Detection and diagnosis of single faults in quantum circuits

Debajyoti Bera

***Abstract*—Detection and isolation of faults is a crucial step in the physical realisation of quantum circuits. Even though quantum gates and circuits compute reversible functions, the standard techniques of automatic test pattern generation (ATPG) for classical reversible circuits are not directly applicable to quantum circuits. For faulty quantum circuits under the widely accepted single fault assumption, we show that their behaviour can be fully characterised by the (single) faulty gate and the corresponding fault model. This allows us to efficiently determine test input states as well as measurement strategy for fault detection and diagnosis. Building on top of these, we design randomised algorithms which are able to detect every non-trivial single-gate fault with minimal probability of error. We also describe similar algorithms for fault diagnosis. We evaluate our algorithms by the number of output samples that needs to be collected and the probability of error. Both of these can be related to the eigenvalues of the operators corresponding to the circuit gates. We experimentally compare all our strategies with the state-of-the-art ATPG techniques for quantum circuits under the "single missing faulty gate" model and demonstrate that significant improvement is possible if we can exploit the quantum nature of circuits.**

***Index Terms*—ATPG, fault diagnosis, test generation, testing**

I. Introduction

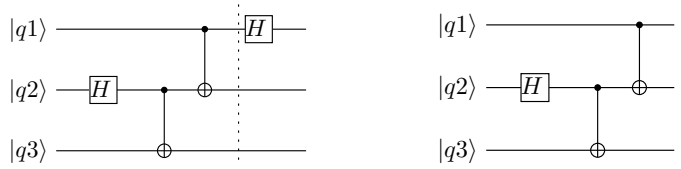
DETECTION and diagnosis of faults in classical digital circuits have been part of mainstream circuit manufacturing research and industry for several decades. A common approach for this is to analyse outputs when a circuit is given a fixed set of carefully chosen input patterns (also known as test patterns or vectors). ATPG (short for "automatic test pattern generator") based techniques essentially try to efficiently generate a "test set" – an effective set of such inputs. The intent to is to determine the fault site (or simply establish existence of a fault) by observing the output of the circuit on these input. If an input exists for a particular fault, then the fault is said to be *testable*, otherwise, undetectable (also called as *redundant*) [1], [2]. The goal of ATPG is to generate at least one input for each testable fault. Apart from the fraction of total faults that a test set covers (also known as "fault coverage"), an ATPG technique can be evaluated on other parameters like, size of its test set, time complexity of test set generation, and running time of the testing algorithm. The problem of test set generation is in general computationally challenging; determining if a fault is testable or redundant is in fact an NP-complete problem (similar to Boolean formulæ satisfiability) [3]. However, several algorithms have been proposed over the years, starting from Roth's D-Algorithm in 1966, which are based on satisfiability, back-propagation techniques, convex optimisation, etc. These have been found to be practically efficient and are now well-accepted in VLSI.

The two main components of ATPG are *fault-activation* and *fault-propagation* [4]. Fault activation requires choosing an input pattern to generate suitable logical values carried by wires at the fault-site and fault-propagation requires that any difference of value at the fault-site should also lead to a difference at the output. Even though these steps are computationally hard for arbitrary classical circuits; however, they are easy for classical reversible circuits [5]–[8]. A reversible circuit is a logic circuit which does not allow any fan-out and is composed of reversible gates. Reversible gates are logic gates whose output vector is a permutation of the input vector. Two examples of reversible gates are the NOT gate and the 2-input CNOT gate that maps logic values (a, b) to $(a, b \oplus a)$. As a result, reversible circuits have two crucial properties that general (irreversible) circuits do not possess, *controllability* (any value at any set of wires can be generated by a suitable input vector) and *observability* (any single fault that changes the values of any set of wires will change the final output) [9]. Both these properties make fault-activation and fault-propagation quite straight-forward in classical reversible circuits under the single-gate fault model. As a result, reversible circuits do not have any untestable single-gate faults and generating a test-set is rather simple.

Quantum computing has advanced a lot in the last decade, both on the algorithmic front as well as physical realisation. Extremely efficient simulators are now available for common use [10]. Several groups have reported successful implementation of important quantum gates in hardware [11], [12]. While the jury is still out on the "right model" of quantum computing, we believe that one of the successful models could be that of quantum circuits constructed out of basic quantum gates. Theoretically a quantum circuit may have endless faults, but practically, any method of fabricating a circuit limits the possible set of faults. For this work, we chose a commonly used fault-model that is based on the "single-fault assumption", i.e., the cause of a circuit failure is attributed to only one faulty gate. ATPG is an obvious avenue to explore single-fault detection in quantum circuits; however, current results on ATPG for quantum circuits are few and do not seem to fully exploit the quantum nature of these circuits [13]–[15].

Quantum circuits are structurally similar to reversible circuits, with quantum gates in place of reversible gates. The operators corresponding to the quantum gates are unitary and hence the gates are reversible. Quantum circuits, therefore, also have the same controllability and observability properties

D. Bera is with Indraprastha Institute of Information Technology (IIIT-Delhi), Okhla Phase-III, New Delhi 110020, India. E-mail: dbera@iiitd.ac.in



(a) Teleportation circuit with two Hadamard gates and two $CNOT$ gates

(b) Teleportation circuit with one H gate missing

Fig. 1. Quantum teleportation circuit and its faulty version (SMGF model)

of the classical reversible circuits. However, it has been noted by some researchers that conventional methods used for classical (reversible) circuits may not be directly applicable to quantum circuits [4] which is the primary motivation behind this work. Our goal was to study the primary questions related to testing of quantum circuits in the realm of single-gate faults. *Can a quantum circuit have untestable faults? How to generate a test for any particular fault? What should be a testing method to detect any testable fault? What is the best way to identify the site of a fault?*

A. Background and Related Work

Figure 1a illustrates a simple quantum circuit (for quantum teleportation) that acts on three qubits and is composed of first an H (Hadamard) gate on the second qubit, followed by a CNOT gate on the second and third qubits, followed by another CNOT gate on the first and second qubits and finally another H gate on the first qubit. Input to the circuit is fed at its left end in the form of some "quantum state" over three qubits and the state of the qubits at the right end is called as the "output state". The state of the qubits at any stage (input, output or intermediate) can be seen as a linear combination ("superposition") of "basis states" (equivalent to bit vectors) of the form $\alpha_0 |000\rangle + \alpha_1 |001\rangle + \alpha_2 |010\rangle + \alpha_3 |011\rangle + \alpha_4 |100\rangle + \alpha_5 |101\rangle + \alpha_6 |110\rangle + \alpha_7 |111\rangle$ where α_i's are complex numbers such that $\sum_i |\alpha_i|^2 = 1$. However, a quantum state can never be "observed" as they are; instead, if measured, a state in a superposition of basis states like the above will "collapse" into one of the observable basis states with certain probability. So, even if the input state to the circuit is fixed, the observed output may change every-time its output state is measured. The probability distribution of the different output states will be referred to as the "output distribution".

Fault detection in quantum circuits requires different techniques because unlike classical circuits which acts on bit vectors, quantum gates and circuits act on quantum states which are, mathematically, complex vectors of unit length. Therefore existing techniques used to select a suitable bit vector as a test pattern, like Boolean satisfiability [7] or integer programming [9], become inapplicable.

Even more difficulty arises since an observation of the output of a quantum circuit is merely a state from its output distribution. For example, suppose we run the teleportation circuit on the input state $|100\rangle$. It can be calculated that the output state, when measured "in the standard basis", would yield any of these four states $|010\rangle$, $|110\rangle$, $|001\rangle$, $|101\rangle$ with equal probability. However, if the H gate on the first qubit is missing (as in the circuit illustrated in Figure 1b), then standard measurement of the output state on the same input state $|100\rangle$ would yield any one of $|110\rangle$, $|101\rangle$ with equal probability. Therefore, we cannot be sure which circuit it is in case we observe $|110\rangle$ or $|101\rangle$ as the output. To resolve this kind of problem we measure the output states in non-standard basis, something that is quite unique to quantum circuits, such that the output distributions of the two situations have little in common. In any case, measurement outcomes of quantum circuits are probabilistic and so, appropriate randomised analysis of the outcomes is necessary. In fact, Perkowski et al. first highlighted the necessity of probabilistic testing for quantum circuits and suggested the use of measurement in a non-standard basis [13], [16]. However, they used a greedy heuristic and furthermore, used standard basis vectors as input; we show later that these are not very effective in fault testing.

Biamonte et al. [17] studied many fault models for a subclass of quantum circuits (built only using CNOT gates). In a later work, Hayes et al. [5] suggested that the "missing gate fault" (MGF) model is more suitable for quantum circuits; however they also considered circuits built using CNOT gates. In the MGF model one or more quantum gates are missing; such missing gates can also be modelled by replacing them by an "identity gate" that keeps its input state unchanged. The work that is closest to our work for general quantum circuits was done by Paler et al. where they studied fault detection and diagnosis under the "single missing gate fault" (SMGF) model [15]. In this work, basis states were used as input states and output states were measured in the standard basis[1]. When a circuit is run many times on the same input state and the output states are measured, we obtain a (probability) distribution on the observations which is called a *tomogram*. The authors first used a quantum circuit simulator to numerically obtain the output distributions for both the fault-free and the faulty circuits. Then they ran the given circuit many times to generate a tomogram. They analysed the tomogram with respect to the two distributions to identify whether the circuit is circuit is faulty or fault-free. However, there is no reason to believe that arbitrarily selecting input states or measuring output states in the standard basis is the best way to go about testing quantum circuits. Our work adds support to this claim by suitably choosing input states and measurement operators as part of test sets. This was also suggested by Biamonte et al. [17] but for different fault models and different types of circuits.

The "LRM technique" suggested by Paler [18] bears a resemblance to our methods. In this technique, two sets of gates, S_{pre} and S_{post}, are applied to the circuit before and after all other gates are applied. These gates are chosen to handle certain gates (like the Phase gate) that only modify some (non-global) phase. Application of S_{pre} is equivalent to choosing a non-$|0\ldots0\rangle$ input state whereas application of S_{post} is akin to choosing a non-standard measurement operator. The LRM technique prescribes the same S_{pre} and S_{post} gates irrespective of the circuit to be tested, whereas, our method can also be viewed as a selection of the optimal gates for every circuit.

[1]The authors do not mention specific measurement operators. Our inference is based on the fact that they used the quantum circuit simulator QuIDDPro which is only able to measure in standard basis.

B. Contribution

Our work answers four major questions regarding single gate faults in quantum circuits. First, we resolve the question of detectability of single gate faults. This problem is NP-hard for irreversible logic circuits and trivial for reversible ones. We show that the detection of any such fault in a quantum circuit using tomograms is inherently error-prone due to the probabilistic nature of output. But nevertheless, all faults are detectable with some positive probability. Since a missing gate can be modelled by replacing the gate by an identity gate, if the operator for a gate is almost similar to the identity operator (e.g., $R_z(\pi/2^{12})$ that maps $|0\rangle \rightarrow |0\rangle$ and $|1\rangle \rightarrow \exp(i\pi/2^{12})\,|1\rangle$, then detecting whether it is *present or missing* is going to be difficult, if not impossible (both the faulty and fault-free circuits behave almost identically). However, formal justification of this statement was not available so far which we provide here.

Next, we describe how to generate test sets for a circuit by solving quadratic programs and numerically simulating the circuit. Our test set consists of one input state and one measurement operator for each gate in the circuit. Integer linear programs have been used to generate test sets earlier [9], however, the size of those programs are exponential on the number of input-output bits. In contrast, the size of our quadratic program is exponential only on the number of inputs of the gate which is quite small in practice (usually at most three). Using high performance quantum circuit simulators, it is possible to generate test sets for any quantum circuit.

Finally, we provide algorithmic solutions for testing a circuit for fault and diagnosing which gate could be faulty by generating and analysing tomograms obtained by running the tests in a test set. We use ideas from hypothesis testing to design randomised algorithms which cleverly apply the tests in a careful manner to minimise any chance of error. A notable improvement compared to the state-of-the-art technique is with respect to detection of missing gates that apply a phase-change to only some input states, e.g., the Phase gate (that maps $|1\rangle \rightarrow i\,|1\rangle$ but leaves $|0\rangle$ unchanged). It was earlier reported that it is difficult to detect if such gates are missing and the "LR modifier technique (with slicing)" was proposed to handle them [15], [18]. First, we show that no formal relationship exists between detectability of a gate and its phase-change behaviour. For example, we show that a Pauli-Z gate can be detected easily, a Phase gate can be detected with small effort and a rotation gate $R_z(\pi/2^{12})$ gate is actually hard to detect (refer to Table I) – all three are gates which only modify phases of input states. Secondly, we wanted to avoid "slicing" since it requires testing portions of a circuit which may not be possible in some implementations. Finally, we empirically show that our approach, without any special handling of such gates, is more efficient and superior than the LR modifier technique in detecting Phase and similar gates.

The analytic and experimental results in this paper demonstrate that it is possible to achieve high efficiency in fault detection and diagnosis even for quantum circuits if the classical techniques are carefully adopted keeping in mind the quantum nature of the circuits.

C. Organisation

Following is the plan for the rest of this paper. We formally state our problem and highlight the main results in Section II. Section III contains the main technical tools and Section IV describes the essential tables and subroutines on which our algorithms are developed. The fault detection algorithms are presented in Section V where we also prove optimality properties of our tests. We similarly present and theoretically analyse our fault diagnosis algorithm in Section VI. In Section VII, we evaluate the performance of our algorithms in comparison with the best known approach.

For background on quantum circuits, we refer the reader to the excellent book by Nielsen and Chuang [19]. ATPG for classical circuits has been around for a while and any book on ATPG (e.g., by Bushnell and Agrawal [2]) can be consulted.

II. Summary of results

Since quantum circuits have inherently probabilistic output so any tomogram-based method must be prepared to handle erroneous solutions. Our detection and diagnosis algorithms require an input parameter $\delta \in (0, 1)$ indicating the maximum allowed probability of error and their running time is polynomial in $\ln\left(\frac{1}{\delta}\right)$.

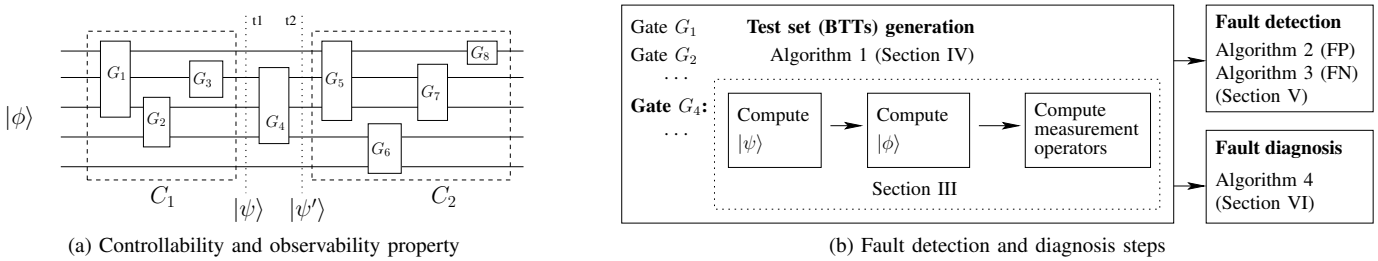
We will denote by C the circuit to be diagnosed which acts on n qubits and represent its gates by $G_1, G_2 \ldots G_s$ when enumerated in the standard manner from left to right (see Figure 2a). To simplify our notations, we will use the phrase "G_0 is faulty" to mean that "C is fault-free" [2]. In the single-gate fault model that we consider, at most one of these s gates is faulty, and moreover, the "fault is known", i.e., the exact operator corresponding to the faulty gate is also available to us. The operators for the fault-free and faulty i-th gate are denoted by G^i and G^i_f, respectively (G^i_f is set to the identity operator under the SMGF model). Let C^0 denote a circuit in which no gate is faulty, and C^i denote a circuit in which (only) the i-th gate is faulty. That is, $C^0 = G^s \ldots G^{i+1} G^i G^{i-1} \ldots G^1$ and $C^i = G^s \ldots G^{i+1} G^i_f G^{i-1} \ldots$ when the circuits and gates are represented as operators.

Problem statement: Given a parameter $\delta \in (0, 1)$ denoting the maximum allowed probability of error, we want to solve the following two problems with respect to the SMGF model.

[Detection problem] *Given a circuit C, we want to detect if C is fault-free or if any one of its gates is faulty.* According to the above notation, we want to determine if C can be represented as C^0 (in which case C would be fault-free) or as C^j for some non-zero j (in which case C would be faulty). Note that, in the latter case, we only need to establish that $j > 0$, but the explicit value of j need not be determined.

[Diagnosis problem] *Given a circuit C which is known to be faulty, find which gate is faulty.* Technically, given a C, find non-zero j such that $C = C^j$.

[2] G_0 is not an actual gate in C, which is composed of $G_1, \ldots G_s$; the phrase is used for notational uniformity in expressions.



(a) Controllability and observability property

(b) Fault detection and diagnosis steps

Fig. 2. A quantum circuit (left) and the pipeline for testing it (right)

The first novel contribution of this work is a characterisation of a "hard to detect" faulty gate based on the probability of error in detecting it.

Definition 1. *For the i-th gate of a circuit C, $\Delta(i) \in [0, 0.5]$ is defined as the minimum probability of error in detecting whether that gate is faulty or not, given that other gates are not faulty, using a single measurement of the circuit output. The minimum is taken over any measurement operator and any input state.*

We show how to relate $\Delta(i)$ to the eigenvalues of the operator corresponding to the i-th gate and its faulty version. So, essentially, $\Delta(i)$ is a property of the gate and its fault model and is *completely independent of* C. While $\Delta = 0.5$ indicates an "untestable fault" (that is impossible to detect), Δ close to 0 implies that the fault is easy to detect. To allow our algorithms to run within manageable limits, they take another user-defined hardness parameter $\lambda \in (0, 0.5]$ and does not look for faults for which $\Delta \geq \lambda$.

We define a fault to be "trivial" if it changes only the *global phase* of the gate operator, say, from G to $e^{i\theta}G$, for some non-zero θ; for example, $X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ to $iX = \begin{pmatrix} i & 0 \\ 0 & i \end{pmatrix}$. On *any* input state $|\psi\rangle$, if $|\psi'\rangle$ is the output state of the fault-free circuit, then the output state of the faulty-circuit would be $e^{i\theta}|\psi'\rangle$. Global phases have no observable effect upon measurement, and so both the output states would appear identical to any measurement operator. Therefore, a circuit with either G or $e^{i\theta}G$ are identical for all practical purpose, and furthermore, it is not possible to test such circuits solely using tomograms. This was also observed earlier by Perkowski et al. [13] who classified quantum faults on the basis of their detectability; they claimed that trivial faults belong to those faults which are untestable. Our next contribution is a tight version of their claim in which we show that "trivial faults" are exactly the *only untestable faults*. Therefore, unlike classical logic circuits but like classical reversible circuits, it is easy to determine if a single-gate fault is testable or untestable.

Lemma 2. *A trivial single-gate fault is untestable. Vice versa, if a single-fault is untestable then it must be trivial. Therefore, trivial faults are untestable and all other faults are testable.*

For the faults that are testable, we adopt the basic approach for classical reversible circuits by leveraging the fact that quantum circuits have both the *controllability* and *observability* properties. The outline of the process is illustrated in Figure 2b. Take for instance, the quantum circuit illustrated in Figure 2a and denote it by C. Suppose we want to know if the gate G_4 in that circuit is missing or not (our results hold for any single gate fault model – for this example, we use the SMGF model). For some input state $|\phi\rangle$, let $|\psi\rangle$ be the state of qubits at stage $t1$ and $|\psi'\rangle$ be the state at stage $t2$. If G_4 is missing, then $|\psi'\rangle = |\psi\rangle$ and otherwise, we can write $|\psi'\rangle = G_4|\psi\rangle$. For fault-activation, we want $|\phi\rangle$ to be such that $G_4|\psi\rangle$ is *very different* from $|\psi\rangle$. For fault-propagation, we want to ensure that this difference is also observable at the output after measurement. Due to the observability property of C, any difference between $|\psi\rangle$ and $G_4|\psi\rangle$ is carried forward to the corresponding output states. And by the controllability property, there is a unique $|\phi\rangle$ which will generate $|\psi\rangle$ at stage $t1$. Therefore, we need to carefully choose $|\psi\rangle$ and then determine $|\phi\rangle$ accordingly.

Paler et al. defined a *binary tomographic test* (BTT) as an input state for a quantum circuit analogous to a test vector for classical circuits [15]. However, the observed output of a circuit is determined not only by input state but also by the operators used for measurement. Therefore, we extend BTTs to also include the most appropriate measurement operators corresponding to every input state; thus BTT for us will mean *both an input state and a corresponding measurement operator*. Our second contribution is a quadratic programming based approach to obtain optimal BTT for any gate.

Theorem 3. *There is a BTT with probability of error $\Delta(i)$ which can detect if the i-th gate of a circuit is non-trivially faulty or fault-free, given that all other gates are not faulty. If efficient quantum circuit simulators are used, the BTT can be generated in time that is exponential in the number of qubits of the i-th gate.*

Even though our approach can be theoretically exponential in the number of qubits that the circuit acts on, since practical circuits are constructed using small gates (acting on 1 to 3 qubits), it is possible to compute BTT very efficiently even for large circuits with many gates. Several simulators are available that can simulate such circuits, even with many inputs [10].

It appears straightforward to design an ATPG method by constructing a test set that is composed of BTTs and then observing the circuit output on this test set. However, since the measurement output of a quantum circuit is a sample from a probability distribution, unlike classical circuits, multiple observations may be required to produce a tomogram that is close to the underlying output distribution. Furthermore,

even this tomogram can change from one round of testing to another; so, the naive approach of applying each BTT in the test set in some sequence may not be effective. Our next contribution is a randomised fault-detection algorithm that uses the Kullback-Leibler divergence and other tools from hypothesis testing to prescribe efficient employment of BTTs. The proposed testing method therefore involves two stages, first, creating the tomograms from sufficient number of outputs and secondly, processing the tomograms. Since individual BTTs ensure optimal error for every faulty gate, we would like to claim that our detection algorithm is the most efficient among all detection algorithms using tomograms. Clearly, if the circuit has no trivial faults, then fault-coverage is 100%.

Theorem 4. *For any $\delta \in (0,1)$ and $\lambda < 0.5$, all non-trivial single gate faults (whose $\Delta < \lambda$) in any quantum circuit can be detected with error at most δ. The test set consists of one BTT for each gate and the number of measurement outputs the testing algorithm requires is $O(\max\{s, \ln\frac{1}{\delta}, \frac{1}{\frac{1}{2}-\lambda}\})$.*

Our final contribution is a fault diagnosis algorithm that builds upon the detection algorithm. When acting on a faulty circuit, the algorithm returns a possible set of suspected faults, the fewer the better. The algorithm uses a test set consisting of s BTTs and the number of outputs it requires is polynomial in s.We show empirically that it performs quite well in practice by almost uniquely identifying the actual fault site.

We claim that it will be expensive to run a quantum circuit multiple times to measure its output, and so, according to us, the crucial component in testing efficiency is the number of times a circuit is run and its measurement outcome recorded. Accordingly, we define "M-cost" (for measurement cost) as the number of required measurement outcomes and treat it as the measure for testing complexity. We compare our approach with the state-of-the-art technique by Paler et al. [15] and show significant improvements in testing complexity.

The different components of our solutions are illustrated in Fig. 2b which we describe in the next four sections.

III. Detecting a specific faulty gate

This section describes our main tools. In the scenario that *all but the i-th gate are fault-free*, we want to detect if this gate is faulty or not.

For the sake of brevity, we will use the following notation in this section: $\Delta = \Delta(i)$, $G = G^i$, $G_f = G^i_f$, $C = C^0$ and $C' = C^i$. Thus, we want to know if the ith-gate is G or G_f. Broadly, our approach will be to:

1) find an input state $|\phi\rangle$ such that $|\psi\rangle = C|\phi\rangle$ and $|\psi'\rangle = C'|\phi\rangle$ are at the farthest "distance" possible.
2) find measurement operators which can distinguish between those two faraway states $|\psi\rangle$ and $|\psi'\rangle$ with minimal probability of failure.

Techniques for answering such questions for general quantum states operators are well-known [20], [21]. We show how to apply those techniques to our specific problem.

A. Optimal input state

The appropriate measure of "distance" for pure quantum states with respect to distinguishability is the trace distance defined by $D(|\psi\rangle, |\psi'\rangle) = \sqrt{1 - |\langle\psi|\psi'\rangle|^2}$. Trace distance is also equal to the maximum L1 distance of the probability distributions obtained from the two states upon any measurement [19]; given one of these states chosen equally at random, any algorithm is bound to incur at least $(1-D)/2$ probability of error while identifying the state.

Definition 5. *Given two operators G and G_f, we say that a state $|\phi\rangle$ is a (G, G_f)-separator (similarly, (C, C')-separator) if this state, given as input, maximises the trace distance between $G|\phi\rangle$ and $G_f|\phi\rangle$ (respectively, $C|\phi\rangle$ and $C'|\phi\rangle$).*

This definition immediately leads to the following lemma.

Lemma 6. *$\Delta = \frac{1}{2}\big(1 - D(C|\phi\rangle, C'|\phi\rangle)\big)$ where $|\phi\rangle$ is a (C, C')-separator state.*

Therefore, our first goal is to find a (C, C')-separator input state $|\phi\rangle$ which minimises $|\langle\psi|\psi'\rangle|$. Our main observation here is that we can decompose our circuits into common sub-circuits excluding the i-th gate: $C = C_2 G C_1$ and $C' = C_2 G_f C_1$. Let $S = G^\dagger G_f$. Without loss of generality, we can consider that G (hence, G' and S) acts on all n qubits (maybe by tensoring with an identity operator of suitable dimensions).

Let the the eigenvalues of S be denoted by $e^{-i\theta_1} \dots e^{-i\theta_m}$ (including duplicates) and the corresponding eigenvectors by $|v_1\rangle, \dots |v_m\rangle$. Consider this quadratic optimisation problem.

$$\mathbf{OPT}(\mathbf{S}): \quad \min \sum_j a_j^2 + \sum_{j\neq k} a_j a_k \cos(\theta_j - \theta_k) \quad (1)$$
$$\text{where} \sum_j a_j = 1, \quad 0 \le a_j \le 1$$

Let $\bar{a} = \{a_1 \dots a_m\}$ be a solution to this optimisation problem: Observe that minimising Eqn. 1 is equivalent to minimising $\sqrt{\sum_j a_j^2 + \sum_{j\neq k} a_j a_k \cos(\theta_j - \theta_k)}$

$$= \Big|\sum_j a_j \cos\theta_j - i\sum_j a_j \sin\theta_j\Big| = \Big|\sum_j a_j e^{-i\theta_j}\Big|$$
$$= |\langle\phi'|S|\phi'\rangle| = |\langle\phi'|G^\dagger G_f|\phi'\rangle|$$

where, $|\phi'\rangle = \sum_j \sqrt{a_j}|v_j\rangle$ is a state on n qubits. Therefore the optimum $\bar{a}$ for $\mathbf{OPT}(\mathbf{S})$ also minimises $|\langle\phi'|G^\dagger G_f|\phi'\rangle|$, which makes $|\phi'\rangle$ a (G, G_f)-separator. We can now choose $|\phi\rangle = C_1^\dagger|\phi'\rangle$ as our required (C, C')-separator input. Since $|\langle\phi'|S|\phi'\rangle| = |\langle\phi|C_1^\dagger G^\dagger C_2^\dagger C_2 G_f C_1|\phi\rangle| = |\langle\phi|C^\dagger C'|\phi\rangle| = |\langle\psi|\psi'\rangle|$, the optimum $\bar{a}$ also minimises $|\langle\psi|\psi'\rangle|$ and this minimum value is simply $|\sum_j a_j e^{-i\theta_j}|$. This gives us our main technical lemma.

Lemma 7. *Let a_j, θ_j and v_j be as defined in the description of $\mathbf{OPT}((\mathbf{G^i})^\dagger \mathbf{G^i_f})$ above. Then the state $\sum_j \sqrt{a_j}|v_j\rangle$ acts as (G^i, G^i_f)-separator and $\Delta(i) = \frac{1}{2}\left(1 - \sqrt{1 - |\sum_j a_j e^{-i\theta_j}|^2}\right) \in [0, \frac{1}{2}]$.*

If the fault in question belongs to the single missing gate fault model, then we can treat it is a special case of the above where $G_f = I$ and therefore $S = G^\dagger$.

If G acts on n' qubits and $n' \ll n$ (say, $n' = 1$ or 2), then it is possible to solve $\mathbf{OPT}(\mathbf{S})$ using the larger n-qubit operator $I_{n-n'} \otimes G_i$. This may be computationally expensive, so a better alternative is to let $S = G^\dagger G_f$ as before, and let $T = I_{n-n'} \otimes S$ be the extension of S to n qubits. If $\{(e^{-i\theta_j}, |v_j\rangle)\}$ are the eigen-pairs of S then it is easy to see that $\{(e^{-i\theta_j}, |v_j\rangle \otimes |0\rangle^{\otimes n-n'})\}$ are the eigen-pairs of T. Thus our required input state can be derived as $|\phi\rangle = C_1^\dagger\big(|\phi'\rangle \otimes |0\rangle^{\otimes(n-n')}\big)$ where $|\phi'\rangle$ is a (G, G_f)-separator input state. For example, if G is a single qubit gate, then we only need to store that $|\phi'\rangle$ is $\frac{1}{\sqrt{2}}(|v_1\rangle + |v_2\rangle)$ where $|v_1\rangle$ and $|v_2\rangle$ are the eigenvectors of $G^\dagger G_f$, irrespective of the value of n.

It should be obvious that our method of decomposing a circuit into portions before and after the gate in question can also be used for multiple missing/defective gate faults as long as the faulty gates can be grouped together and the circuit can be sliced around them. For example, our method is applicable to multiple gate faults if they act on distinct set of qubits, and/or are adjacent to each other; trivial extension is required to the computation of optimal state described earlier.

We end this subsection with a proof of Lemma 2 which characterises faults with $\Delta = 0.5$ as the trivial faults.

Proof of Lemma 2. Consider a trivial fault, i.e., $G_f = e^{i\alpha}G$ for some G acting on w qubits. Then, $S = e^{i\alpha}I$ and this has one eigenvalue $e^{i\alpha}$ with multiplicity $m = 2^w$. The solution of the optimisation problem can be readily seen to be $a_i = \frac{1}{m}$. This gives us $\Delta = 0.5$ which implies an untestable fault.

For the other direction, consider some gate G and its untestable faulty version G_f. Therefore, $\Delta = 0.5$ which implies that $|\sum_j a_j e^{-i\theta_j}|^2 = 1$. Suppose that at least two of θ_j's are distinct. It is easy to verify that any convex combination of such a set of points on the unit circle in the complex plane has modulus less than 1. Therefore, all the θ_j must be equal. Since S is diagonalisable (it is unitary), $G^\dagger G_f = S = V \cdot e^{-i\theta_j} I \cdot V^{-1}$ for some V. Therefore, $G^\dagger G_f$ is same as $e^{-i\theta_j} I$ which implies that $G_f = e^{-i\theta_j} G$, i.e., the fault simply changes the global phase of G. □

B. Optimal measurement operators

Once we have obtained the optimal input state $|\phi\rangle$, we can compute the two possible output states $|\psi\rangle = C|\phi\rangle$ and $|\psi'\rangle = C'|\phi\rangle$. Quantum states are manifested only by their measurement outputs. It is thus important to design and implement measurement operators that are able to distinguish between $|\psi\rangle$ and $|\psi'\rangle$, and thereby determine if the circuit in question is C or C'. However, unlike input states, measurement operators depend on the actual circuit and has to be computed once for every circuit and every fault model.

The question of distinguishing between two given quantum states is one of the classical problems of quantum computing [22]. Two states can be differentiated (using measurements) with certainty if and only if they are orthogonal. So, if G_f is almost same as G, then obviously no measurement should be able to distinguish between them with high confidence.

There are two known modes of distinguishing between a pair of states. Helstrom measurement is a two-output (von Neumann) projective measurement which *minimises* the error of incorrect labelling [20]. If we prohibit incorrect outcome and instead allow our measurement operators to either label a state with certainty or report "§"*(inconclusive)*, then we would be performing *unambiguous state discrimination* (USD) [23]–[25]. We will use Helstrom projective measurement in the rest of this paper for explaining our technique; however, we could have also used USD for doing the same.

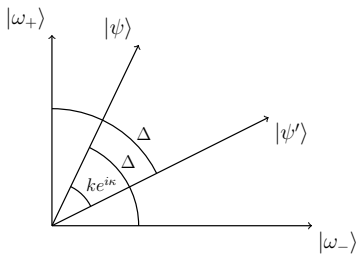


Fig. 3. Schematic diagram for the Helstrom projective measurement basis. The angles represent the inner product between the corresponding state vectors.

The concept behind Helstrom projective measurement is explained in Figure 3. Our derivation follows similar approach as in [20], [21] but is in a form that suits our problem.

First we create an orthonormal basis $|\omega_+\rangle$ and $|\omega_-\rangle$ which spans $|\psi\rangle$ and $|\psi'\rangle$. This basis will be used for measurement and we will infer the state as $|\psi\rangle$ or $|\psi'\rangle$ upon measurement outcome $|\omega_+\rangle$ or $|\omega_-\rangle$, respectively. We want to minimise the probability of error (when state is $|\psi\rangle$ but outcome is incorrectly $|\omega_-\rangle$ and similarly for the other pair); so the basis states should be maximally away from the output states, i.e., $|\langle\omega_-|\psi\rangle|^2 = |\langle\omega_+|\psi'\rangle|^2$.

We will represent by $ke^{i\kappa}$ the complex number $\langle\psi|\psi'\rangle = \sum_j a_j e^{-i\theta_j}$ in which a_j's are the solution of $\mathbf{OPT}(\mathbf{S})$ and $e^{i\theta_j}$ are the eigenvalues of $S = G^\dagger G_f$.

We first represent our states in terms of our basis states, i.e., $|\psi\rangle = \alpha_1|\omega_+\rangle + \beta_1|\omega_-\rangle$ and $|\psi'\rangle = \alpha_2|\omega_+\rangle + \beta_2|\omega_-\rangle$. Without loss of generality, we can take α_1 as a real number r_1. The condition of equal probability of error enforces these representations: $\beta_1 = r_2 e^{ix_1}$ for some real $r_2 = \sqrt{1-r_1^2}$, $\alpha_2 = r_2 e^{ix_2}$ and $\beta_2 = r_1 e^{ix_3}$. Equating $ke^{i\kappa} = \langle\psi|\psi'\rangle = r_1 r_2 e^{ix_2}(1 + e^{i(x_3 - x_1 - x_2)})$, we obtain one possible solution: $x_1 = 0$, $x_2 = \kappa$, $x_3 = \kappa$ and $r_{1,2} = \left(\sqrt{1+k} \pm \sqrt{1-k}\right)/2$ which produces this basis:

$$|\omega_+\rangle = \frac{-r_1}{r_2^2 - r_1^2}|\psi\rangle + \frac{r_2 e^{-i\kappa}}{r_2^2 - r_1^2}|\psi'\rangle$$

$$|\omega_-\rangle = \frac{r_2}{r_2^2 - r_1^2}|\psi\rangle - \frac{r_1 e^{-i\kappa}}{r_2^2 - r_1^2}|\psi'\rangle$$

Therefore, we obtain the three following projectors to distinguish between $|\psi\rangle$ and $|\psi'\rangle$: $\{P_{\mathtt{0}} = |\omega_+\rangle\langle\omega_+|, P_{\mathtt{1}} = |\omega_-\rangle\langle\omega_-|, P_{\S} = \mathbb{I} - P_{\mathtt{0}} - P_{\mathtt{1}}\}$ with outcomes $\mathtt{0}$, $\mathtt{1}$ and §, respectively. The outcome $\mathtt{0}$ corresponds to the output state being $|\psi\rangle$ and hence implies that the circuit is (probably) fault-free; similarly, outcome $\mathtt{1}$ implies that the i-th gate is probably faulty. Outcome § is never observed if circuit is fault-free or if the i-th gate is faulty; therefore, outcome § immediately signifies that the circuit has fault at some other gate.

C. Optimal BTT

Without loss of generality, we will henceforth assume that all our faults are non-trivial, i.e., $\Delta < 0.5$.

Definition 8. Binary tomographic test *for the i-th missing gate, denoted by $BTT(i)$, is defined as the combination of a (C^0, C^i)-separator input and corresponding measurement operator $\{P_{\mathtt{o}}, P_{\mathtt{1}}, P_{\S}\}$.*

We capture the result of application of any BTT on any circuit C by the probability distribution on the measurement outcomes.

Definition 9. *$\mu_{i,j}$ is defined as the probability distribution $\{p(\mathtt{o}), p(\mathtt{1}), p(\S)\}$ of measurement outcomes when $BTT(i)$ is employed on circuit C^j.*

The probability of error after one measurement would be at most $|\langle\psi|\omega_-\rangle|^2 = r_2^2 = \left(1-\sqrt{1-k^2}\right)/2 = \left(1-\sqrt{1-|\sum_j a_j e^{-i\theta_j}|^2}\right)/2$ which matches the minimum probability of error in distinguishing $|\psi\rangle$ and $|\psi'\rangle$ by any projective measurement. This establishes the following lemma on the optimality of a BTTs error.

Lemma 10. *The output distribution of $BTT(i)$ on the fault-free circuit C^0 is $\mu_{i0} = \{1-\Delta(i), \Delta(i), 0\}$ and the distribution on the faulty circuit C^i is $\mu_{ii} = \{\Delta(i), 1-\Delta(i), 0\}$.*

Discussion of Theorem 3. Lemma 10 shows that $BTT(i)$ can *optimally distinguish* between a fault-free gate and a faulty gate, as long as it is a non-trivial fault.

The computationally significant steps are that of determining the (G, G_f)-separator and calculating the input state to the circuit and the optimal measurement operator.

The latter involves (a) determining the input state $|\phi\rangle$ to the circuit from the (G, G_f)-separator $|\phi'\rangle$ (b) determining the output states $|\psi\rangle$ and $|\psi'\rangle$ of the fault-free and faulty circuits from the input state, and (c) determining the measurement operator $\{P_{\mathtt{o}}, P_{\mathtt{1}}, P_{\S}\}$ on the output states. Many efficient programming platforms and libraries exist today to numerically calculate these states and operators.

The former step involves solving the quadratic program (1) on 2^w variables, where w is the number of qubits of the i-th gate. Quadratic programming is in general an NP-hard problem, but we believe its use in deriving the BTT is not a computational hurdle for a couple of reasons. First, the number of variables is exponential only in the dimension of the gate involved, which is usually quite small in literature we have encountered so far; also, it is also likely that synthesis of gates acting of many qubits is going to be difficult. Secondly, observe the interesting fact that the separator input for a gate in a circuit depends fundamentally on the gate in question and corresponding fault model. It *does not* depend at all on the portion of the circuit coming after the faulty gate, and its dependence on the portion of the circuit before the faulty gate is really incidental. Therefore, it is feasible to have a pre-computed table of (G, G_f)-separators for different gates under common fault models. The required separator input for any circuit can be obtained by running the first portion of the circuit in reverse on a gate-separator input. Thus, the major computation tasks of eigen-decomposition of S and solving $\mathbf{OPT(S)}$ can be done only once and reused as needed. □

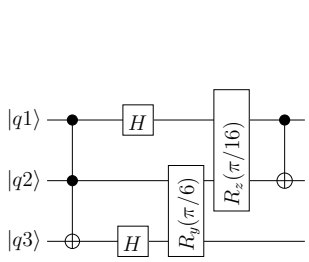


Fig. 4. Benchmark circuit 3qubit-CNOT on 3 qubits with 6 gates. The 2-qubit rotation gates apply the rotation operations to both the qubits.

Table I shows the probability of error in detecting SMGF faults for some of the commonly used quantum gates. The table demonstrates that (a) for most gates (for example, Hadamard), missing gate faults can be easily detected with certainty, and (b) there are some gates (for example, $R_z(\pi/2^{12})$) for which SMGF will be quite hard to detect.

IV. Test generation

Algorithm 1 Test generation stage

Input: $C^0 = \langle G^1 \dots G^s\rangle$: Fault-free gates
$\{G_f^1, \dots G_f^s\}$: Faulty gates
Output: T: BTT table
Δ: Fault error table

1: Initialise empty $s \times (1+s)$ table T and $s \times 1$ table Δ
2: **for** $i = 1$ to s **do**
3: Break C^0 into C_1 and C_2 around i-th gate
4: Compute C^i from C^0 and G_f^i
5: $\{(e^{-i\theta_1}, |v_1\rangle), \dots (e^{-i\theta_m}, |v_m\rangle)\} \leftarrow$ eigen-decomposition of $(G^i)^\dagger G_f^i$
6: $\vec{a} \leftarrow$ solution of $\mathbf{OPT}((\mathbf{G^i})^\dagger \mathbf{G_f^i})$
7: Compute $\Delta[i] \leftarrow |\sum_j a_j e^{-i\theta_j}|$
8: Compute $|\phi'\rangle = \sum_j \sqrt{a_j}\,|v_j\rangle$ and $|\phi\rangle = C_1^\dagger\,|\phi'\rangle$
9: Compute $|\psi\rangle = C^0|\phi\rangle$, $|\psi'\rangle = C^i\,|\phi\rangle$
10: Compute $|\omega_+\rangle$ and $|\omega_-\rangle$
11: Compute $P_{\mathtt{o}}, P_{\mathtt{1}}, P_{\S}$
12: **for** $j = 0$ to s **do**
13: Compute $|\alpha\rangle = C^j\,|\phi\rangle$
14: Compute $p_k = \langle\alpha\,|P_k|\alpha\rangle$, for $k = \mathtt{o}, \mathtt{1}, \S$
15: $T[i,j] \leftarrow (p_{\mathtt{o}}, p_{\mathtt{1}}, p_{\S})$
16: **end for**
17: **end for**
18: **return** T, Δ

In this section we use the BTTs defined in Section III to generate a test-set that, for us, consists of *BTT-table* and a *fault-error table*. The BTT-table is a generalisation of the *fault-table* constructed by Aligala et al. [16]; for each BTT, it contains the corresponding probability distribution on outcomes.

The test generation stage (illustrated in Algorithm 1) takes as input a description of the circuit, along with each of the fault-free and faulty operators. First, for each gate G_i, we construct the input state and measurement operator for $BTT(i)$. Then, we construct a *BTT table* with s rows and

TABLE I
MINIMUM PROBABILITY OF ERROR (Δ) FOR DETECTING (SINGLE) MISSING GATE. THE GATES IN THE CLIFFORD+T SET OF QUANTUM GATES ARE MARKED IN **bold** – THIS SET IS GAINING POPULARITY FOR CONSTRUCTING FAULT-TOLERANT QUANTUM CIRCUITS.

Gate	Error prob.	Gate	Error prob.	Gate	Error prob.	Gate	Error prob.	Gate	Error prob.	Gate	Error prob.
Hadamard	0.00	**Pauli-Z**	0.00	**CNOT**	0.00	**Phase** = $R_z(\frac{\pi}{2})$	0.15	$R_z(\pi/16)$	0.415	$R_z(\pi/2^8)$	0.495
Pauli-X	0.00	**Pauli-Y**	0.00	Toffoli	0.00	**T** = $R_z(\pi/4)$	0.31	$R_z(\pi/32)$	0.458	$R_z(\pi/2^{12})$	0.4997

$(1+s)$ columns whose (q,r)-th cell contains the distribution $\mu_{q,r}$ when $BTT(q)$ is applied to circuit C^r and a *fault-error array* whose r-th entry $\Delta(r)$ contains the minimum possible probability of error in distinguishing between C^0 and C^r.

BTT table and fault-error array for a benchmark circuit *3qubitcnot* (illustrated in Figure 4) are given in the Table II. It can be readily inferred by comparing the C^0 and C^5 columns that detecting missing G^5 gate is more error-prone and is going to require lots of samples.

Next we describe some important properties of the test-set that are used by upcoming detection and diagnosis algorithms.

A. Test-set properties

If $\Delta(r)$ is close to 0.5, then C^0 is almost similar to C^r and then there is a large chance of error in distinguishing between C^0 and C^r (it becomes impossible when $\Delta = 0.5$). So, we use a subroutine `CleanupBTTTable(`λ`)` which takes a user-parameter $\lambda \in (0, 0.5)$ and ignores all faults with error more than Δ. It does so by removing the $(1+r)$-th column (corresponding to C^r) and r-th row (corresponding to $BTT(r)$) from the BTT table.

Suppose, we apply a BTT on a circuit several times and record the distribution of outcomes. We use $\tau(i, C, m)$ to denote this distribution obtained by applying $BTT(i)$ on $C \in \{C^0, \ldots, C^s\}$ m number of times. After sufficient number of times of applying the test, the distribution is expected to converge towards $\mu_{i,j}$. But more importantly, for any i, m and C^j, $\tau(i,0,m) = (a_1, a_2, 0)$ and $\tau(i,i,m) = (a_3, a_4, 0)$ for some a_1, a_2, a_3, a_4. So, in essence, we need to choose a large enough m such that a sampled distribution τ can be confidently attributed to come from either $\mu_{i,0}$ or $\mu_{i,i}$. We will use the Kullback-Leibler divergence, denoting it by D_{KL}, for testing closeness of τ with the source distributions. The next lemma states the minimum number of samples needed.

Lemma 11. *Choose any $\delta \in (0,1)$ and some $i \in \{1, \ldots s\}$ such that $\Delta(i) \neq 0.5$. If m is selected as $\left\lceil \frac{\ln(1/\delta)}{-\ln(2\sqrt{\Delta(i)(1-\Delta(i))})} \right\rceil$, then the following events have a probability at most δ.*

(a) $D_{KL}(\tau\|\mu_{i,i}) < D_{KL}(\tau\|\mu_{i,0})$ where τ is the distribution of m samples drawn from $\mu_{i,0}$.

(b) $D_{KL}(\tau\|\mu_{i,i}) > D_{KL}(\tau\|\mu_{i,0})$ where τ is the distribution of m samples drawn from $\mu_{i,i}$.

Proof. We will give the proof for part (a). Part (b) can be proved similarly.

Let p denote $1 - \Delta(i)$. Note that for any i, $\mu_{i,0} = (p, 1-p, 0)$ and $\mu_{i,i} = (1-p, p, 0)$. Since τ is obtained from $\mu_{i,0}$, $\tau(\S) = 0$ and $\tau(\circ)$ can be written as x/m where x follows the distribution $Binomial(m, p)$.

$$
\begin{aligned}
&\Pr[D_{KL}(\tau\|\mu_{i,i}) < D_{KL}(\tau\|\mu_{i,0})] \\
&= \Pr\left[\sum_{z\in\{\circ,\mathbf{1},\S\}} \tau(z)\frac{\lg \mu_{i,0}(z)}{\lg \mu_{i,i}(z)} < 0\right] \\
&= \Pr\left[(m-2x)\lg\frac{p}{1-p} > 0\right] \\
&= \Pr[x < m/2] \text{ since, } p > 0.5 \\
&\leq \exp\Big(-mD_{KL}\big((0.5,0.5)\|(1-p,p)\big)\Big) \\
&= \exp\left(m[1 + \frac{1}{2}\ln p(1-p)]\right)
\end{aligned}
$$

The last inequality is an application of the Chernoff-Hoeffding bound. The required bound on m follows immediately by setting the final expression to δ. □

We use $\eta(i,\delta)$ (or η_i in short when δ is fixed) to denote the above bound m on the number of samples required for $BTT(i)$ for a particular δ. The subroutine `NumSamples()` computes η_i for all $i = 1 \ldots s$.

Let η_{max} denote the maximum sample size and η_{avg} denote $\sum_{i=1}^{s} \eta_i/s$. We now upper bound all these values.

Lemma 12. *Suppose for some i, $\Delta(i) \leq \lambda$ for some $\lambda \leq 1/2$. Then, $\{\eta_i, \eta_{max}, \eta_{avg}\} \leq \dfrac{\ln(1/\delta)}{2\left(\frac{1}{2}-\lambda\right)^2} + 1$.*

TABLE II
BTT-TABLE AND FAULT-ERROR ARRAY FOR CIRCUIT 3QUBITCNOT WITH SINGLE MISSING GATE FAULTS

Test	C^0	C^1	C^2	C^3	C^4	C^5	C^6
BTT(1)	(1.00,0.00,0.00)	(0.00,1.00,0.00)	(0.28,0.09,0.63)	(0.30,0.50,0.20)	(0.92,0.04,0.05)	(0.99,0.00,0.01)	(0.91,0.00,0.09)
BTT(2)	(1.00,0.00,0.00)	(0.02,0.13,0.85)	(0.00,1.00,0.00)	(0.50,0.00,0.50)	(0.87,0.00,0.13)	(0.99,0.00,0.01)	(0.76,0.00,0.24)
BTT(3)	(1.00,0.00,0.00)	(0.73,0.13,0.15)	(0.50,0.00,0.50)	(0.00,1.00,0.00)	(0.87,0.06,0.07)	(0.98,0.00,0.02)	(0.86,0.00,0.14)
BTT(4)	(0.75,0.25,0.00)	(0.75,0.25,0.00)	(0.38,0.13,0.50)	(0.00,0.00,1.00)	(0.25,0.75,0.00)	(0.74,0.24,0.02)	(0.19,0.07,0.74)
BTT(5)	(0.60,0.40,0.00)	(0.29,0.20,0.52)	(0.20,0.30,0.50)	(0.55,0.38,0.07)	(0.52,0.35,0.13)	(0.40,0.60,0.00)	(0.25,0.25,0.50)
BTT(6)	(1.00,0.00,0.00)	(0.33,0.29,0.38)	(0.06,0.22,0.72)	(0.10,0.42,0.48)	(0.87,0.02,0.11)	(0.99,0.01,0.00)	(0.00,1.00,0.00)
Δ	N.A.	0	0	0	0.25	0.4	0

Proof. It suffices to lower bound $-\ln(2\sqrt{\Delta(i)(1-\Delta(i))})$ by $2(\frac{1}{2}-\lambda)^2$. This is done as follows.

$$
\begin{aligned}
&-\ln(2\sqrt{\Delta(i)(1-\Delta(i))}) \\
&\geq -\ln(2\sqrt{\lambda(1-\lambda)}) \\
&\textit{(since, } \Delta(1-\Delta) \textit{ is increasing in } [0,0.5]\textit{)} \\
&= -\ln 2 - \frac{1}{2}\ln\lambda - \frac{1}{2}\ln(1-\lambda) \\
&= -\frac{1}{2}\left[\ln(1+2\epsilon)+\ln(1-2\epsilon)\right] \quad (\text{let } \epsilon = \tfrac{1}{2}-\lambda) \\
&\geq -\frac{1}{2}\left[-2\frac{(2\epsilon)^2}{2} - 2\frac{(2\epsilon)^4}{4} - \dots\right] \\
&\geq 2\epsilon^2 = 2(\frac{1}{2}-\lambda)^2
\end{aligned}
$$

□

V. Detection of single-fault circuits

In the last two sections we discussed how to efficiently obtain a test set. In this section we give an algorithm that uses the test set to decide if a circuit is faulty or fault-free, under the SMGF model: formally, given some C^j as the circuit C for testing, our goal is to detect whether $j=0$ or $j>0$. The output of quantum circuits being probabilistic, our approaches will be prone to detection error. We define two types of error: *false-positive* (FP) when a fault-free circuit is claimed to be faulty and *false-negative* (FN) when the converse happens and our algorithms will only allow limited scope of such errors.

Our algorithms for fault detection are presented in Algorithms 2 and 3. At the core of both of them is the subroutine `RunBTT`(C,i,T,η_i) which runs $BTT(i)$ on the given circuit C sufficient number of times specified by η_i. It then compares the output distribution thus obtained with the two distributions $\mu_{i,0}$ and $\mu_{i,i}$ and returns the distribution from which τ is more likely to have been derived. Recall that the number of samples η_j is sufficiently large to ensure, with high probability, that C is faulty if and only if there exists some $BTT(i)$ for which $\tau(i,C,\eta_i)$ is closer to $\mu_{i,i}$ than $\mu_{i,0}$. Furthermore, as an additional optimisation, if `RunBTT` finds some $\S$ among the outcomes of $BTT(i)$, then it declares that the circuit cannot be fault-free (as per Lemma 10).

Algorithm 2(FP) simply chooses a random BTT and uses it to decide if the circuit or not. It ensures that false-positive is at most δ, i.e., a fault-free circuit is seldom reported as as faulty. On the other hand, a faulty-circuit (say C^j) is reported correctly (with probability $1-\delta$) when $i=j$; therefore, false-negative probability is at most $1-(1-\delta)/s$. Clearly, its M-cost can be at most η_{max} with an expected value of η_{avg}.

The other one, Algorithm 3(FN) guarantees low false-negative, i.e., it ensures that a faulty circuit is reported as faulty with probability at least $1-\delta$. It does so by serially running all the BTT until it finds one that indicates that the circuit is faulty; it returns fault-free only if none of the tests indicate any fault with the circuit. If $C=C^j$, then $BTT(j)$ will surely detect C as faulty. On the other hand, a fault-free C is reported as faulty only if all tests report "fault-free"; thus, false positive probability is $1-(1-\delta)^s \approx s\delta$.

Input: T: BTT table
Δ: Fault error table
s: Number of gates
C: Circuit to be checked for fault
δ: Max. probability of error
λ: Threshold of fault error
Output: `Faulty` or `Fault-free`

Algorithm 2 Fault Detection (FP)

```
1: CleanupBTTTable(λ)
2: [η_1, η_2, ... η_s] ← NumSamples(δ)
3: i ←_R {1, ..., s}
4: return RunBTT(C, i, T, η_i)
```

Algorithm 3 Fault Detection (FN)

```
1: CleanupBTTTable(λ)
2: [η_1, η_2, ... η_s] ← NumSamples(δ)
3: Sort BTTs in increasing Δ
4: for BTT i = 1 to s (in sorted order) do
5:     result ← RunBTT(C, i, T, η_i)
6:     if result == Faulty then
7:         return Faulty
8:     end if
9: end for
10: return Fault-free
```

```
function RunBTT(C, i, T, η_i)
    Run BTT(i) on C η_i times
    τ ← distribution of outcomes
    if τ(§) ≠ 0 then
        return Faulty
    end if
    d1 ← D_KL(τ‖μ_{i,0})
    d2 ← D_KL(τ‖μ_{i,i})
    if d1 > d2 then
        return Faulty
    else
        return Fault-free
    end if
end function
```

We do an additional optimisation in this algorithm. We first sort the BTTs before using them sequentially (Line 3). This is because among all orderings of BTTs, the one that leads to the smallest average number of samples is the one in which the tests are sorted according to the increasing number of samples η_i (this is exactly the property of minimum average waiting time in the "Shortest Job First" scheduling algorithm). This ensures that Algorithm 3 uses the least number of samples (averaged across faults). The M-cost in this case is always $s\eta_{avg}$.

Proof of Theorem 4. Both Algorithms 2(FP) and 3(FN) use a test set consisting of s BTTs and can detect all non-trivial faults with $\Delta < \lambda$. Algorithm 2(FP) ensures that the probability of false-positive is at most δ and Algorithm 3(FN) ensures that the false-negative probability is at most δ. The number of circuit outputs required by the former algorithm is at most $1 + \ln(1/\delta)/2(0.5 - \lambda)^2$ and the same by the latter algorithm is at most $s(1 + \ln(1/\delta)/2(0.5 - \lambda)^2)$. □

VI. Diagnosis of Single-Fault Circuits

In this section we discuss our solution for fault diagnosis — not only we are interested in finding out if a given quantum circuit has a (single gate) fault, but we want to identify the particular fault as well. As in Section V, our diagnostic strategy uses the BTT-table and the fault-error table generated by the test generation stage (Algorithm 1).

Our algorithm is described in Algorithm 4. It first generates some candidate faults (denoted by $CAND$), and then verifies if each candidate can be the possible fault by comparing the obtained distribution with the possible distribution. Described in lines 8–17 and similar to the algorithm for detection, $CAND$ is obtained by first removing all i for which $BTT(i)$ has non-zero § outcomes. The faults left behind are further filtered using KL-Divergence. For any i for which $\tau_i[\S] = 0$, we accept i as a candidate if the obtained distribution τ is closer to $\mu_{i,i}$ than $\mu_{i,0}$. We wanted to ensure two properties of $CAND$ — (i) if $C = C^j$, then j should be in $CAND$ (no false negative), and (ii) if $C = C^j$, the $CAND$ should not contain any $i \neq j$ (no false positives).

However, the above steps are insufficient in removing false positives, so we propose two additional heuristics to reduce false positives and also ensure no false negative. In those two heuristics (lines 18–24 and lines 25–35, respectively), the set of candidates are further pruned based on additional properties of the entire list of distributions $[\tau_1, \tau_2, \ldots \tau_s]$ using two additional tables $Map1$ and $Map2$, created by subroutines `CreateMap1` and `CreateMap2`, respectively. For every faulty circuit C^j, $i \in Map1[j]$ if $BTT(i)$, when applied on C^j, would never output §. On the other hand, $Map2[j]$ contains those BTTs whose majority outcome would be ɔ.

Similar to the earlier algorithm for fault detection, this algorithm also choose the number of samples carefully to ensure bounded error during diagnosis. The subroutine `NumSampleDiag()` essentially sets η_i to be $\max(\eta_i^a, \eta_i^b, \eta_i^c)$ (see Algorithm 4) where these have the following properties.

- $\eta_i^a = \eta(i, \delta)$ is defined in Lemma 11,
- η_i^b is a large enough integer such that if for any j, $\mu_{i,j}[\S] > 0$, then η_i^b samples from $\mu_{i,j}$ will contain non-zero § with probability at least $1 - \delta$ (see Lemma 13),
- η_i^c is a large enough integer such that if any $\mu_{i,j}[\mathfrak{o}] > 0.5$, then η_i^c samples from $\mu_{i,j}$ will have ɔ as the majority sample with probability at least $1 - \delta$ (see Lemma 14).

Lemma 13. *Suppose for some i and j, $\mu_{i,j}(\S) \neq 0$. Let τ_i denote a distribution obtained from $\eta_{i,j}^b$ independently chosen samples from $\mu_{i,j}$. Then $\Pr_{\tau_i}[\tau_i(\S) = 0] \leq \delta$. Here, $\eta_{i,j}^b = \frac{\lg(\delta)}{\lg(1-\mu_{i,j}(\S))}$ if $\mu_{i,j}(\S) \neq 1$ and 1 otherwise.*

The proof of the above lemma is straight-forward and omitted. We set $\eta_i^b = \max_j \eta_{i,j}^b$.

Lemma 14. *Suppose for some i and j, $\mu_{i,j}(\mathfrak{o}) > 1/2$. Let τ_i denote a distribution obtained from $\eta_{i,j}^c$ independently chosen samples from $\mu_{i,j}$ where $\eta_{i,j}^c = 2\mu_{i,j}(\mathfrak{o})\frac{\ln(1/\delta)}{(\mu_{i,j}(\mathfrak{o})-1/2)^2}$. Then $\Pr_{\tau_i}[\tau_i(\mathfrak{o}) \leq 1/2] \leq \delta$.*

This lemma can be proved using Chernoff bound [3]. We set $\eta_i^c = \max_j \eta_{i,j}^c$.

Since η_i is set to $\max\{\eta_i^a, \eta_i^b, \eta_i^c\}$, we can say that the sample size for $BTT(i)$ ensures that Lemma 11, 13 and 14 are satisfied. We denote $\eta_{max} = \max_i \eta_i$ and $\eta_{avg} = \sum_i \eta_i / s$.

It is clear that Algorithm 4 always uses $\sum_i \eta_i$ samples. We prove its correctness next.

Theorem 15. *Suppose $C = C^j$ for some $j > 0$.*

(a) Then, $SUSP$ returned by Algorithm 4 contains j with probability at least $1 - 2\delta$.

(b) Consider any $0 < i \neq j$. Then, $SUSP$ returned by Algorithm 4 does not contain i with probability at least $1 - \delta$ unless $\mu_{i,j} = (p, 1-p, 0)$ for some $p \leq 1/2$.

Furthermore, M-cost of Algorithm 4 is $s \cdot \eta_{avg}$.

Proof of part(a). We will essentially prove three claims; $j \in CAND$, $j \notin REJ1$ and $j \notin REJ2$, each holding with high probability.

Since $\mu_{j,j}(\S) = 0$, therefore, $\tau_j(\S) = 0$ when $BTT(j)$ is applied. Lemma 11 gives us that $\Pr[j \text{ added to } CAND] \geq 1 - \delta$ (lines 11–15).

By the property of $Map1$, for any $i \in Map1[j]$, $\tau_i(\S) = 0$ since τ_i is obtained by running $BTT(i)$ on C^j. Therefore, j will *not* be added to $REJ1$ (in line 21).

Finally, we show $j \notin REJ2$ *w.h.p.* by using a probabilistic reasoning. For all $k \in \{1, \ldots, s\}$, let X_k denote the following indicator random variable.

$$X_k = \left\{ \begin{array}{ll} 1, & \text{if } k \in Map2[j] \ \&\ \tau_k(\mathfrak{o}) < 1/2 \\ 0, & \text{otherwise} \end{array} \right\}$$

First, note that $\sum_k X_k$ is precisely the value of *count* before line 32. Secondly, $\Pr[X_k] < \delta$ since (a) if $\mu_{k,j}(\mathfrak{o}) \leq 1/2$, $k \notin Map2[j]$, and (b) if $\mu_{k,j}(\mathfrak{o}) > 1/2$, then $\Pr[\tau_k(\mathfrak{o}) \leq 1/2] < \delta$ (by Lemma 14). Therefore, $\mathbb{E}[count] = \sum_k \mathbb{E}[X_k] < s\delta$. We apply Chernoff bound [4] to upper bound $\Pr[j \text{ added to } REJ2] = \Pr[count \geq s\delta(1+t)] \leq \exp(-\frac{t^2}{2+t}s\delta)$.

Case (a) $s\delta \leq 1$**:** In this case, $t = \frac{2}{s\delta}\ln\frac{1}{\delta}$ and so, $2t > 2+t$. Therefore, $\exp(-\frac{t^2}{2+t}s\delta) \leq \exp(-\frac{t}{2}s\delta) \leq \delta$.

Case (b) $s\delta > 1$**:** Note that in this case, if $t > s$, then $(1+t)s\delta > s^2\delta > s$; however, *count* can never be more than s. Therefore, we can take $t \leq s$ and so, $2+t < 2s$. Therefore, $\exp(-\frac{t^2}{2+t}s\delta) \leq \exp(-\frac{t^2}{2s}s\delta) = \exp(-\frac{t^2\delta}{2})$ which is at most δ since $t = \sqrt{\frac{2}{\delta}\ln\frac{1}{\delta}}$ in this case.

[3] We use the following application of Chernoff bound. Let X denote the number of heads when n coins are tossed, each with probability of head equal to $p > 1/2$. Then, $\Pr[X > n/2] \geq 1 - \exp(-\frac{n}{2p}(p - \frac{1}{2})^2)$.

[4] We are using the following version: For any $\sigma > 0$, $\Pr[\sum X_k \geq (1+\sigma)E] \leq \exp(-\frac{\sigma^2}{2+\sigma}E)$ where E is $\mathbb{E}[X_k]$ or any upper bound on it [26].

Input: T: BTT table
Δ: Fault error table
s: Number of gates
C: Circuit to be checked for fault
δ: Max. probability of error
λ: Threshold of fault error
Output: `Faulty` or `Fault-free`

Algorithm 4 Fault Diagnosis

`CleanupBTTTable(`λ`)`
For $i = 1$ to s, $\eta_i \leftarrow$ `NumSampleDiag(`δ, i`)`
For $j = 1$ to s, $Map1[j] \leftarrow$ `CreateMap1(`δ, j`)`
For $j = 1$ to s, $Map2[j] \leftarrow$ `CreateMap2(`δ, j`)`
Initialise $CAND \leftarrow \{\}$, $REJ1 \leftarrow \{\}$, $REJ2 \leftarrow \{\}$
if $s\delta \leq 1, \quad t \leftarrow \sqrt{\frac{2}{\delta} \ln \frac{1}{\delta}}$ **else** $\quad t \leftarrow \frac{2}{s\delta} \ln \frac{1}{\delta}$
for BTT $i = 1$ to s **do**
    Run $BTT(i)$ on C η_i times
    $\tau_i \leftarrow$ distribution of outcomes
    if $\tau_i(\S) = 0$ **then**
        $d0 \leftarrow D_{KL}(\tau_i \| \mu_{i,0})$
        $d1 \leftarrow D_{KL}(\tau_i \| \mu_{i,i})$
        if $d0 > d1$ **then**
            Add i to $CAND$
        end if
    end if
end for
for j in $CAND$ **do**
    for i in $Map1[j]$ **do**
        if $\tau_i[\S] \neq 0$ **then**
            Add j to $REJ1$; break;
        end if
    end for
end for
for j in $CAND$ **do**
    $count \leftarrow 0$
    for k in $Map2[j]$ **do**
        if $\tau_k[\mathfrak{o}] < 1/2$ **then**
            $count \leftarrow count + 1$
        end if
    end for
    if $count \geq s\delta(1 + t)$ **then**
        Add j to $REJ2$
    end if
end for
$SUSP \leftarrow CAND \setminus REJ2 \setminus REJ1$
return $SUSP$

function NUMSAMPLEDIAG(δ, i)
    $d0 \leftarrow \mu_{i,0}$
    $d1 \leftarrow \mu_{i,i}$
    if $d0 = (1, 0, 0)$ or $(0, 1, 0)$ **then**
        $\eta_i^a \leftarrow 1$
    else
        $\eta_i^a \leftarrow \eta(i, \delta)$ (Lemma 11)
    end if
    $\eta_i^a \leftarrow 1$, $\eta_i^c \leftarrow 1$
    for BTT $j = 1$ to s *s.t.* **do**
        if $0 < \mu_{i,j}(\S) < 1$ **then**
            $\eta_i^b \leftarrow \max\left\{\eta_i^b, \frac{\log(\delta)}{\log(1-\mu_{i,j}(\S))}\right\}$
        end if
        if $\mu_{i,j}(\mathfrak{o}) > 1/2$ **then**
            $\eta_i^c \leftarrow \max\left\{\eta_i^c, \frac{2\mu_{i,j}(\mathfrak{o}) \ln(1/\delta)}{(\mu_{i,j}(\mathfrak{o})-1/2)^2}\right\}$
        end if
    end for
    return $\max\{\eta_i^a, \eta_i^b, \eta_i^c\}$
end function

function CREATEMAP1(δ, j)
    Initialise $Map1[j] = \{\}$
    for faulty circuit $i = 1 \ldots s$ **do**
        if $\mu_{i,j}(\S) = 0$ **then**
            Add i to $Map1[j]$
        end if
    end for
    return $Map1[j]$
end function

function CREATEMAP2(δ, j)
    Initialise $Map2[j] = \{\}$
    for faulty circuit $k = 1 \ldots s$ **do**
        if $\mu_{k,j}(\mathfrak{o}) > 1/2$ **then**
            Add k to $Map2[j]$
        end if
    end for
    return $Map2[j]$
end function

Therefore, j is added to $REJ2$ with probability at most δ. Combining all the scenarios, j is not in $SUSP$ either if j is not added to $CAND$ or j is also added to $REJ2$ – which happens with probability at most 2δ. □

We need the following results on the sample sizes to prove that false positives happen with low probability.

Lemma 16. *Let τ_i be a distribution obtained from η_i runs of $BTT(i)$ on a circuit C^j for some $i \neq j$. If $\mu_{i,j}[\S] = 0$, $\mu_{i,0}[\mathfrak{o}] < 1$ and $\mu_{i,j}[\mathfrak{o}] > 1/2$, then with probability at least $1-\delta$, $D_{KL}(\tau\|\mu_{i,0}) \leq D_{KL}(\tau\|\mu_{i,i})$ holds.*

Proof. Let $a = \tau[\mathfrak{o}]$ and $p = \mu_{i,0}[0]$ (note that $1/2 < p < 1$). Therefore, $\mu_{i,0} = (p, 1-p, 0)$ and $\mu_{i,i} = (1-p, p, 0)$.

$$
\begin{aligned}
&D_{KL}(\tau\|\mu_{i,0}) - D_{KL}(\tau\|\mu_{i,i}) \\
&= a \lg \frac{a}{p} + (1-a) \lg \frac{1-a}{1-p} - a \lg \frac{a}{1-p} - (1-a) \lg \frac{1-a}{p} \\
&= a \lg \frac{1-p}{p} + (1-a) \lg \frac{p}{1-p} \\
&= (1-2a) \lg \frac{p}{1-p} \\
&> 0 \text{ iff } a < 1/2
\end{aligned}
$$

□

Now we are ready to give a bound on the false positive. Please observe that, unlike the bounds given until now, this bound does not hold for all scenarios. In fact, if $\mu_{i,j}$ is similar to $\mu_{i,i}$ (i.e., both have the form $(p, 1-0, 0)$ for some $p < 1/2$), then we allow i to be erroneously included in $CAND$. We do this to keep our tests simple and hope that the additional heuristics of $Map1$ and $Map2$ will ensure that such i is eliminated from $SUSP$. We will demonstrate the effectiveness of these heuristics with the help of our experiment results.

Proof of Theorem 15(b). Let τ_i be the distribution obtained from $\mu_{i,j}$. We will separately analyse two possible scenarios.

First possibility is that $\mu_{i,j}(\S) > 0$. Then, $\tau_i(\S) \neq 0$ with probability at least $1-\delta$ using Lemma 13. Therefore, with high probability, i will *not* be added to $CAND$ (lines 11–15).

The other possibility is that $\mu_{i,j} = (p, 1-p, 0)$ for some $p > 1/2$ leading to $\tau_i = (r, 1-r, 0)$ for some r. Now, let $\mu_{i,0} = (q, 1-q, 0)$ for some $q > 1/2$. (a) If $q = 1$, then $\mu_{i,i} = (0, 1, 0)$. By Lemma 14, $\Pr[r > 1/2] \geq 1-\delta$. Therefore, with high probability of $1-\delta$, $\tau_i \neq (0, 1, 0)$ and in that case, i will not be added to $CAND$ in lines 11–15 since $D_{KL}(\tau_i\|\mu_{i,i})$ is not even defined for such $\mu_{i,i}$ and τ_i. (b) On the other hand, if $q < 1$, then $d0$ and $d1$ in lines 11–15 are well-defined. But $D_{KL}(\tau\|\mu_{i,0}) \leq D_{KL}(\tau\|\mu_{i,i})$ with high probability (Lemma 16) and in that case, i is not added to $CAND$ in those lines. □

VII. Performance Evaluation

The primary objective of our work was to derive theoretical upper bounds for detecting and diagnosing every non-redundant fault with high probability.

However, we also evaluated our approach for practical scenarios by running simulations to compare our algorithm with the state-of-the-art algorithm by Paler et al. [15]. We used the SMGF model for faults in our experiments, i.e., the faulty gate is assumed to be missing. Due to the obvious difficulty of not having access to an affordable quantum computer, we simulated a run of BTT, say $BTT(i)$ on circuit C^j, by sampling from the distribution $\mu_{i,j}$ in the BTT table. For every circuit and every fault, we ran fault detection algorithms 10,000 times and report their mean. For our experiments, we fixed the probability of error (δ) at 1%, same as that in the work we are comparing with. We set $\lambda = 0.499995$, so every fault was within the threshold for detection. All our programs, written in python, as well as fault-error and BTT tables for our benchmark circuits are available on our website [5].

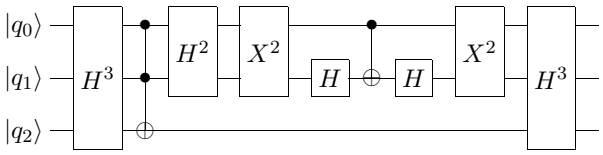


Fig. 5. Benchmark circuit simpleGrover on 3 qubits with 9 gates

The benchmark circuits that we used are 3qubitcnot (illustrated in Figure 4), simpleGrover3 (implementation of Grover's algorithm on 3 qubits and illustrated in Figure 5) and qecc9 (part of QuIDDPro software package [27]). We could not obtain exactly same circuits for qftadd5 and qftadd12 that were used in the compared work, so we include results of our experiments but leave out corresponding results from the previous work. In Table III we have listed their relevant properties along with a histogram of how many faulty gates have error ($\Delta(G, G_f)$) in different intervals. The gates with $\Delta \approx 0$ are very easy to detect, if faulty. On the other hand, the gates with $\Delta \approx 0.5$ would be impossible to detect when faulty and such faults are called as redundant faults. However, it can be observed that none of the single missing gate faults in our benchmark circuits were redundant and moreover, most gates in 3qubitcnot, simpleGrover3 and qecc9 appear to be easily detectable. In contrast, qftadd5 and qftadd12 contains several gates with high Δ; so, it is expected that those circuits will incur a high M-cost for detection and diagnosis.

A. Results for detection

For comparison, we choose the BTT_gen algorithm along with the LRM technique (denoted by BTT+LRM) proposed by Paler et al. [15] which is currently the best fault detection algorithm (it substantially improves upon their earlier work [14]). Even though they were only concerned with false-negatives, we report M-costs for both types of errors.

We chose to implement "Algorithm FN" since the *BTT_gen* algorithm was evaluated only for faulty circuits. Since most faults in the benchmark circuits have low Δ, their corresponding η is quite small. We observed that for such circuits, $\eta_{max} \approx s\eta_{avg}$ (also reported in Table III), so Algorithm FN should be able to detect most of their faults using few samples.

Detailed performance of our algorithm for all 7 types of faults (including "fault-free") for the 6 gates of the 3qubitcnot

[5] https://www.iiitd.edu.in/~dbera/smgf/

TABLE III
BENCHMARK CIRCUITS

Benchmark circuit	qubits	gates	faults with $\Delta = 0$	faults with $0 < \Delta \leq 0.05$	faults with $0.05 < \Delta \leq 0.46$	faults with $0.46 < \Delta < 0.5$	faults with $\Delta = 0.5$ (redundant)	$\max \Delta$
3qubitcnot	3	6	4	0	2	0	0	0.40000
simple-Grover3	3	9	9	0	0	0	0	0.00000
qecc9	9	60	25	35	0	0	0	0.00006
qftadd5 (*)	5	15	5	0	10	0	0	0.45099
qftadd12 (*)	12	78	12	0	26	4	0	0.49967

TABLE IV
COMPARISON OF FAULT DETECTION ALGORITHM (ALGORITHM FN) ON BENCHMARK CIRCUITS.

Benchmark circuit	Our		BTT + LRM [15]		Additional statistics (our)					
	False -ve	True +ve M-cost (median)	False -ve	True +ve M-cost	False +ve	True -ve M-cost	True +ve M-cost (q3)	True +ve M-cost (max)	η_{max}	η_{avg}
3qubitcnot	0 fault (99% times)	2.3	1 fault	50	1%	273	18.1	156.4	237	46
simple-Grover3		1.2	0 fault	40	0%	9	1.7	2	1	1
qecc9		1.3	8 faults	600	0.3%	60	1.9	12	1	1
qftadd5 (*)		4.8	n.a.		3.8%	1659	16.1	178.1	954	111
qftadd12 (*)		261.8	n.a.		27.8%	3.8×10^7	5381	3.1×10^7	2.1×10^7	4.8×10^5

TABLE V
FAULT DETECTION OF 3QUBITCNOT CIRCUIT

Type of fault	Fault-free	Faulty					
	C^0	C^1	C^2	C^3	C^4	C^5	C^6
Δ	-	0	0	0	0.25	0.4	0
Error (%)	1.1	0	0	0	0	0.3	0
M-cost (if correct output)	192	1	1.3	1.5	17.9	126.2	3.2

TABLE VI
HISTOGRAM OF (MEDIAN) SUSPECT SET SIZES

Benchmark circuit	faults with $\lvert SUSP \rvert = k$				Additional stats.	
	Our		BTT + LRM [15]			
	$k=1$	$k=2$	$k=1$	$k=2$	η_{avg}	η_{max}
3qubitcnot	6	0	1	5	1267	5888
simple-Grover3	8	1	5	4	477	4004
qecc9	54	6	n.a.		89	1297
qftadd5 (*)	15	0	n.a.		9898	20466

circuit are presented in Table V. Mean M-cost of the runs only when the output is correct is reported. Not only all faults are detectable but most of the faults are perfectly detectable and they incur a small M-cost (within 20). Also, as expected M-cost increases significantly for faults with large error (e.g., for C^5). This is in contrast to the fact that the earlier technique BTT_gen with LRM used 50 samples but could not detect one faulty gate. Though the specific fault was not reported, we suspect it was G^5 (of type $R_z(\pi/16)$) and the reason is that 50 samples are too few for C^5.

In Table IV, we give a summary of how our algorithm performs vis-a-vis the BTT_gen with LRM technique. M-cost is reported only for those cases in which the output is correct. As expected, false negatives are practically absent. Surprisingly we see that false positives also happen rarely. M-cost for true negative is bound to be high, since *all BTTs* have to be tried before correctly claiming a circuit to be fault-free. Median M-costs (or even 3rd quartile values) for the true positive scenarios are also lower than the earlier results.

For the simpleGrover circuit, we could correctly detect all single-gate faults using merely 2 samples, compared to the 40 samples that were used earlier. All faults of the qecc9 circuit have low Δ and so could be detected using only 12 samples. Note that the earlier technique was not able to detect 8 out of 60 faults even with 600 samples. The comparatively large values for the qftadd5 and qftadd12 circuits should be attributed to many of their gates with Δ between 0.4 and 0.5.

B. Results for diagnosis

In Table VI, we experimentally compare our algorithm with the state-of-the-art technique proposed by Paler et al. [15] (denoted by BTT+LRM) for the benchmark circuits 3qubitcnot and simpleGrover3 under SMGF. We also include results for qecc9 and qftadd5 even though they were not covered in the existing work (Paler et al. considered a different implementation of qftadd5 which is not publicly available). We did not experiment with qftadd12 because many of its gates have $\Delta > 0.49$ which will lead to astronomical samples sizes.

As per Theorem 15, the correct fault for any faulty circuit is included in the list of suspected faults $SUSP$ (with 99% probability) returned by our algorithm. The same has also been reported for BTT+LRM. Therefore, we compare the size of $SUSP$, which is better if smaller and best if $|SUSP| = 1$. It can be seen that both for 3qubitcnot and simpleGrover3, the suspect set is considerably smaller for our algorithm. Even for circuits like qftadd5 which contains several gates with Δ close to 0.5, our algorithm manages to uniquely identify all the faults. Such a high rate of diagnosis comes at the cost of more measurements. We have included the average and maximum η_i used for diagnosis in Table VI and it can be readily seen that there is a wide variation in the various η_i for different gates. This explains the high M-cost of nearly $150{,}000$ for qftadd5 and gives an indication that fewer samples (say, of the order of hundreds) may be insufficient for diagnosing this circuit.

VIII. Conclusion

In this paper we present a clear outline of how one should detect and diagnose single-gate faults in quantum circuits using tomograms. We focus on generating a test set that covers almost all faults of any quantum circuit and give efficient randomised testing and diagnosis algorithms using the test sets. We experimentally show that for single missing-gate faults in some benchmark circuits our approach performs significantly better than the currently known best technique. Our main contribution here is to demonstrate that while studying faults in quantum circuits, one should consider the properties of quantum circuits for properly choosing input states as well as measurement strategies. Since an output of a quantum circuit is probabilistic in nature, tools from statistical hypothesis testing may further improve efficiency of testing.

We believe that our work is an initial response to the exciting challenges brought forth by faults in quantum circuits. It should be noted that our results are applicable to circuits realised by a network of basic quantum gates and may not extend to other physical realisations of the quantum computer. Even for the quantum circuit model, there may be additional challenges arising from the implementation technology; we hope that our work can be suitably extended in such cases. One specific direction that we did not pursue was optimising the size of the test set; in our case this is equal to the number of gates in a circuit but this could be reduced, e.g., by finding tests covering multiple faults. Like that of the reversible circuits, we suspect that the problem of finding the minimum sized test set is a computationally difficult problem.

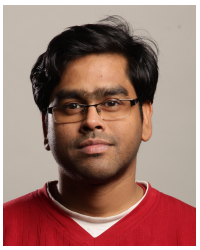
Quantum circuits with noisy gates may require different techniques (e.g., quantum error correcting codes) and were left out from this paper. An important question of quantum circuits is its debugging with respect to its intended function; while not directly applicable, some of the ideas used in diagnosis of faults may be useful in debugging as well. We conclude with the conjecture that the problem of finding the optimal BTT for any gate is NP-hard, possibly by a reduction from the well-known NP-hard quadratic optimisation problem [28].

Acknowledgement

The author thanks the DAE-SEC project "Cryptography & Cryptanalysis: bridge the gap between Classical and Quantum Paradigm" (PI is Subhamoy Maitra). He is grateful to the latter and Susanta Chakraborty for introducing the problem and holding valuable discussions. Sparsa Roychowdhury wrote some of the scripts that were used for simulation. The author also acknowledges the reviewers for providing excellent suggestions that significantly improved the paper.

References

[1] T. Kirkland and M.R. Mercer, Algorithms for Automatic Test-Pattern Generation, IEEE Design and Test: 5(3), 1998.
[2] M. Bushnell and V. Agrawal, Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits, Springer Publishing Company, 2013.
[3] O.H. Ibarra, S.K. Sahni, Polynomially Complete Fault Detection Problems. IEEE Transactions on Computers, **C-24**, 3 (1975).
[4] Ilia Polian and J.P. Hayes, Advanced Modeling of Faults in Reversible Circuits, In Design & Test Symposium (EWDTS), 2010 East-West.
[5] J.P. Hayes, I. Polian, B. Becker, Testing for missing-gate faults in reversible circuits. in Proceedings of the 13th Asian Test Symposium (2004).
[6] I. Polian, T. Fiehn, B. Becker and J. P. Hayes, A Family of Logical Fault Models for Reversible Circuits, 14th Asian Test Symposium (ATS'05), 2005.
[7] R. Wille, H. Zhang and R. Drechsler, ATPG for Reversible Circuits Using Simulation, Boolean Satisfiability, and Pseudo Boolean Optimization, 2011 IEEE Computer Society Annual Symposium on VLSI.
[8] H. Rahaman, D. K. Kole, D. K. Das and B. B. Bhattacharya, On the Detection of Missing-Gate Faults in Reversible Circuits by a Universal Test Set, 21st International Conference on VLSI Design (VLSID 2008).
[9] K. N. Patel, J. P. Hayes and I. L. Markov, Fault testing for reversible circuits, in Proceedings of the 21st VLSI Test Symposium (2003).
[10] Quantum Architectures and Computation Group (QuArC), Microsoft Research, Language-Integrated Quantum Operations: LIQUi.
[11] T. Monz, K. Kim, W. Hänsel, M. Riebe, A. S. Villar, P. Schindler, M. Chwalla, M. Hennrich, and R. Blatt, Realization of the Quantum Toffoli Gate with Trapped Ions, Phys. Rev. Lett. 102, 040501, 2009.
[12] R.B. Patel1, J. Ho, F. Ferreyrol, T.C. Ralph and G.J. Pryde, A quantum Fredkin gate, Science Advances: 2(3), 2016.
[13] M. Perkowski, J. Biamonte and M. Lukac, Test generation and fault localization for quantum circuits in Proceedings of the 35th International Symposium on Multiple-Valued Logic (ISMVL'05) (2005).
[14] A. Paler, A. Alaghi, I. Polian, J.P. Hayes, Tomographic Testing and Validation of Probabilistic Circuits. in Proceedings of the 16th IEEE European Test Symposium, Trondheim (2011).
[15] A. Paler, I. Polian, J. P. Hayes, Detection and diagnosis of faulty quantum circuits. in Proceedings of the 17th Asia and South Pacific Design Automation Conference (2012).
[16] S. Aligala, S. Ratakonda, K. Narayan, K. Nagarajan, M. Lukac, J. Biamonte, and M. Perkowski, Deterministic and Probabilistic Test Generation for Binary and Ternary Quantum Circuits, in Proceedings of the 13th International Workshop on Post-Binary ULSI Systems (2004).
[17] J. D. Biamonte, J. S. Allen, M. A. Perkowski, Fault Models for Quantum Mechanical Switching Networks, Journal of Electronic Testing, **26**, 5 (2010).
[18] Alexandru Paler, PhD Dissertation: Design Methods for Reliable Quantum Circuits, Universität Passau, 2015.
[19] M. Nielsen and I. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, (2000).
[20] C. W. Helstrom, Quantum Detection and Estimation Theory, Journal of Statistical Physics, **1**, 2 (1969).
[21] P. Kaye, R. Laflamme and M. Mosca, An Introduction to Quantum Computing, Oxford University Press, Inc. (2007).
[22] A. Chefles, Quantum State Discrimination, Contemp. Phys. **41**, 401 (2000), Available at http://arxiv.org/abs/quant--ph/0010114.
[23] I. D. Ivanovic, How to distinguish between non-orthogonal states, Phys. Lett. A **123**, 257 (1987).
[24] D. Dieks, Overlap and Distinguishability of Quantum States, Phys. Lett. A **126**, 303 (1988).
[25] A. Peres, How to distinguish between non-orthogonal states, Phys. Lett. A **128**, 19 (1988).
[26] M. Goemans, Lecture notes for course 18.310: Principles of Discrete Applied Mathematics, MIT, USA (2015), http://math.mit.edu/~goemans/18310S15/chernoff-notes.pdf
[27] G.F. Viamontes, I.L. Markov, J.P. Hayes, Quantum Circuit Simulation, Springer (2009).
[28] S. Sahni, Computationally Related Problems, SIAM Journal on Computing 3:4, 262–279, 1974.



Debajyoti Bera received his B.Tech. in Computer Science and Engineering in 2002 at Indian Institute of Technology (IIT), Kanpur, India and his Ph.D. degree in Computer Science from Boston University, Massachusetts, USA in 2010. Since 2010 he is an assistant professor at Indraprastha Institute of Information Technology, (IIIT-Delhi), New Delhi, India. His research interests include computational complexity theory and quantum computing, application of algorithmic techniques in data mining, network analysis & information security.