Grover operator is executed is $O(\frac{1}{a})$. □

Remark 2 Similar to Remark 1, for Algorithm 3 we need to consider the case when the function is m-resilient, i.e., $a = 0$. In this case r will be $O(\log 2^n)$, i.e., $O(n)$ and t_r will be $O(2^n)$, that provides the worst case scenario.

Remark 3 We like to point out that the number of measurements in both Algorithm 2 and Algorithm 3 are r. In case of Algorithm 2, r is $O(\frac{1}{a^2})$ and can be exponential in n worst case. However, for Algorithm 3, r is $O(\log \frac{1}{a})$, which is polynomial in n in worst case. For Algorithm 2, the number of queries using the Deutsch-Jozsa operator is $r = O(\frac{1}{a^2})$ and for Algorithm 3, the number of queries using the Grover operator is $T = O(\frac{1}{a})$ and both of them could be exponential in the worst case. In summary,

- in terms of number of queries, Algorithm 3 provides quadratic improvement over Algorithm 2, though both can be exponential in worst case;
- in terms of number of measurements, Algorithm 3 requires polynomial many measurements in worst case, while Algorithm 2 requires exponential many.

3 Checking m-resiliency among functions with three valued Walsh spectrum

Form the analysis in the previous section, we note that the Deutsch-Jozsa algorithm or the Deutsch-Jozsa algorithm (without measurement) followed by the Grover algorithm can be used to check whether a Boolean function is m-resilient or not. It is very clear that the second strategy provides a quadratic speed-up over the first one. It is also evident that the quantum algorithms, in worst case, may take exponential queries in n. Thus, it would be interesting to consider a class of Boolean functions for which the problem can be solved in polynomial many queries in n in quantum paradigm.

In [9–11], several characterizations and constructions of resilient functions have been presented. In particular, it has been pointed out in [9] that the Walsh spectrum values of any m-resilient function will be divisible by 2^{m+2}. In this direction, we will concentrate on Boolean functions with Walsh spectrum values multiple of 2^{m+2}. Let us define

$$\mathcal{A}_n = \{f \in \mathcal{B}_n | W_f(\omega) \equiv 0 \bmod 2^{m+2}\}.$$

In this case, if the function is not m-resilient, then $a \geq \frac{2^{m+2}}{2^n}$ and thus, the query complexity of checking resiliency is $O(2^{n-m-2})$ using Algorithm 3. In case, $m \geq n - O(poly(\log n))$, it is clear that the query complexity of checking resiliency is $O(poly(n))$.

We do not know of any classical algorithm that can efficiently decide whether a function $f \in \mathcal{A}_n$ is m-resilient. Thus, we get an exponential speed-up in this case using quantum algorithm over classical ones.

4 Conclusion

For the first time, we study the problem of checking resiliency of a Boolean function in quantum paradigm. We try to obtain algorithms where the input is an n-variable, m-resilient Boolean function and the algorithm should output YES if the function is m-resilient and NO if it is not. Our algorithm provides the NO answer correctly, while the YES answer with probability greater than some predefined constant. We use the well known Deutsch-Jozsa and Grover algorithms for the purpose. Algorithm 3 shows that it requires exponential many queries but polynomial many measurements in n in the worst case. We also identify a subclass of Boolean functions for which we require polynomial many queries as well as polynomial many measurements in the worst case. For such a class no efficient classical algorithm is known. A more elaborate study for such sub-classes will be presented in the full version of this paper.

References

1. S. L. Braunstein, B.-S. Choi, S. Ghosh and S. Maitra. Exact quantum algorithm to distinguish Boolean functions of different weights. Journal of Physics A: Mathematical and Theoretical, Volume: 40, Pages 8441-8454, doi:10.1088/1751-8113/40/29/017, published: 3 July 2007.
2. K. Chakraborty, B.-S. Choi, A. Maitra and S. Maitra. Efficient quantum algorithm to construct arbitrary Dicke states. Available at `http://arxiv.org/abs/1209.5932`
3. E. Demenkov, A. Kojevnikov, A. Kulikov and G. Yaroslavtsev, New upper bounds on the Boolean circuit complexity of symmetric functions. Information Processing Letters 110 (2010) 264–267.
4. D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation. Proceedings of Royal Society of London, A439:553–558 (1992).
5. L. Grover, A fast quantum mechanical algorithm for database search. In *Proceedings of 28th Annual Symposium on the Theory of Computing (STOC)*, May 1996, pages 212–219. Available at `http://xxx.lanl.gov/abs/quant-ph/9605043`.
6. S. Maitra and P. Mukhopadhyay, Deutsch-Jozsa Algorithm Revisited in the Domain of Cryptographically Significant Boolean Functions. In International Journal on Quantum Information, Pages 359–370, Volume 3, Number 2, June 2005.
7. X. Guo-Zhen and J. Massey. A spectral characterization of correlation immune combining functions. *IEEE Transactions on Information Theory*, 34(3):569–571, May 1988.
8. C. H. Bennett, E. Bernstein, G. Brassard, U. Vazirani, The strengths and weaknesses of quantum computation, SIAM Journal on Computing 26(5): 15101523 (1997).
9. P. Sarkar and S. Maitra. Nonlinearity bounds and constructions of resilient Boolean functions. In *Advances in Cryptology - CRYPTO 2000*, number 1880 in Lecture Notes in Computer Science, pages 515–532. Springer Verlag, 2000.
10. Y. V. Tarannikov. On resilient Boolean functions with maximum possible nonlinearity. In *Progress in Cryptology - INDOCRYPT 2000*, number 1977 in Lecture Notes in Computer Science, pages 19–30. Springer Verlag, 2000.
11. Y. Zheng and X. M. Zhang. Improved upper bound on the nonlinearity of high order correlation immune functions. In *Selected Areas in Cryptography - SAC 2000*, number 2012 in Lecture Notes in Computer Science, pages 264–274. Springer Verlag, 2000.