Exploring Grover-enhanced QAOA for Minimal Cut Set Identification

Gabriel San Martín Silva[a,b,c], Enrique López Droguett[a,b,c]
[a]Department of Civil & Environmental Engineering, University of California, Los Angeles, USA
[b]Center for Reliability Science and Engineering, University of California, Los Angeles, USA
[c]The B. John Garrick Institute for the Risk Sciences, University of California, Los Angeles, USA

Abtract: Fault Tree Analysis is a powerful technique used in a wide variety of fields to perform both quantitative and qualitative analysis of complex systems. Among the capabilities of Fault Tree Analysis, one of the most critical and computationally expensive is the identification of Minimal Cut Sets. With the ever-growing demand for more efficient, sustainable, and safe systems, we can expect the computational complexity of finding minimal cut set configurations to increase even further. To prepare for this challenge, this paper explores the feasibility of finding minimal cut set configurations in standard, coherent fault trees using a novel quantum-based optimization approach. The approach, previously proposed to find satisfying clauses for 3-SAT problems in general settings, is modified to accommodate the context of Fault Tree Analysis. To validate the proposed approach, we perform numerical tests on a quantum computing simulation environment. The results show that while the identification of minimal cut sets is feasible, challenges arise when the size of the fault tree increases.

Keywords: Fault Tree Analysis, Minimal Cut Set Identification, Quantum Computation, QAOA, Grover.

1. INTRODUCTION

Fault Tree Analysis (FTA) is one of the most well-known techniques currently used in reliability engineering to find the root causes of a system's failures. Over the years, FTA has been widely applied across various sectors, including health, nuclear power, aerospace, and many others. One of the most important capabilities of FTA is the identification of Minimal Cut Sets, defined as irreducible configurations of basic events whose failure induces the failure of the overall system [1]. The identification of these important configurations stands as one of the most computationally challenging tasks within FTA. Moreover, with the ever-increasing demand for more efficient, sustainable, and safe systems, the expectation is for the complexity of this task to grow at an accelerated pace.

From a mathematical point of view, the identification of minimal cut sets can be categorized within the broad spectrum of problems known as SAT (Boolean satisfiability) problems. Identifying all the minimal cut sets in a fault tree can be recognized as a subcategory of SAT problems, known as ALL-SAT, where the objective is to find all configurations of Boolean variables that satisfy an expression composed of logical operations. Due to the complexity induced by this problem [2], existing methods present non-linear scaling in their time to solution concerning the number of variables, making the identification of critical scenarios in complex systems a relevant, but very demanding task.

As a novel algorithmic paradigm, Quantum Computation has been heralded over the last few years as a promising candidate to tackle problems that exceed our current computational capabilities. While still nascent, the field has shown exciting results in areas such as Machine Learning [3], Probabilistic Inference [4], and Combinatorial Optimization [5]. This motivates the following question: can these techniques be used in the solution of SAT instances, and in particular, for the identification of Minimal Cut Sets?

The literature shows that general SAT problems have been tackled using two main approaches within the context of Quantum Computation. The first technique applies a sampling approach to the SAT problem, where the objective is to generate a probability distribution that with high likelihood generates satisfying configurations. In this pursuit, these approaches make use of the Grover Algorithm [4], a quantum computing technique designed to selectively increase the likelihood of a set of outcomes in a discrete probability distribution. As an example, Campbell et al. [6] provides an extensive exploration of Grover-like approaches in the solution of constrained satisfaction problems. The second technique uses a combinatorial optimization approach to solve the SAT problem, where an objective function that assigns a higher cost to unsatisfying configurations is generated. In a quantum computation context, this approach is usually solved through the Quantum Approximate Optimization Algorithm (QAOA) [5]. As an example, Boulebnane and Montanaro [7]

propose a framework to use the QAOA for the solution of combinatorial optimization problems with hard constraints, which can be easily related to SAT instances.

However, the application of these techniques to the solution of SAT problems currently has several shortcomings. On one hand, the use of the sampling approaches uses an excessive amount of quantum resources, far exceeding the capabilities of current quantum devices. On the other hand, the QAOA is notorious for getting trapped in local minima of the objective function, a problem known in the literature as "barren plateaus" [8], returning suboptimal solutions that may not even satisfy the original satisfiability problem. For this reason, a recent trend is to combine both approaches to alleviate some of their disadvantages. An example of this combined approach was proposed by Zhang et al. [9] for the general context of 3-SAT problems. However, to the best of the author's knowledge, an experimental verification of this technique within the field of FTA has not been explored.

To address this relevant gap, we adopt the approach proposed by Mandl et al. [10], adapting it with a quantum operator capable of recognizing MCS in Fault Trees. This operator was introduced by the authors in a paper recently published as a preprint [11]. By combining our operator with the approach presented by Mandl et al., we make possible the application of QAOA in the context of FTA. To validate our approach, we perform numerical tests on a quantum computing simulation environment, executed on a traditional computer.

The paper is organized as follows. Section 2 presents the mathematical foundations behind quantum computation, stating the concepts required to understand the rest of the paper. Then, Section 3 describes the original QAOA, as proposed by Fahri et al. in 2014. Section 4 presents the fundamentals behind the QAOA modification proposed by Mandl et al., followed by a brief description of how we apply that technique to the context of FTA. Section 5 presents the case studies used to experimentally verify the proposed technique. Finally, Section 6 described our concluding remarks, outlining future avenues for research in the area.

2. BACKGROUND: QUANTUM COMPUTATION

In this section, we review the general principles of quantum computation. For this, we base our exposition on a probabilistic interpretation of the field, abstracting physics as much as possible from the discussion. For a complete discussion regarding quantum computation, the reader is referred to [12].

2.1. Mathematical Foundations

Through this section, we denote a quantum system with the symbol ψ, and the state in which that system resides as $|\psi\rangle$. The ket notation $|\psi\rangle$ is used in quantum computation to denote a complex vector. The state-space of a quantum system is represented as the finite set $\{|e_i\rangle\}_{i=0}^{N-1}$, where $|e_i\rangle$ is the i-th standard vector of an N-dimensional space. This definition enables us to represent the uncertainty about the current state of a quantum system as a linear combination of its state space, $|\psi\rangle = \sum_{i=0}^{N-1} c_i|e_i\rangle$. With this formulation, the likelihood of finding the system ψ in the state $|e_i\rangle$ is defined as $|c_i|^2$, i.e., the square modulus of the corresponding complex coefficient. For this reason, valid quantum states are unit-length vectors, with a set of complex coefficients $\{c_i\}_{i=0}^{N-1}$ fulfilling the relationship $\sum_{i=0}^{N-1}|c_i|^2 = 1$. The use of complex numbers instead of real numbers to represent uncertainty is one of the main differences between quantum computation and traditional computation.

Quantum states can be modified through their multiplication with unitary matrices known as *quantum operators*. Unitary matrices, denoted as U, are matrices that fulfill the condition $UU^\dagger = U^\dagger U = I$. There are two relevant considerations about unitary matrices within the context of quantum computation. First, unitary matrices preserve the norm of the vectors upon which they upon, maintaining the validity of quantum states. Second, unitary matrices are closed under matrix multiplication, enabling the composition of very intricate operators by multiplying simpler unitary matrices together.

How quantum states and quantum operators are defined enables the use of a very useful probabilistic interpretation of quantum computation. To see this, note that a quantum state can be interpreted as a probability distribution over a discrete set of N outcomes, where N is the dimensionality of the quantum state. This probability distribution is given by $p(e_i) = |c_i|^2, i \in \{0,1,\dots,N-1\}$. Moreover, the transformation of a quantum state by a quantum operator effectively modifies this probability distribution. By carefully designing the operator U, quantum computation enables the redistribution of probability mass towards states that represent the solution of a particular problem. Consequently, the process of determining the coefficients of the matrix U is known as *quantum algorithmic design*.

The remainder of this section is focused on linking these concepts with the practicalities involved in gate-based quantum computation, one of the main hardware implementations of the theory.

2.2. Gate-based Quantum Computation

Quantum computers are physical machines that apply the principles of quantum computation to solve computational tasks. In doing so, they must impose certain limitations on the theory described in Section 2.1. These limitations are reviewed below.

The first limitation is related to the allowed dimensionality of quantum states. Quantum computers are machines that implement large-scale quantum states through the combination of smaller, two-dimensional quantum states known as qubits, denoted as $|q\rangle$. The combination of N qubits into a larger quantum state is computed through the Kronecker product, following Eq. (1). The Kronecker product of N vectors with dimensionalities $d_1, d_2, \dots, d_N$ produces a resulting vector with a dimensionality equal to $d_1 \cdot d_2 \cdot \dots \cdot d_N$. Consequently, quantum states are limited to dimensions equal to $2^N, N \in \mathbb{N}_{\geq 1}$.

$$|\psi\rangle = \bigotimes_{i=0}^{N-1} |q_i\rangle \in \mathbb{C}^{2^N} \tag{1}$$

The second limitation is related to the approach used to generate quantum operators. While in theory any unitary matrix can be used as a valid quantum operator, quantum devices usually implement only a limited number of small-scale unitary matrices. These small-scale unitary matrices, commonly referred to as *quantum gates*, only operate over subsystems composed of at most one or two qubits. The approach used by quantum computers to combine quantum gates into matrices that can be applied to a quantum state also makes use of the Kronecker product. Because of this limitation, the design of quantum algorithms is reduced from defining the coefficients of a $2^N \times 2^N$ matrix to select the order and location for the application of the set of available quantum gates. However, for this paper, we can ignore the complexities imposed by this limitation and focus on the effects produced by applying a quantum operator over a quantum state. For every new quantum operator we introduce, we shall point the interested reader towards a suitable reference that details its implementation as a set of basic quantum gates.

The third, and final limitation produced by the implementation of quantum computation as a physical device involves how quantum states are observed. Due to restrictions imposed by quantum mechanics, the set of complex coefficients that compose a quantum state cannot all be observed simultaneously. Instead, the quantum computer performs a process known as *measurement,* in which the quantum state is collapsed, producing as an output one of the states $|e_i\rangle$ following its underlying probability distribution. Consequently, by repeatedly preparing and measuring a quantum state, the measurement operation provides an approach to estimating the square amplitude of its complex coefficients.

The output of the measurement operation can also be interpreted in an alternative manner. For this, recall that a quantum state is generated as a combination of the independent states of N qubits. Consequently, measuring the quantum state is equivalent to measuring each one of the qubits. As a two-dimensional system, the output of a qubit's measurement can be interpreted as a binary variable. Without loss of generality, let us refer to these states as 0 or 1. Then, it follows that measuring the quantum state results in a bitstring $|x\rangle \in \{0,1\}^N$. Matter of fact, it can be proved that if the measurement of a quantum state results in the state $|e_i\rangle \in \mathbb{C}^{2^N}$, then the corresponding bitstring $|x\rangle$ will match the bit representation of integer i.

This correspondence between the resulting bitstrings and the set of possible states in a quantum system has an additional implication. Since there exists a bijective relationship between the sets of bitstrings $|x\rangle \in \{0,1\}^N$ and the set of standard vectors $\{|e_i\rangle\}_{i=0}^{2^N-1}$, the quantum state can be represented as $|\psi\rangle = \sum_{i=0}^{2^N-1} c_i |x_i\rangle$, where $|x_i\rangle$ is the bitstring representation of integer i. We will use this notation to interpret how a quantum operator modifies the probabilities associated with each measurable bitstring in a quantum state.

3. QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM (QAOA)

The QAOA, originally proposed by Fahri et al. [5], is one of the main techniques used in quantum computation to solve combinatorial optimization problems. Originally designed to solve Quadratic Unconstrained Binary Optimization (QUBO) problems, recent advances have modified the QAOA to enable its application to a wider

class of discrete, combinatorial problems. By encoding a set of binary variables $\{x_i\}_{i=0}^{N-1}$ into the states of an N-qubit registry, the QAOA uses the probabilistic interpretation of quantum states to find a quantum operator that assigns a higher probability mass to those bitstrings that minimize (or maximize) a given objective function.

The QAOA consists of three main steps. In the first step, a special quantum state is generated as $|+\rangle = U_H|0\rangle^{\otimes N}$. In the previous expression, $|+\rangle$ represents the quantum state where all complex coefficients have the same value, $|0\rangle^{\otimes N}$ represent an initial quantum state in which all qubits are deterministically prepared in the state identified as 0, and U_H is the quantum operation that results from joining N Hadamard gates together. For more details about the Hadamard gate and the operation U_H, the reader is referred to [12]. From a probabilistic point of view, this quantum state assigns equal probability to all bitstrings. From an optimization point of view, this is equivalent to assuming that we do not have prior knowledge about which solutions are more optimal, i.e., all possible bitstrings are equally likely solutions to the optimization problem.

The second step consists of the application of the quantum operation U_{QAOA} over the quantum state, assigning higher probability mass to configurations that are deemed more optimal following an objective function F. This operation is formed by repeatedly composing two parameterized sub-operations, $U_{QAOA}(\vec{\gamma},\vec{\beta}) = \prod_{i=1}^{P} U_C(\gamma_i)U_B(\beta_i)$, where $U_C(\gamma_i)$ encodes the objective function of the optimization problem, and $U_B(\beta_i)$ controls which states are visited with a higher frequency during the optimization process. For this reason, the operations U_C and U_B are commonly referred to as *cost* and *mixer* operations, respectively. The repeated application of U_C and U_B can be interpreted as an exploitation and exploration routine over the candidate space, a technique commonly found in other iterative optimization methods. The resulting quantum state after the application of U_{QAOA} is shown in Eq. (2):

$$|\psi(\vec{\gamma},\vec{\beta})\rangle = \left(\prod_{i=1}^{P} U_C(\gamma_i)U_B(\beta_i)\right)|+\rangle \tag{2}$$

The form of the operation U_C is problem-dependent, as it needs to encode the objective function of the original optimization problem. However, the general effect of applying U_C over a quantum state can be written as $U_C(\gamma_i)|\psi\rangle = \sum_{i=0}^{2^N-1} c_i e^{-i\gamma_i F(\vec{x}_i)}|x_i\rangle$, i.e., it applies a complex phase to each possible state that depends on its objective function value. For the application toward minimal cut set identification, we briefly describe its formulation in Section 5. For an overview of the encoding process of general objective functions as quantum operators, the reader is referred to [13].

Eq. (2) clearly shows the dependence of the quantum state, and consequently of its underlying probability distribution, on the parameter vectors $\vec{\gamma}$ and $\vec{\beta}$. The determination of these parameters is the main objective of the third step in the algorithm. For this phase, the quantum state $|\psi(\vec{\gamma},\vec{\beta})\rangle$ is repeatedly prepared and measured, generating a set of candidate solution vectors $\{\vec{x}_i\}_{i=1}^{K}$. The expectation of the objective function as a function of the parameters $\vec{\gamma}$ and $\vec{\beta}$ can be estimated following a traditional Monte-Carlo approach as:

$$\mathbb{E}_{\vec{x}\sim|\psi(\vec{\gamma},\vec{\beta})\rangle}[f(\vec{x})] = \frac{1}{K}\sum_{i=1}^{K} f(\vec{x}_i) \tag{3}$$

Eq. (3) enables the interpretation of the QAOA approach as the transformation of a discrete optimization problem into one that is continuous. This interpretation allows the use of varied continuous optimization algorithms, including powerful gradient-based methods such as Stochastic Gradient Descent, to determine the parameters $\vec{\gamma}$ and $\vec{\beta}$.

However, as mentioned in the introduction, there are three main shortcomings regarding the application of QAOA to SAT problems. First, the cost operation U_C was originally designed to incorporate QUBO functions. These types of expressions are different in nature from Boolean functions and, therefore, a different encoding approach is required. The second shortcoming is the difficulty in finding the parameters $\vec{\gamma}$ and $\vec{\beta}$ through the optimization cycle, in part due to the linear scaling with the number of circuit repetitions, P. The third, and final shortcoming is related to the mixing operator, U_B. It results that in its original form, the application of U_B results in quantum states that assign a different measurement probability to bitstrings with equal objective

function values, adding noise to the computation of Eq. (3). Several authors have proposed approaches to tackle these challenges. While some of these approaches are based on modifying the optimization cycle itself, others are based on changing the forms of U_B and U_C. The algorithm we explore in this paper belongs to the latter category, and it is reviewed in the following section.

4. GROVER-INSPIRED QAOA

Mandl et al. [10] proposed the modification of both the cost and mixing operators to tackle the challenges mentioned in Section 3. These modifications are inspired by a well-known quantum computing algorithm known as the Grover algorithm. For that reason, the resulting algorithm is denoted as "Grover-inspired QAOA". In this section, we describe these modifications and how they are used to tackle each one of the challenges.

For the first challenge, Mandl et al. propose a key modification to the operator U_C to incorporate Boolean functions into the QAOA. Recall that the general effect of the operator U_C over a quantum state $|\psi\rangle$ is given by $U_C(\gamma_i)|\psi\rangle = \sum_{i=0}^{2^N-1} c_i e^{-i\gamma_i F(\vec{x}_i)}|x_i\rangle$, where F is the objective function of the optimization problem. In an SAT context, we define $F: \{0,1\}^N \rightarrow \{0,1\}$ as:

$$F(\vec{x}) = \begin{cases} 1, if\ \vec{x} \in \mathcal{F}_1 \\ 0, if\ \vec{x} \in \mathcal{F}_0 \end{cases} \tag{4}$$

where $\mathcal{F}_1$ and $\mathcal{F}_0$ are two mutually exclusive, collectively exhaustive subsets of the space $\{0,1\}^N$. The approach depends on the generation of a quantum operator U_F, which encodes the original Boolean function F. The operator U_F, when applied over a quantum state composed of $N+1$ qubits, uses the measurement of the first N qubits to compute the result of F, which is stored in the last qubit of the registry. Mathematically, this operation can be written as $U_F \sum_{i=0}^{2^N-1} c_i|x_i, 0\rangle = \sum_{i=0}^{2^N-1} c_i|x_i, F(\vec{x}_i)\rangle$. Under the assumption that the Boolean function of interest can be encoded as a unitary operation to generate U_F, Mandl et al. propose to re-define U_C as:

$$U_C(\gamma_i) = U_F^\dagger P(\gamma_i) U_F \tag{5}$$

where P is a phase operator that selectively applies a phase $e^{-i\gamma_i}$ to those states in which the last qubit would have been measured as 1. For details on the implementation of P from a set of basic quantum gates, the reader is referred to [10]. The application of this modified cost operator over a quantum state results in the following expression:

$$\begin{aligned} U_C(\gamma_i) \sum_{i=0}^{2^N-1} c_i|x_i, 0\rangle &= U_F^\dagger P U_F \sum_{i=0}^{2^N-1} c_i|x_i, 0\rangle = U_F^\dagger P \sum_{i=0}^{2^N-1} c_i|x_i, F(\vec{x}_i)\rangle \\ &= U_F^\dagger \sum_{i=0}^{2^N-1} e^{-i\gamma_i F(\vec{x}_i)} c_i|x_i, F(\vec{x}_i)\rangle = \sum_{i=0}^{2^N-1} e^{-i\gamma_i F(\vec{x}_i)} c_i|x_i, 0\rangle \end{aligned} \tag{6}$$

Note that the operation shown in Eq. (6) divides the original quantum state into two subsets, $\mathcal{F}_1$ and $\mathcal{F}_0$, based on their corresponding complex phase. Moreover, this complex phase depends exclusively on the Boolean function F, thus encoding its effect as an objective function. For a maximum separation between states, Mandl et al. propose to fix $\gamma_i = \pi$, resulting in:

$$U_C(\gamma_i) \sum_{i=0}^{2^N-1} c_i|x_i, 0\rangle = \sum_{x_i \in \mathcal{F}_0} c_i|x_i, 0\rangle - \sum_{x_i \in \mathcal{F}_1} c_i|x_i, 0\rangle \tag{7}$$

Eq. (7) clearly shows that the effect of applying the modified cost operator U_C to the quantum state is to add a negative phase to those bitstrings that verify the original Boolean function, thus encoding the original Boolean function and overcoming the first shortcoming of the traditional QAOA methodology. As an added benefit, by

fixing $\gamma_i = \pi$, this cost operator also halves the number of parameters required in the optimization cycle, eliminating the need for $\vec{\gamma}$, and thus also tackling the second shortcoming.

As mentioned in Section 3, the third shortcoming of the original mixing operator is that it does not assign equal likelihood to states with the same objective function value, resulting in an irregular optimization process. As a solution, Mandl et al. propose the definition of an alternative mixing operator as $U_B(\beta_i) = e^{-i\beta_i|+\rangle\langle+|}$, where $|a\rangle\langle b|$ represents the outer product between the quantum states $|a\rangle$ and $|b\rangle$, and $|+\rangle$ was defined as a uniform quantum state in Section 3. For the implementation of this mixing operator as a series of quantum gates, the reader is referred to [10]. The effect of this modified operator is to assign equal probability amplitude to all bitstrings that have the same objective function value, thus solving the third shortcoming present in the traditional QAOA methodology.

In principle, the approach proposed by Mandl et al. is directly applicable to finding Boolean variable configurations that satisfy a given expression. However, the main gap preventing the application of this technique towards FTA is the lack of a suitable unitary operation U_F. To overcome this gap, we propose the use of our recently developed quantum operator U_{MCS}, capable of identifying minimal cut sets in standard, coherent fault trees, thus enabling the application of G-QAOA towards FTA.

In the remainder of this section, we briefly describe the motivation behind the definition of the operator U_{MCS}. For a detailed algorithm explaining its implementation as a series of basic quantum gates, the reader is referred to our preprint [11]. The definition of our operator U_{MCS} is based on the following definition of a minimal cut set configuration, presented in Definition 1.

Definition 1: Minimal cut set configuration

A given configuration of basic events, represented as a bitstring $\vec{x}$, is a minimal cut set if and only if:

1. *Configuration $\vec{x}$ causes the occurrence of the TOP event, i.e., $\vec{x}$ is a cut set.*
2. *Preventing the failure of any failed basic event in $\vec{x}$ also prevents the failure of the system, i.e., configuration $\vec{x}$ is irreducible as a cut set.*

As seen in section 5, the Grover-inspired QAOA, proposed by Mandl et al., requires a suitable Boolean objective function, F, to be implemented as a quantum operator U_F. In this section, we propose a novel approach to generate a quantum operator U_{MCS}, which encodes a Boolean function $F_{MCS}: \{0,1\}^{N_{BE}} \rightarrow \{0,1\}$ that is only satisfied when the input $\vec{x}$ is a minimal cut set configuration of a fault tree with N_{BE} basic events.

This definition enables us to write the following minimal cut set identifier Boolean function, $F_{MCS}: \{0,1\}^{N_{BE}} \rightarrow \{0,1\}$, shown in Eq. (8).

$$F_{MCS}(\vec{x}) = F_{FT}(\vec{x}) \wedge \bigwedge_{i \in Failures(\vec{x})} \overline{F_{FT}}\big(s(\vec{x}, i)\big) \tag{8}$$

In Eq. (8), F_{FT} is the underlying Boolean function representing the fault tree, $Failures(\vec{x})$ is a collection of the indices of failed basic events in configuration $\vec{x}$, $Failures(\vec{x}) = \{j | x_j = 1\}$, and $s(\vec{x}, i)$ is a *switching* function that flips the state of basic event i in configuration $\vec{x}$.

In a previous paper [14], we described the approach used to encode the fault tree function F_{FT} into a quantum operation U_{FT} over a quantum system composed of three sub registries of lengths N_{BE}, N_{IE}, and 1 qubits. In this encoding process, the state of each basic event is identified with the quantum state of one of the qubits in the first registry. Then, the series of logic gates that define each one of the intermediary events in the tree are encoded into the quantum operator, storing their result in the qubits allocated for the second registry. Finally, the TOP event state is encoded similarly into the state of the third registry.

Note that Eq. (8) requires $|Failures(\vec{x})| + 1$ evaluations of the fault tree Boolean function F_{FT} to identify whether configuration $\vec{x}$ is a minimal cut set. In each evaluation, the input is slightly changed by using the switching function s, to test whether the prevention of one of the failing basic events would prevent the failure of the overall system. The approach used to generate a quantum operation U_{MCS} uses a similar idea by repeating the application of U_{FT} using in each application a slightly different set of qubits representing the basic events. This enables the generation of a quantum state that when measured, carries a minimal cut set identifying signal in one of its qubits. As shown at the beginning of this section, this operation can be used in the approach

proposed by Mandl et al. to directly apply the G-QAOA toward qualitative FTA. In the next section, we describe the case study used to experimentally verify this approach, along with the results obtained by executing the technique in a quantum simulation environment.

5. CASE STUDY AND EXPERIMENTAL VALIDATION

In this section, we present a numerical validation of the approach described in Section 4. For this, we make use of a quantum simulator, which is a specialized software environment that can replicate the operations performed by a quantum computer in a traditional device. The quantum simulator used in this paper is the Python library PennyLane v0.34.0, executed on a computer equipped with 128 GB of RAM.

The fault tree used to validate the proposed approach is depicted in Figure 1. The structure of the fault tree was chosen to enable the increase in the number of basic events and minimal cut sets, without incorporating additional challenges related to its structure.

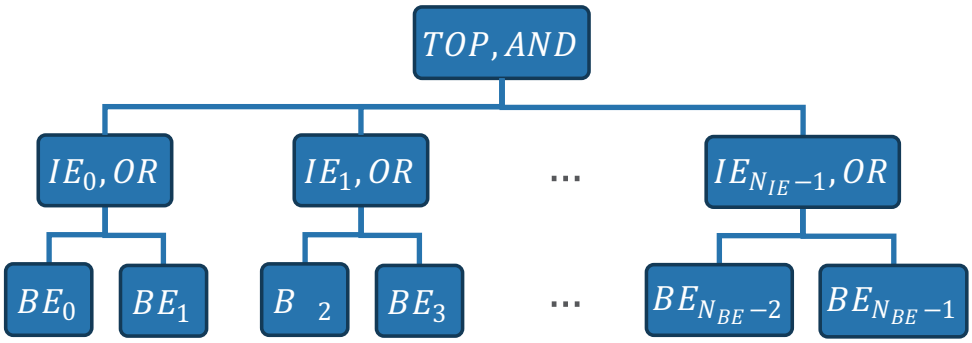
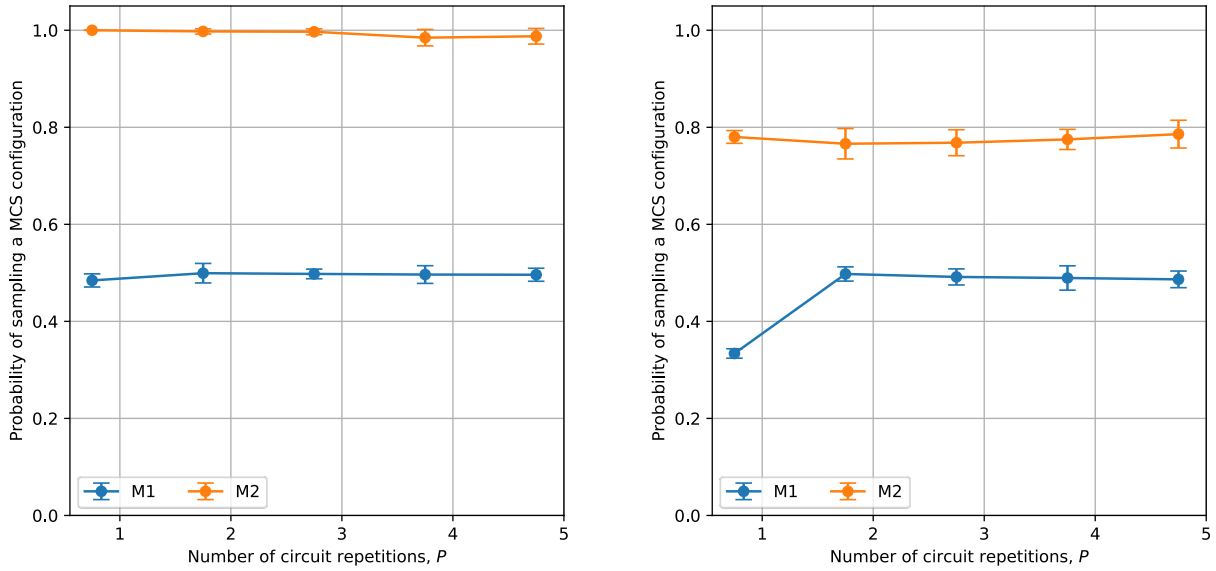


Figure 1. Fault tree structure used in this paper. Note that the number of minimal cut sets in this fault tree is easily computed as $N_{MCS} = 2^{N_{BE}/2}$.

For this validation, we compare two optimization models. The first model, henceforth denoted as M_1, uses U_{MCS} as the quantum operation encodes the objective function, but it does not utilize the modified mixing operator proposed by Mandl et al, using the mixing operator originally proposed for the QAOA. The second model, henceforth denoted as M_2, uses both U_{MCS} as the objective function, and the modified mixing operator. The objectives behind this model selection are two. First, by observing the results of both models independently, we seek to test the hypothesis that quantum-based optimization approaches can be used to identify minimal cut set configurations in fault trees. Second, by comparing the models against each other, we explore whether the modified mixing operator can enhance the optimization process and therefore assign a higher likelihood to minimal cut set configurations.

Due to limitations related to the exponential scaling of quantum states and their impact in memory usage for their simulation, only systems up to 32 qubits can be simulated. This severely limits the fault tree sizes that can be tested numerically to $N_{BE} \leq 10$. However, the optimization process requires for the quantum states to be prepared and measured multiple times. Thus, the number of basic events that can be realistically simulated in our hardware is $N_{BE} \leq 6$. For this reason, we present results concerning two fault trees following the structure presented in Figure 1. The first one use $N_{BE} = 4$, while the second fault tree uses $N_{BE} = 6$. For both fault trees, we compare the average probability of sampling a minimal cut set obtained using either model, after the optimization process is finalized. To study the effect of the number of circuit repetitions on the final sampling probability, we test the cases $P = \{1,2,3,4,5\}$. Since the optimization process may be highly dependent on the initial values for $\vec{\beta}$, we performed 10 rounds of optimization for each value of P, sampling the initial vector $\vec{\beta}$ uniformly from the hypercube $[-\pi, \pi]^P$, reporting both the average result and the associated standard deviation. The optimization technique used in this paper is the *Constrained Optimization by Linear Approximation* (COBYLA) method, a common choice for the QAOA. The results are summarized in Figure 2, where the error bar indicates the standard deviation obtained after the 10 executions.



(a) $N_{BE} = 4$ (b) $N_{BE} = 6$

Figure 2. Average sampling probability obtained with models M_1 and M_2. (a) case where $N_{BE} = 4$. (b) case where $N_{BE} = 6$.

The first fault tree, observed in Figure 2a, consists of $2^4 = 16$ possible configurations of basic events, with only 4 recognized as minimal cut sets. As such, if we would uniformly sample configurations, we would expect to obtain a minimal cut set with a probability of 1/4. Similarly, the second fault tree, observed in Figure 2b, consists of $2^6 = 64$ configurations, with 8 of them being recognized as minimal cut sets. As such, the probability of obtaining a minimal cut set through uniform sampling is 1/8.

The results clearly show that both models achieve sampling probabilities higher than this, indicating that the optimization procedure is capable of recognizing minimal cut set configurations. The analysis of a slightly larger tree, shown in Figure 2b, leads to a similar conclusion. However, We can clearly see a relevant decrease in the sampling probability of minimal cut sets, from ~1.0 in Figure 2a, to ~0.8 in Figure 2b. This is a clear indication that the scale of the fault tree is negatively affecting the complexity of the optimization problem that the G-QAOA is attempting to solve. Nonetheless, note that as the fault tree size grows, so does the number of possible configurations, and therefore the baseline probability given by random sampling. In Fig. 2a, this probability is 1/4, while in Fig 2b, this probability is 1/8. The study of how the sampling probability given by G-QAOA scales with respect to the probability given by uniform sampling of the configuration space would require a larger number of qubits to reach $N_{BE} > 6$, and therefore is left for future research. However, it is sensible to expect that improvements over this method will be required to realize this technique in practical fault trees.

When comparing both models against each other, Figure 2 clearly shows that the M_2 model presents significant advantages over M_1 in terms of the minimal cut sets sampling probability. In other words, once trained, M_2 is more likely than M_1 to return configurations that can be identified as minimal cut sets. This is observed for both fault trees, and all values of the number of circuit repetitions, P. From here, we can conclude that for the cases tested in this paper, the modified mixer operator proposed by Mandl et al. presents improved performance against the mixer operator used in the traditional QAOA methodology.

Moreover, we observe that the standard deviation obtained for all cases is low, even though the initial vector was sampled from a uniform distribution. This indicates that the optimization process easily reaches the same local or global optimum, independently of where it starts. The effect of increasing the number of circuit repetitions from $P = 1$ to $P = 5$ is almost null for the cases tested in this paper. According to the theory, an increment in the value of P should result in a better expectation for the objective value. However, this theoretical assurance does not indicate how large P should be to start observing these benefits. A possible explanation for the results obtained in this paper is that for these fault trees, the required value of P is very high, and thus an increase in performance is not captured for the interval tested in this paper.

6. CONCLUDING REMARKS

This paper proposed using quantum-based optimization to identify minimal cut sets in coherent fault trees. To achieve this, we proposed the integration of a recently developed quantum operation, denoted as U_{MCS}, in the Grover-enhanced Quantum Approximation Optimization Algorithm (G-QAOA) developed by Mandl et al. [10]. This combination allows the experimental exploration of quantum computation in qualitative Fault Tree Analysis tasks.

Overall, the experimental results verify the feasibility of identifying minimal cut sets in standard, coherent fault trees using the G-QAOA methodology. However, due to hardware limitations, this verification was only conducted in relatively small fault tree models. Regarding the use of a modified mixer operation, as proposed by Mandl et al., the results demonstrate significant performance advantages over the mixer operator used in traditional QAOA. Therefore, our main recommendation for future studies in this area is to focus on incorporating this modified operator into newly developed models.

As for future research opportunities in the intersection of quantum computation and Fault Tree Analysis, we mention two currently underexplored ideas. First, the G-QAOA and QAOA approach are notorious for their complicated optimization landscape. Developing methods to navigate this landscape could benefit the parameter initialization process and guide the selection of optimization algorithms. Second, our study highlights a limitation in the size and complexity of fault trees that can be efficiently simulated. The naïve solution to this problem is to use a quantum computer, which would allow us to have a higher number of qubits available, enabling the encoding of larger and more complex Fault Trees. However, error-corrected quantum computers are yet to be commercially available, and therefore as a research community it is imperative to provide solutions that do not depend directly on quantum hardware. For this, we provide as a potential future avenues of research the study of GPU acceleration for quantum simulation, given the exponential advances in capacity that have occurred in that space over the last decade due to the interest on AI/ML applications, to perform large scale quantum operations in a more efficient manner.

References

[1] E. Ruijters and M. Stoelinga, “Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools,” *Comput. Sci. Rev.*, vol. 15–16, pp. 29–62, Feb. 2015, doi: 10.1016/j.cosrev.2015.03.001.

[2] Y. Yu, P. Subramanyan, N. Tsiskaridze, and S. Malik, “All-SAT Using Minimal Blocking Clauses,” in *2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems*, India: IEEE, Jan. 2014, pp. 86–91. doi: 10.1109/VLSID.2014.22.

[3] M. Schuld, “Supervised quantum machine learning models are kernel methods.” arXiv, Apr. 17, 2021. doi: 10.48550/arXiv.2101.11020.

[4] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, “Quantum Amplitude Amplification and Estimation,” vol. 305, 2002, pp. 53–74. doi: 10.1090/conm/305/05215.

[5] E. Farhi, J. Goldstone, and S. Gutmann, “A Quantum Approximate Optimization Algorithm,” 2014, doi: 10.48550/ARXIV.1411.4028.

[6] E. Campbell, A. Khurana, and A. Montanaro, “Applying quantum algorithms to constraint satisfaction problems,” *Quantum*, vol. 3, p. 167, Jul. 2019, doi: 10.22331/q-2019-07-18-167.

[7] S. Boulebnane and A. Montanaro, “Solving boolean satisfiability problems with the quantum approximate optimization algorithm,” 2022, doi: 10.48550/ARXIV.2208.06909.

[8] L. Binkowski, G. Koßmann, T. J. Osborne, R. Schwonnek, and T. Ziegler, “From barren plateaus through fertile valleys: Conic extensions of parameterised quantum circuits.” arXiv, 2023. doi: 10.48550/ARXIV.2310.04255.

[9] Z. Zhang *et al.*, “Grover-QAOA for 3-SAT: Quadratic Speedup, Fair-Sampling, and Parameter Clustering,” 2024, doi: 10.48550/ARXIV.2402.02585.

[10] A. Mandl, J. Barzen, M. Bechtold, F. Leymann, and K. Wild, “Amplitude amplification-inspired QAOA: improving the success probability for solving 3SAT,” *Quantum Sci. Technol.*, vol. 9, no. 1, p. 015028, Jan. 2024, doi: 10.1088/2058-9565/ad141d.

[11] G. San Martin and E. Lopez, "Quantum Fault Trees and Minimal Cut Sets Identification," 2024, doi: 10.48550/ARXIV.2404.05853.

[12] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, 10th anniversary ed. Cambridge ; New York: Cambridge University Press, 2010.

[13] G. San Martín and E. López Droguett, "Quantum-Based Combinatorial Optimization for Optimal Sensor Placement in Civil Structures," *Struct. Control Health Monit.*, vol. 2024, pp. 1–28, Jan. 2024, doi: 10.1155/2024/6681342.

[14] G. S. M. San Martin, T. Parhizkar, and E. L. Droguett, "Quantum Fault Trees," 2022, doi: 10.48550/ARXIV.2204.10877.