Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P1941-cd

Evaluating Quantum Algorithms: Closing the Gap between Theory and Practice

Gabriel San Martín Silva
Department of Civil and Environmental Engineering,
Center for Reliability Science and Engineering,
University of California Los Angeles, United States of America,
gsanmartin@ucla.edu

Lavínia Mendes Araújo
Department of Production Engineering,
Center for Risk Analysis, Reliability Engineering and Environmental Modeling (CEERMA),
Universidade Federal de Pernambuco (UFPE), Recife, Brazil
lavinia.mendes@ufpe.br

Isis Didier Lins
Department of Production Engineering,
Center for Risk Analysis, Reliability Engineering and Environmental Modeling (CEERMA),
Universidade Federal de Pernambuco (UFPE), Recife, Brazil
isis.lins@ufpe.br

Enrique López Droguett
Department of Civil and Environmental Engineering,
Center for Reliability Science and Engineering,
University of California Los Angeles, United States of America,
eald@ucla.edu

Motivated by its unique capabilities, quantum computation has gained significant attention over the last decade with numerous models and algorithms proposed for dealing with engineering challenges. The field of risk and reliability has also seen a growing interest in this area, with studies exploring Quantum Machine Learning for remaining useful life, Quantum Optimization for condition monitoring in civil structures, and Quantum Inference for enhancing Bayesian network models, to name a few. However, a common limitation across these works is the lack of thorough comparisons between the proposed quantum algorithms and their state-of-the-art classical counterparts. This critical gap must be addressed not only to evaluate the field's current state reliably but also to guide its development toward the most promising paths for achieving a quantum advantage. There are two key challenges to addressing this gap. First, quantum computation operates on fundamentally different principles than traditional computing, making direct comparisons – such as using the number of iterations – often infeasible. Second, large-scale, error-corrected quantum computers are not yet available, so machine-to-machine comparisons are also not yet possible. In this paper, we detail a novel methodology to evaluate quantum algorithms against their classical counterparts. Our technique is based on a simple observation: quantum computers do not extend the operations that a classical computer can perform. Instead, they have the potential to make them more efficient. As such, when large problems are considered, they ought to present a shorter runtime than classical algorithms to surpass, in any sense of the word, a classical algorithm. We validate the proposed methodology by applying it to an exciting application of quantum computation within the field of system reliability: the identification of minimal cut sets via the leverage of the Grover algorithm and Quantum Amplitude Amplification.

Keywords: Quantum Computation, Performance Evaluation, Grover Algorithm, Quantum Amplitude Amplification, Minimal Cut Set Identification.

1. Introduction

With the promise of enhanced performance, quantum computation has raised significant attention from the research community over the past decade. Initially, this attention was centered within the theoretical computer science and experimental physics communities. However, early applications have now emerged across various fields. The field of risk and reliability is no exception, with examples such as the use of Quantum Machine learning for Prognostics and Health Management (PHM) of rotating machinery (Maior et al. 2023) or quantum-based reliability estimation for power systems (Nikmehr and Zhang 2022).

However, as potential applications gain traction within our community, a highly relevant question arises: do these early algorithms demonstrate evidence – or at least potential – of a computational advantage? Addressing this requires conducting non-trivial specialized tests and comparisons. However, these types of comparisons, while fundamental to advancing the field, are rarely observed in quantum applications within the domain of risk and reliability. This represents a significant gap that must be addressed if quantum computation is to achieve practical utility.

In this paper, we propose a unified methodology for conducting an unbiased comparison between quantum and classical techniques. This methodology is designed to remain agnostic to specific applications or the use of any classical models. The methodology focuses on identifying the number of significant operations required for each quantum algorithm and estimating a lower bound for the runtime of a large-scale quantum computer to execute them. This allows for direct comparison with any classical algorithm, irrespective of its complexity or solution method. The approach is based on a simple observation: quantum computers do not expand the range of operations a classical computer can perform; they merely have the potential to execute them more efficiently. Consequently, for large and practical problems, quantum algorithms must demonstrate shorter runtimes than their classical counterparts to prove superior utility.

The methodology is tested on a practical, relevant case study for the field of risk and reliability: the identification of Minimal Cut Set configurations in Fault Trees representing large-scale systems. Using this case study as a basis, we provide an honest assessment of the applicability of this quantum-based technique towards the analysis of engineering systems, and most importantly, define relevant next steps in the field.

The paper is organized as follows. Section 2 provides a mathematical overview of the field of Quantum Computation, with particular emphasis on the Grover algorithm, which is the main quantum algorithm in this study. Section 3 outlines the proposed methodology in general terms. Section 4 introduces the task of Minimal Cut Set identification and applies the comparison methodology to this significant task. Finally, Section 5 presents the concluding remarks and outlines potential avenues for future research.

2. Theoretical Background: Quantum Computation

This section provides a general introduction to the field of quantum computation. It begins by describing the process that a quantum computer performs from a mathematical standpoint. Then, it uses this formulation to define the concept of a *quantum algorithm*. Finally, it specifies the main quantum algorithm used in this paper: the Grover algorithm.

2.1. Mathematical Formulation

We begin by identifying the symbol ψ as a quantum system. The state of the quantum system ψ is represented using the complex vector notation $|\psi\rangle = \sum_{i=0}^{n-1} c_i |e_i\rangle$, where c_i is a complex number and $|e_i\rangle$ is the vector of dimension N full of zeros, with the exception of position $i+1$ which contains a 1. From a physical point of view, the collection $\{|e_i\rangle\}_{i=0}^{n-1}$ represents all feasible states of system ψ, enumerated in an arbitrary but consistent order. The linear combination of these feasible states represents our knowledge (and uncertainty) regarding the current state of the quantum system.

This uncertainty is reflected through the complex coefficients $\{c_i\}_{i=0}^{n-1}$, which encode the probability of finding the quantum state $|\psi\rangle$ in

state $|e_i\rangle$ through the relationship $p(e_i) = |c_i|^2$. An important consequence of this definition is that all quantum systems must consist of unit-length vectors, since the quantum system is bound to be in at least one of its possible states, and therefore $\sum_{i=0}^{n-1}|c_i|^2 = 1$.

The set $\{c_i\}_{i=0}^{n-1}$ can be modified by applying a *quantum operation* to the quantum state. This operation is mathematically represented as the matrix multiplication between a matrix U and the vector $|\psi\rangle$. Two important considerations are noted regarding these matrices. First, they must be unitary to preserve the unit-length property of quantum states. Second, since unitary matrices are closed under matrix multiplication, a quantum operation can be generated as the composition of several unitary matrices multiplied together.

By modifying the set of complex coefficients, a quantum operation modifies the underlying distribution encoded in the quantum state. A quantum computer is a physical machine capable of both generating a quantum state and applying quantum operations to it. By doing this, the core objective is to modify the stochastic behavior of a quantum system to guide it toward states that represent the solution to a particular problem.

To fulfill this objective, practitioners design *quantum algorithms*. The design of an algorithm for a quantum computer is done through the specification of the set of unitary matrices that are multiplied with the quantum system $|\psi\rangle$. In the following section, we delve deeper into this concept.

2.2. Gate-based Quantum Computation

In simple terms, a quantum computer is a machine composed of several two-dimensional quantum systems, commonly referred to as *qubits*. Following common quantum computing nomenclature, we describe qubits with the ket vector $|q\rangle = [c_0 \;\; c_1]^T = c_0|e_0\rangle + c_1|e_1\rangle \in \mathbb{C}^2$. The set of qubits available in a quantum computer is referred to as its *registry*. In a quantum computer, larger quantum states are produced by joining several qubits together. The operation used to join N qubits is the *Kronecker product* between complex vectors: $|\psi\rangle = \otimes_{i=0}^{N-1}|q_i\rangle \in \mathbb{C}^{2^N}$. This operation also results in a unit-length complex vector, but with a dimension that is exponentially larger than any of the vectors used as inputs. Additionally, while theory specifies that quantum states can have any integer dimension, in practice they are limited to *dimensionalities equal to* 2^N, $N \in \mathbb{N}_{>0}$.

Similar practical restrictions are placed over quantum operations. While in mathematical terms any unitary operation is a valid quantum operation, quantum computers implement only a limited subset of them. Usually, these operations are applied over smaller quantum systems, $N \in \{1, 2\}$. These operations are commonly referred to as quantum gates. Quantum algorithms are created by specifying which quantum gates are applied and to which qubits in the system.

The Kronecker product is also used to join quantum gates of reduced dimension into quantum operations that can be multiplied with quantum states of dimension 2^N. Of particular importance for this paper are the quantum operations denominated as Multi-Controlled NOT gates, or by their abbreviation MCNOT. These gates are used to impose conditional probability relationships between qubits, and are fundamental in the implementation of logic structures for quantum computation.

A relevant limitation of quantum computers is related to the information that can be retrieved from a quantum state. Due to physical restrictions, the set $\{c_i\}_{i=0}^{2^N-1}$ cannot be read directly from the machine. As a consequence, since the quantum state cannot be completely determined, the underlying probability distribution cannot be exactly characterized. However, it can be estimated through a process known as *quantum measurement.* During the quantum measurement operation, the quantum state is collapsed to one, and only one of the feasible quantum states contained in the set $\{|e_i\rangle\}_{i=0}^{2^N-1}$. The feasible state to which the quantum state collapses is controlled by the probability distribution defined as $p(e_i) = |c_i|^2$.

This measurement process is *destructive*: when applied, the quantum state is lost from the machine. For this reason, the accurate estimation of the underlying probability distribution depends on the generation and measurement of multiple quantum states in series. This is a potential source of disadvantages for quantum computation, and represents one of the main motivations for this paper.

Since each qubit in the system is a two-dimensional quantum state, their independent measurement can only result in one of two

feasible states, i.e., its measurement result is a binary variable. With this, we can characterize the output of a global measurement operation as a bitstring of length N, where N is the length of the qubit registry. This characterization is fundamental for quantum computation, as it enables the reformulation of the quantum state' probability distribution as one that is defined over all bitstrings of length N, instead of the vectors $\{|e_i\rangle\}_{i=0}^{2^N-1}$. We shall use this characterization for the remainder of this paper.

2.3. Quantum Protocol: Grover Algorithm

The Grover algorithm (Brassard et al. 2002) is one of the most important quantum protocols. Its objective is to selectively increment the sampling probability of a selected set of bitstrings from the set of all bitstrings of length N. In simpler terms, it is used to alter the probability distribution induced by a quantum algorithm, increasing the likelihood of measuring a bitstring that fulfills a set of desirable characteristics. These *desired* bitstrings are identified through the definition of a Boolean function, f, which only returns True when one of the desired bitstrings is used as input. Since the role of this Boolean function is to *mark* the bitstrings we wish to sample with a higher probability, it is commonly referred to as an *Oracle* function.

In mathematical terms, the Grover algorithm is implemented through the application of a unitary matrix known as the *Grover operator*, U_G. The Grover operator is defined as the composition $U_G = U_A^\dagger S_0 U_A S_f$. We explain each of the terms below:

- U_A: is the quantum operator used to initially implement into the quantum system a given probability distribution. This is, it represents all the quantum gates applied to the initial quantum state prior to the application of the Grover algorithm.
- $U_A^\dagger$: is simply the conjugate transpose of U_A.
- S_0: is a diagonal matrix given by $S_0 = diag([1, -1, -1, \dots, -1]) \in \mathbb{C}^{2^N \times 2^N}$. This matrix can be easily generated from a basic set of quantum gates. For a detailed explanation on this procedure, the reader is directed to (San Martin and Lopez 2024).
- S_f: is a quantum operation that implements the oracle function f. The effect of S_f is to invert the phase of all coefficients that can be mapped to one of the desired bitstrings.

If measuring the quantum state prior to the application of the Grover algorithm results in one of the desired bitstrings with a probability p_a, then the probability after the application of the Grover algorithm is computed as $\sin^2\big((2k+1)\theta_a\big)$, where:

- k is the number of times the operation U_G is applied to the quantum state.
- θ_a depends on the initial probability, and is calculated as $\text{asin}\big(\sqrt{p_a}\big)$.

As inferred from the sinusoidal expression, the probability of sampling a desired state can be equal to 1. This happens for the first time for the following particular value of k:

$$k^* = \lfloor \pi/(4\theta_a) \rfloor \tag{1}$$

As shown, the Grover algorithm can be used to enhance the probability of sampling desired bitstrings up to a probability equal to 1. This can be used to accelerate immensely certain sampling and search processes if one is capable of defining a correct Oracle function. However, a question that is often overlooked in the literature is what is the actual computational cost of applying the Grover algorithm. In this paper, we attempt to tackle this question via the application of the following methodology.

3. Proposed Evaluation Methodology

Estimating the resources required by a quantum computer to solve a computational task presents several difficulties. First, quantum computation operates on fundamentally different principles than traditional computing, making direct comparisons – such as using the number of iterations – often infeasible. Second, large-scale, error-corrected quantum computers are not yet available, so exact machine-to-machine comparisons are also not possible as of the time of this writing.

As a solution, we implement an alternative strategy that returns a lower bound of the time cost it would require an ideal quantum computer. This time can be used as an idealized cost and compared against the time that it takes a classical algorithm to solve the same task in a traditional computer. While not definitive, and definitely not

exact, this comparison will shed light on whether current quantum algorithms can achieve any advantage. To justify this, we note a simple observation: quantum computers do not extend the operations that a classical computer can execute, instead, they have the potential to make them more efficient. As such, when large problems are considered, they ought to present a shorter runtime than classical algorithms in order to surpass, in any sense of the word, a classical algorithm.

To estimate a lower bound on the execution time, an important consideration is that quantum algorithms are usually dominated by a limited set of quantum operations. These operations commonly fulfill two characteristics. First, they are applied over the whole qubit registry and, therefore, cannot be parallelized. Second, to generate them from the set of basic one and two-qubit quantum operations, they need to be decomposed into an exponential number of gates (Barenco et al. 1995; Da Silva and Park 2022). In general, for most algorithms these operations are recognized as Multi-Control NOT operations.

To estimate the time these relevant operations take in a quantum computer, we make use of the information provided by Microsoft Quantum through their publicly available software stack (Beverland et al. 2022). While the execution time is bound to decrease as quantum computers are enhanced and further developed, they offer a realistic estimate of current quantum performance. Table 1 presents the estimated times required by an MCNOT gate as a function of the qubits over which it is applied. It is interesting to note that this time increases linearly with the qubit count, and therefore interpolation can be used to accurately extend the results shown in Table 1 to any algorithm, even those that cannot be executed today due to resource limitations.

We conclude this section by noting two relevant considerations. First, since quantum algorithms can be composed of thousands, if not millions of quantum gates, the implemented technique is purposefully designed as an *ad-hoc* methodology: it demands the analysis of each particular algorithm to identify those operations that control the total execution time. Second, a fundamental stage in the methodology is the multiplication of the estimated execution time by the number of executions required to solve the task. Due to the stochastic nature of quantum computation and the destructive result of the measurement operation, multiple executions in series are needed to accurately estimate the underlying probability distribution. As such, the total time required to solve a task can be computed as shown in Eq. (2).

$$T_T = T_E \times N_E \tag{2}$$

Where $T_T, T_E,$ and N_E stand for the total time, the execution time, and the number of executions, respectively. In the following section, we validate and exemplify the technique for the case of Quantum Fault Trees and Minimal Cut Set Identification.

Table 1. Estimated time to execute a Multi-Control NOT gate using one target qubit and n control qubits. Estimations from the Quantum Resource Estimator (Microsoft Quantum, (Beverland et al. 2022)).

Number of Control Qubits	Time [s]
1	4.00E-06
2	3.10E-05
4	5.80E-05
8	1.30E-04
16	3.34E-04
32	8.11E-04
64	2.00E-03
128	4.00E-03
256	8.00E-03
512	1.70E-02

4. Case Study: Quantum Fault Trees and Minimal Cut Set Identification

This case study considers standard, coherent fault trees with an arbitrary number of basic and intermediary events. We further assume that all basic events can be modeled using Bernoulli random variables. Finally, we assume that both the intermediary events and TOP events are only represented by the logic gates $\{OR, AND\}$. As described in (San Martin, Parhizkar, and Droguett 2022), any fault tree of these characteristics can be easily converted into a quantum algorithm. The resulting quantum model is denominated *Quantum Fault Tree*. A quantum fault tree is composed of a qubit registry of length equal to the count of basic, intermediary, and top events in the

original tree. For brevity, we refer readers to our previous work for details on how this conversion is performed (San Martin and Lopez 2024). Some readers may find particularly interesting the conversion from logical gates to quantum gates that is described in that reference.

The objective of this case study is the identification of minimal cut sets within this fault tree and the posterior evaluation of its performance when compared to classical techniques. The quantum fault tree model can be used for this purpose, although in an indirect manner. To see how, note that any standard, coherent fault tree is just a graphical approach to express a Boolean function. As such, it can be used as an oracle operation within the Grover algorithm to increase the likelihood of sampling system states that verify this Boolean function (i.e., result in the TOP event evaluated as *True*). These states are recognized as cut-set configurations. While this may be useful for reliability quantification applications, it is not useful for minimal cut set identification: the set of cut-sets is considerably larger than the set of minimal cut sets.

In order to make use of the quantum fault tree model for minimal cut set identification, we need to modify the underlying Boolean function to one that is verified only when a minimal cut set configuration is found. This function, denoted as f_{mcs}, is given in Eq. (3),

$$f_{mcs}(\vec{x}_{BE}) = f_{FT}(\vec{x}_{BE}) \wedge \left(\bigwedge_{i \in F} \neg f_{FT}\left(s(\vec{x}_{BE}, i)\right) \right) \quad (3)$$

where f_{FT} represents the original fault tree Boolean function, $\vec{x}_{BE}$ is a given system configuration, defined as the fail/no fail state of the basic events (BE), and $s(\vec{x}_{BE}, i)$ is a *switch* function, which changes the binary state of basic event i only if it originally failed. The reader can verify that this Boolean function is only *True* when the configuration $\vec{x}_{BE}$ represents a minimal cut set.

We apply Eq. (3) in conjunction with the Grover algorithm to increase the likelihood of obtaining minimal cut set configurations. As such, following a similar scheme than the one used to convert f_{FT} into a quantum algorithm, Eq. (3) is also converted to an equivalent quantum operation. For a detailed explanation of this conversion process, the reader is referred to (San Martin and Lopez 2024). The resulting quantum operation is recognized as S_f, previously introduced in Section 2.3. However, while this approach is demonstrated to be able to find minimal cut sets more efficiently than random Monte-Carlo sampling in reference (San Martin and Lopez 2024), it is unclear whether it can obtain significant advantages concerning modern minimal cut set solvers.

In the following section, we apply the methodology described in Section 3 toward this case study, and evaluate how effective is the quantum algorithm in finding minimal cut set configurations when compared against SAT-MCS (Satisfiability-MCS) algorithm (Luo, Wei, and Wan 2021), a state-of-the-art minimal cut set solver. As a basis for this comparison, we use a wide variety of benchmark fault trees derived from the literature (Luo, Wei, and Wan 2021).

4.1. Implementation of the Evaluation Methodology

In this section, the primary objective is to quantify a lower bound for the time it requires the quantum algorithm to find all minimal cut set configurations. The smallest component of the quantum algorithm is the execution and measurement of one Grover operator, $U_G = U_A^\dagger S_0 U_A S_f$. As demonstrated in (San Martin and Lopez 2024), the most time-consuming operations within U_G consist on 2 MCNOT sub-operations over N_{BE} qubits. One of these operations is applied in the construction of S_0, while the second operation is applied in the construction of S_f. The costs of U_A and its conjugate transpose are negligible in comparison. If we denote the execution time of an MCNOT operation as T_{MCNOT}, a lower bound for the execution time of a circuit with one Grover operation applied to it is $2T_{MCNOT}$.

Now, as mentioned in Section 2.3, the use of the Grover algorithm usually requires the application of multiple Grover operations. To simplify the analysis, we assume that the number of Grover operations applied to the circuit is such that the probability of sampling a minimal cut set is 1, i.e., k is chosen as its optimal value: $k^* = \lfloor \pi/(4\theta_a) \rfloor$. To quantify this number, we need to compute the value of θ_a, which in turn depends

on the initial probability of randomly finding a minimal cut set configuration, p_a, through $\theta_a = \sin\left(\sqrt{p_a}\right)$. The probability p_a depends on the total number of minimal cut sets, N_{MCS}, and the number of basic events, N_{BE}, through $p_a = N_{MCS}/N_{BE}$. Of course, the number of minimal cut sets in a fault tree is not known a priori. However, in this paper we use a suite of fault trees for which this quantity is known, and therefore they serve as an excellent benchmark the algorithm.

Summarizing the previous paragraphs, a lower bound for the time required to find a minimal cut set configuration with a probability near 1 is estimated as:

$$T_{MCS} = 2T_{MCNOT} \times \left\lfloor \frac{\pi}{4 \sin\left(\sqrt{\frac{N_{MCS}}{N_{BE}}}\right)} \right\rfloor \quad (4)$$

However, the task at hand is to find *all* minimal cut sets, not only one of them. Since the Grover algorithm increases the probability of finding desired bitstrings uniformly, the measurement of the circuit can return all minimal cut sets with equal probability. As such, we can easily compute the estimated number of measurements required to find all of them with a confidence level equal to α. This number, derived from the coupon collector problem (Neal 2008), is equal to:

$$N_S = \frac{N_{MCS} H_n(N_{MCS})}{(1-\alpha)} \quad (5)$$

where H_n is the n-th harmonic number computed at $n = N_{MCS}$. For the results of this paper, we use $\alpha = 0.99$. By combining Eq. (4) and Eq. (5), we can finally write a lower bound for the total time required to find all minimal cut sets configurations using the quantum algorithm:

$$T_T = N_S \times T_{MCS} \quad (6)$$

This number can be evaluated for all the fault trees used as benchmark in this paper, and compared to the time it takes a traditional computer to find all minimal cut sets using the SAT-MCS algorithm.

4.2. Results

Using a selection of fault trees from the benchmark presented in (Luo, Wei, and Wan 2021), Eq. (6) can be used to estimate a lower bound for the time it would require to find all minimal cut sets using the Grover algorithm. This information is presented in Table 2.

Table 2. Time required to find all minimal cut sets for a set of benchmark fault trees, presented in (Luo, Wei, and Wan 2021), using both the Grover algorithm and a classical state-of-the-art method known as SAT-MCS.

Tree ID	N_{BE}	N_{MCS}	Grover Algorithm [s]	SAT-MCS [s]
baobab3	80	24386	3.50E+14	5.27
chinese	25	392	5.66E+04	0.03
das9201	122	14217	7.94E+20	0.06
das9202	49	27778	5.23E+09	0.04
das9203	51	16200	7.91E+09	0.02
das9204	53	16704	1.67E+10	0.02
das9205	51	17280	8.21E+09	0.03
das9206	121	19518	6.71E+20	0.15
das9207	276	25988	3.75E+44	1.03
das9208	103	8060	6.69E+17	0.56
edf9201	183	579720	1.51E+31	0.30
edf9202	458	130112	3.89E+72	0.72
edf9203	362	20807446	1.92E+59	164.27
edf9205	165	21308	3.98E+27	0.20
edfpa14b	311	105955422	8.56E+51	1851.8
edfpa14o	311	105927244	8.56E+51	1348.0
edfpa14p	124	415500	1.14E+22	2012.9
edfpa14q	311	105950670	8.56E+51	1558.9
edfpa14r	106	380412	1.82E+19	1802.6
edfpa15b	283	2910473	6.47E+46	12.01
edfpa15p	276	27870	3.91E+44	12.57
edfpa15r	88	26549	6.43E+15	12.42
elf9601	145	151348	1.07E+25	270.02
ftr10	175	305	1.05E+28	0.12
isp9602	116	5197647	2.77E+21	0.24
isp9603	91	3434	5.60E+15	0.12
isp9604	215	746574	1.34E+36	0.18
isp9606	89	1776	1.85E+15	0.05
isp9607	74	150436	1.17E+14	0.08
jbd9601	533	14007	2.40E+83	1.20

To aid in the interpretation of the results, Fig. 1 presents a graph of time-to-solution versus the number of basic events in the fault tree, N_{BE}. Note that for the classical algorithm, SAT-MCS, the time-to-solution does not depend on the number of basic events. However, for the quantum-based approach, the time-to-solution is strongly explained through an exponential relationship with N_{BE}.

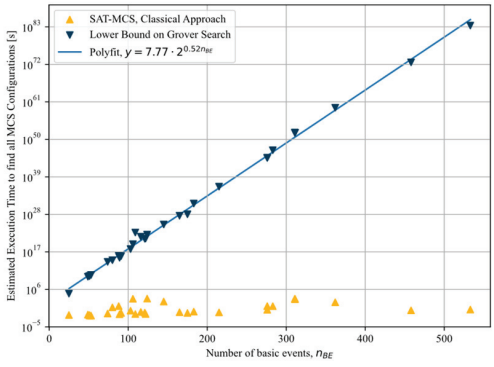

Fig 1. Time-to-solution versus number of basic events for the fault trees presented in Table 1.

5. Discussion and Concluding Remarks

We offer three potential explanations for the lack of performance improvements by using quantum computation for this application.

First, classical algorithms for minimal cut set identification commonly make full utilization of the internal structure of the fault tree, optimizing the search process. The Grover algorithm is agnostic to the fault tree structure, increasing the likelihood of all minimal cut sets uniformly. While this is a desirable property where uniform sampling is needed (for example, in circuit verification), it does not present advantages for this particular task. Second, classical algorithms are usually parallelizable, whereas the Grover algorithm requires the serial application of k^* Grover operators. This operation cannot, by definition, be parallelized and therefore presents one of the main pain-points for Grover-based search algorithms. This is particularly disadvantageous for systems where the initial probability p_a is extremely low, causing k^* to be immensely high. Under these circumstances, even though the application of MCNOT gates is fairly fast, reaching at most the order of 10^{-2} seconds (see Table 1), the whole operation is quickly dominated by the amount of Grover operations needed. Third, the quantum-based algorithm performs a great deal of overwork by finding the same minimal cuts multiple times. While blocking strategies are available, they increase the qubit count of the circuits immensely, reducing the speed of MCNOT gates.

While a quantum approach would perform quadratically faster than a classical one under the assumption of a black-box oracle function and using the number of samples as a metric for performance, these conditions are seldom found in practice. Therefore, the Grover algorithm, as it stands now, is incapable of offering practical advantages for the minimal cut set identification task.

References

Barenco, Adriano, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. 1995. "Elementary Gates for Quantum Computation." *Physical Review A* 52 (5): 3457–67. https://doi.org/10.1103/PhysRevA.52.3457.

Beverland, Michael E., Prakash Murali, Matthias Troyer, Krysta M. Svore, Torsten Hoefler, Vadym Kliuchnikov, Guang Hao Low, Mathias Soeken, Aarthi Sundaram, and Alexander Vaschillo. 2022. "Assessing Requirements to Scale to Practical Quantum Advantage." arXiv. http://arxiv.org/abs/2211.07629.

Brassard, Gilles, Peter Hoyer, Michele Mosca, and Alain Tapp. 2002. "Quantum Amplitude Amplification and Estimation." In , 305:53–74. https://doi.org/10.1090/conm/305/05215.

Da Silva, Adenilton J., and Daniel K. Park. 2022. "Linear-Depth Quantum Circuits for Multiqubit Controlled Gates." *Physical Review A* 106 (4): 042602. https://doi.org/10.1103/PhysRevA.106.042602.

Luo, Weilin, Ou Wei, and Hai Wan. 2021. "SATMCS: An Efficient SAT-Based Algorithm and Its Improvements for Computing Minimal Cut Sets." *IEEE Transactions on Reliability* 70 (2): 575–89. https://doi.org/10.1109/TR.2020.3014012.

Maior, Caio Bezerra Souto, Lavínia Maria Mendes Araújo, Isis Didier Lins, Márcio Das Chagas Moura, and Enrique López Droguett. 2023. "Prognostics and Health Management of Rotating Machinery via Quantum Machine Learning." *IEEE Access* 11:25132–51. https://doi.org/10.1109/ACCESS.2023.3255417.

Neal, Peter. 2008. "The Generalised Coupon Collector Problem." *Journal of Applied Probability* 45 (3): 621–29. https://doi.org/10.1239/jap/1222441818.

Nikmehr, Nima, and Peng Zhang. 2022. "Quantum-Inspired Power System Reliability Assessment." *IEEE Transactions on Power Systems*, 1–14. https://doi.org/10.1109/TPWRS.2022.3204393.

San Martin, Gabriel, and Enrique Lopez. 2024. "Quantum Fault Trees and Minimal Cut Sets Identification." https://doi.org/10.48550/ARXIV.2404.05853.