NExt ApplicationS of Quantum Computing

D6.12: NEASQC Approximations-like QPSA Algorithms state of the art

Document Properties

Contract Number	951821
Contractual Deadline	2024
Dissemination Level	Public
Nature	Report
Editors	Mohamed Hibti, EDF R&D
Authors	Mohamed Hibti, EDF R&D
Reviewers	Vicente Moret Bonillo, UDC Ansta Ratsima, EVIDEN
Date	09/04/2024
Keywords	Quantum algorithm, Probabilistic Safety Assessment, Approximation
Status	Submitted
Release	1.0

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 951821

History of Changes

Release	Date	Author, Organisation	Description of Changes
1.0	26/04/2024	Mohamed Hibti, EDF R&D	First release after NEASQC review
0.2	04/04/2024	Mohamed Hibti, EDF R&D	Draft after internal review
0.1	09/04/2024	Mohamed Hibti, EDF R&D	First Draft

Abstract

In this NEASQC deliverable, we present an overview of state-of-the-art approximation-like quantum algorithms for solving probabilistic safety assessment problems. Classical approximation algorithms, widely used for real-life complex systems, aim to reduce the search space by pruning branches where cutsets or prime implicants have very low or negligible probabilities. These algorithms also employ various approximations for computing sums of probabilities.

In this document, we outline the state-of-the-art in approximation quantum algorithms that could be applied to probabilistic safety problems.

We have identified numerous approximation algorithms from different classes and quantum routines. Notably, hybrid classical/quantum algorithms—including interpolation, parallelization, cooperative search, a search algorithm via st-network connectivity, quantum walks, and approximate counting—appear promising. They should enhance the resolution of such problems using limited performance quantum hardware. Recent advances in quantum communication have also paved the way for distributed algorithms, which may be particularly relevant in this context. In this document, two distributed algorithms are identified: the first provides a general framework for distributing algorithms across a network of quantum nodes or machines, while the second specifically utilizes a Long algorithm.

List of Acronyms

Term	Definition	
BDD	Binary Decision Diagram	8
CBDD	Minimum Cut-Sets BDD	13
CTQW	Continuous-time quantum walks	11
DTQW	Discrete-time quantum walks	11
FTA	Fault Tree Analysis	8
ETA	Event Tree Analysis	9
GenSAT	Genetic Huill-Climbing Algorithms for Satisfiability	20
GSAT	GenSAT algorithm	20
MFT	Master Fault Tree	6
MICSUP	Minimal Cut Sets Upward	8
MOCUS	Method of Obtaining Cut Sets	8
PRA	Probabilistic Risk Analysis	8
PSA	Probabilistic Safety Assessment	8
QPSA	Quantum PSA	8
QSVT	Quantum Singular Value Transformation	17
SAT	Boolean Satifiability Problem	18
VQE	Variational Quantum Eigensolver	15
ZBDD	Zero suppressed Binary Decision Diagram	13

***Table 1**: Acronyms and Abbreviations*

List of Figures

List of Tables

1. Introduction

In PSAQ use case (for Probabilistic Safety Assessment (PSA) or Probabilistic Risk Analysis (PRA)), we are mainly about two approaches of computing risk metrics. The first, known as Fault Tree Analysis (FTA), has been used since 1961 in Bell Labs for analyzing, visualy displaying and evaluating failure paths in a system [25]. Its objective is to identify all the failure nations that lead to sole undesired events and sum-up their probability/frequency[1]. It has been adopted since 1971 by the Nuclear Power Industry to perform probabilistic safety assessment for nuclear power plants. This is what we generally call the static PSA method. The other is more dynamic and aims to credit the dynamics of the events, their order and the possibility to recover components or systems. Instead of searching for cutsets, this approach is about identifying sequences of events that may apply to a system and lead to a failure state starting from a safe one.

There are many algorithms that were used to for fault tree analysis (see section 2) within commercial or academic software applications; Method of Obtaining Cut Sets (MOCUS) by Fussel and Vesely [42] and [41], Minimal Cut Sets Upward (MICSUP) [46], FATRAM, FTAP and SETS [25]. In the recent software applications, some of these algorithms are based on Boolean fusion over modules with approximation procedures for big size instances (Riskspectrum [59], CAFTA [60], XFTA [58]). Others use Binary Decision Diagram (BDD)-like algorithms [8], [24] or/and approximations mixed with BDD algorithms.

Approximation algorithms are widely used technique to solve probabilistic safety assessment for large scale systems (namely PSA for nuclear power plants). Approximations are mainly needed for prunning the search space and avoid spending a huge time in searching prime impliquants and quantifying the main probabilistic metrics. The main approximation algorithms are based on a modularisation mechanism to reduce complexity, and a kind of cut-off procedure that filters out the cut-sets of "too low/neglected" frequencies or probabilities.

In this NEASQC deliverable, we present an overview of approximation algorithms that may be used or adapted to solve Quantum PSA (QPSA). Even if the quantum algorithms are fundamentally approximation algorithms (the idea of approximating a big unitary to a succession of multiple unitaries), we aim to achieve a state of the art of the approaches where one can solve the QPSA problem by some kind of approximations regarding the following aspects:

- Approximating the space solution by a more reduced space where all the relevant solutions may live.
- Approximating the quantitative metrics such as the probability or the frequency of some event or outcome: when evaluating the accident scenarios leading to some undesired consequence, the objectif is to sum up the probabilities of the individual scenarios to get an estimation of the global probability or frequency of its realisation. This probability is noted P_{top} where top is the name of the top gate of the fault tree. Therefore, one can have an approximation of such probability/frequency P_a such that $||P_a - P_{top}|| \leq \epsilon$ for some sufficiently small ϵ). In the rest of this document we call this the probabilistic approximation criteria. In practice, P_a is obtained by deleting those failure combinations that are below some threshold.
- Approximating the quantum algorithms using, for instance, circuit synthesis, interpolation or parallelization of quantum algorithms.

[1] In PSA, both probabilities and frequencies are used to quantify the likelihood of events, particularly in contexts like nuclear power plants, chemical plants, and other industrial facilities where safety is critical. However, there are distinct differences in how probabilities and frequencies are used and interpreted in these assessments. Probabilities refer to the likelihood of a single event occurring within a specific context or under certain conditions, typically expressed as a fraction between 0 and 1. Frequencies, on the other hand, refer to the rate at which repeated events occur over a specified period or in a specified sample. Frequencies are expressed as occurrences per unit of time (e.g., events per year).

2. Preliminaries

In this section, we present some of the concepts we use in this document.

2.1 Fault tree and event tree analysis

FTA is a systematic, deductive method used to analyze the causes behind potential failures within a system. It's particularly useful in complex systems, like those in aerospace, engineering, nuclear power, and software systems, where identifying potential failure modes is important for safety and reliability and non trivial combined failures may lead to the failure of some system mission.

As an example, consider an Aircraft Engine for which we need to idenify the potential causes of a failure. The top event is the undesirable event we have to analyse. Depending on the complexity of the system, one can choose the main potential macroscopic failures of the system in a top-down approach and use a logical "OR" gate to connect causes that can independently lead to the top event (for instance, *Engine Overheat*, *Fuel System Malfunction* and *Mechanical Failure*) these are then considered as intermediate events for which more detailed causes are searched.

The idea is to express these intermediate events into logical gates that express theses failures in terms of other gates or other very elementary events that we call basic events representing single failures or human factors using logical connectors (e.g. OR, AND, NOR, NAND etc).

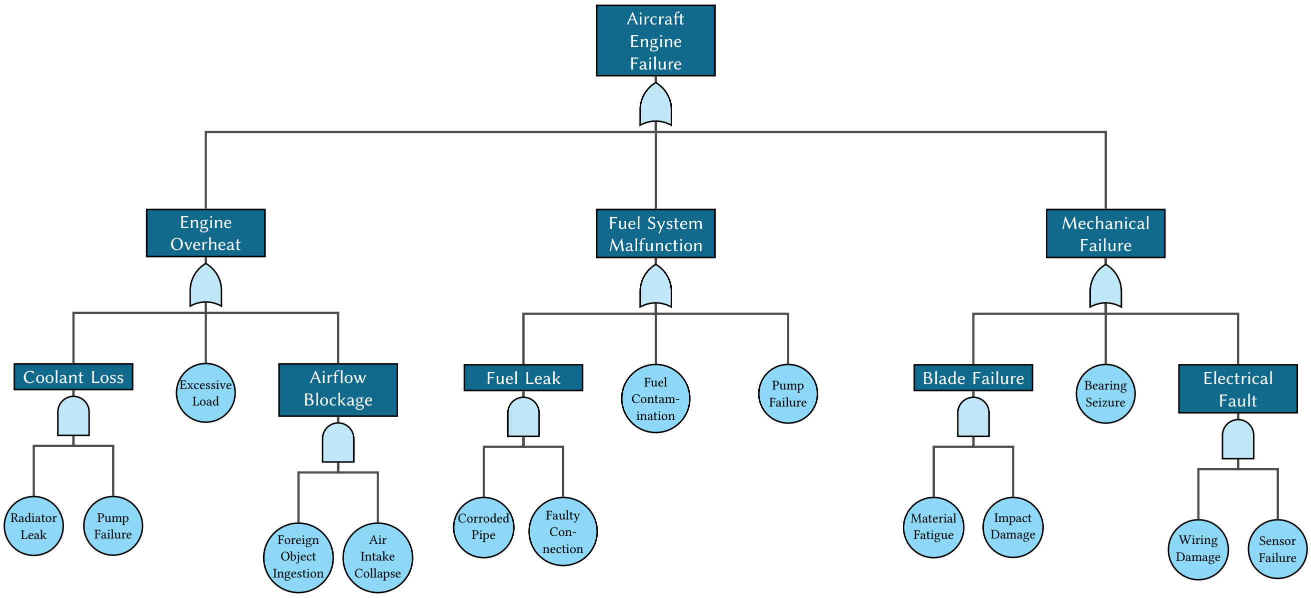


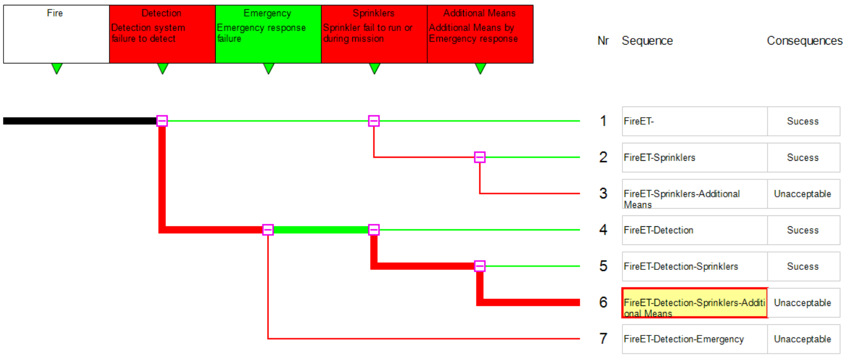
Figure 1: The fault tree corresponding to the failure of the aircraft engine

In the leafs of the fault tree (see figure 1), the basic events have probability of occurrence. In the FTA methodology, the fault tree is quantified to get to top gate probability which gives an indication of how likely the system may fail. In addition to the probability, a qualitative information about the minimal combinations of events that may make the top gate TRUE. We call these combinations minimal cutsets for coherent trees (without negation) or prime impliquants otherwise.

Event Tree Analysis (ETA) is a forward-looking, inductive reasoning technique used in risk assessment and safety management. Unlike FTA, which starts with a specific undesirable event and works backward to identify all possible causes, ETA begins with an initiating event and explores the different outcomes that can result based on subsequent failures or successes of system responses. It helps to map out the various branches of events to see the sequence of consequences and identify mitigation barriers at different stages.

As an example, if we consider a fire in a chemical plant, and assume a fire starting in one of the processing units. One have to identify safety systems and responses that were intended to mitigate such event. Each will have a binary outcome of success or failure. Following the different paths, we can get all the different scenarios that may lead either to acceptable or unacceptable outcomes.

For our example, a simplified event tree will start with the initiator (here the fire) and branch the event tree according to the success or failure of the *Detection System* (*Success*: Fire is detected promptly, *Failure*: Fire goes undetected). Then the Sprinkler System Activation (*Success*: Sprinklers activate and control or extinguish the fire. *Failure*: Sprinklers fail to activate.) and finally the *Evacuation and Emergency Response* (*Success*: Emergency response teams extinguish the fire. *Failure*: Emergency response fails to control the fire.) The missions of the header of the event tree are expressed by fault trees that can explain the root causes of the failures as in example 1.


Fire
Detection
Detection system failure to detect
Emergency
Emergency response failure
Sprinklers
Sprinkler fail to run or during mission
Additional Means
Additional Means by Emergency response

Nr	Sequence	Consequences
1	FireET-	Sucess
2	FireET-Sprinklers	Sucess
3	FireET-Sprinklers-Additional Means	Unacceptable
4	FireET-Detection	Sucess
5	FireET-Detection-Sprinklers	Sucess
6	FireET-Detection-Sprinklers-Additional Means	Unacceptable
7	FireET-Detection-Emergency	Unacceptable

Figure 2: Event tree representing simplfied fire scenarios in a chemical plant

In the case of the ETA methodology, we are interested by the quantitative (probabilities) and qualitative evaluation (cutsets) of the sequences and then the outcomes of interest. In practice, the sequences can be expression in the Boolean logic as fault trees. It is also the case for the consequences which are also expressed as a union of all the sequences that end in it. The corresponding fault tree is called MFT.

For instance, the highlighted sequence in the example 2, can be expressed as a fault tree over the fault trees representing the failures of the systems involved in the mitigation.

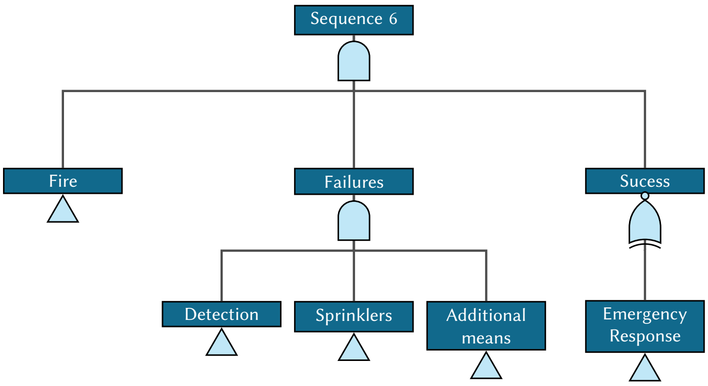


Figure 3: The fault tree corresponding to the sequence 6 of event tree 2 where the leafs in the form of a triangle are transfert gates to to fault trees expressing the failures of the systems/missions in question

2.2 Query complexity

Query complexity measures the efficiency of algorithms or computational processes in terms of the number of queries (or questions) they make to an input. It is a concept primarily used in theoretical computer science and computational complexity theory.

One can define it as the minimum number of queries made by an algorithm to the input in order to solve a specific computational task. It quantifies how much information about the input must be acquired in order to reach a desired output.

Query complexity focuses on understanding how many times an algorithm needs to access or interact with the input data in order to make a decision or compute a result. It is an essential concept in the analysis of algorithms, as it helps assess the efficiency of algorithms in scenarios where minimizing the number of queries is crucial, such as in **database searching**, **decision-making**, or **problem-solving processes**.

Many quantum algorithms such as Shor, Grover and HHL, are phrased in terms of query algorithms.

2.3 Quantum walks

Quantum walks are quantum mechanical analogs of classical random walks. With random walks, the path of the walk is described by stochastic transitions, while in quantum walks these transitions happen in superposition; that is, many positions of the "walker" exist in a probability amplitude. They are fundamental to quantum algorithms, quantum computing, and quantum information processing [7, 12, 5, 6, 31].

Quantum walks can be classified into two main types: Continuous-time quantum walks (CTQW) [26] which are characterized by a continuous evolution of the system and Discrete-time quantum walks (DTQW) [3] that can be separated in a repeated application of two operations called "coin" (acting on the state) and "shift" (acting on the position of the walker) reminding of gate operations in quantum information processing. The behaviour of the quantum walks may then be influenced by a proper choice of the coin and shift operation.

In the following example [66], we can see the evolution of a random walk. In the graph of figure 4, flip a coin in each step, and move to the left if heads, or to the right if tails. Repeat several times to find the probability of being in each position in the final step. The probability of finding the end of each sequence varies. We are not sure that we have found all the paths.

If we consider the corresponding quantum walk, we can see that it can propagate much faster than its classical counterpart (see figure 5, the transitions with the same color apply in a superposition).

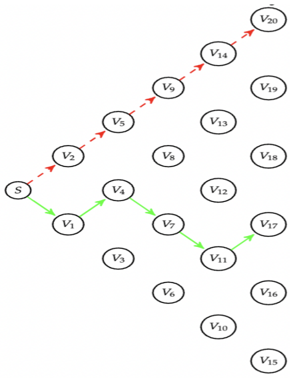


***Figure 4**: Random walk*

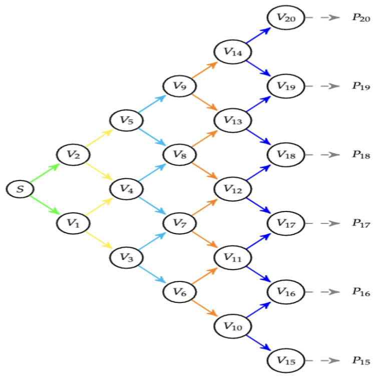


***Figure 5**: Quantum walk*

3. Risk Analysis Computational problems

Computational complexity is one of the major issues regarding probabilistic safety assessment. It is the case for both the conventional static approaches (dealing only with the events logic without any consideration of time neither any dynamic aspects) and dynamic approaches (considering the physical phenomena, the order of the events and their timing).

The computational complexity of PSA models has not only an impact on the running times we can have using classical tools and a wide range of algorithms (MOCUS [41] based or exact approaches such as BDD [52], [40] or ZBDD based approaches [44]), but also on the accuracy of certain computations. In order to prevent optimism and accelerate computations many conservative approximations, sometimes very conservative, are made with unnecessary overestimation of the risk metrics.

Indeed, these algorithms were conceived and implemented in a way to avoid the complexity via a compromise between the running time and their precision.

It is possible to summarize the current computational challenges of PRA with six *central problems* [57]:

- **SAT**: given a boolean formula $f(\mathbf{x})$ over a set of variables, find the valuation(s) that make(s) $f(x) = TRUE$;
- **Reliability**: assuming that the previous formula possesses a related probability structure, find the probability of f, that is, the sum of probabilities of variable valuations such that $f(x) = TRUE$;
- **Reachability** Given a finite state automaton[1] M, assess if there is a *reachable failed state* i.e. identify the sequences of transitions leading to a failed state;
- **FSA-Reliability**: Assuming to have a finite state automaton M with a related probability structure, find the probability to reach a failed state in time t;
- **PA-Reachability**: Given a process algebra model, identify the transition sequences leading to a failed state i.e. find the reachable failure states;
- **PA-Reliability**: Assuming to have a process algebra model M with a related probability structure, find the probability to reach a failed state;

It can be observed that the three problems in which the reliability is not involved can be reduced with various degrees to an initial decision problem (is there a satisfactory variable valuation that fulfill a certain demand?) and a following counting problem (how many satisfactory variable valuations exist?). These two problems are proved to be extremely hard to solve using classical algorithms, as demonstrated by the complexity of each of them: SAT is NP-complete [16], Reachability is PSPACE-complete [51] and the ones involving process algebra are both undecidable [29].

In this document, our focus is on two main reliability problems ; finding cutsets or prime impliquant for a boolean formulae and finding sequences leading from some safe state to a failure state in a Markov graph representing the state space evolution of a system. The first problem can also be seen as a connectivity problem withing an st-network as seen in the case of PSA in [32] or more generally in [43].

[1] Stochastic automata, also sometimes called probabilistic automata, are a type of automaton that incorporates randomness into its behavior. Recall that automata are abstract models of computation that consist of states, transitions between states, and inputs or outputs. They have in common that they deterministically transition between states based on some rules. Therefore, stochastic automata are automata where the transitions between states are not guaranteed, but instead happen with a certain probability. These automata can be used to model dynamic phenomena and are central for the space state method.

4. Approximation approaches in the classical case

Two main categories of algorithms are used in the classical case: Mocus based algorithms and BDD algorithms. In the second category, the representation of fault trees as BDD is somehow quite exact in the sense that the BDD can represent all the cutsets.

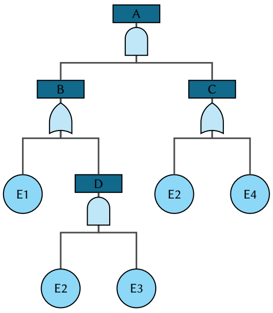
In the first category of algorithms which are mainly used to assess big fault trees (that may represent the failures in complex systems like airplants and nuclear power plants[1], . . .), many approximations are used in different phases of the algorithm:

- While exploring the search space looking for cutsets, some regions are not explored (the correspondig cutsets probabilities are below some threshold).
- While computing non-coherent fault trees (result of the sequence construction by applying negative gates representing success paths), different artifacts (such as delete term) may be used[2].
- The top event probability is computed from the MCSs using the rare-event approximation and even with other techniques such as Zero suppressed Binary Decision Diagram (ZBDD) [44] or Minimum Cut-Sets BDD (CBDD) [10], there is relatively a small part of the cutsets that is considered for the computation.

4.1 MOCUS derived algorithms

These algorithms are mainly based on MOCUS algorithm [41], but with some variants and heuristics to overcome the MOCUS with extension to negative gates and a better optimisation of the algorithm for big instances. MOCUS is based on a top-down approach starting from the top event of a fault tree, using some cutting strategies to remove non-minimal custsets and those whose probability is below some threshold. The intuition of the algorithm is to start constructing the cutsets list in a matrix where the rows and columns start from the top event and are expanded horizontally when meeting an AND gate and vertically when meeting OR gates[3].

For instance, consider the following example:



A =	B AND C
B =	E1 OR D
C =	E2 OR E4
D =	E2 AND E3

Step1.

INITIALISATION

A

Step2.

A = B AND C

B	C

Step3.

B = E1 OR D

E1	C
D	C

Step4.

C = E2 OR E4

E1	E2
D	E2
E1	E4
D	E4

Step5.

D = E2 AND E3

E1	E2	
E2	E2	E3
E1	E4	
E2	E4	E3

The cutsets can then be derived from the resulting list

[1] Actually sometimes depending on the detail level of the model and its size, the BDD (ZBDD) approach is used even for Nuclear Power Plants, however the calculation time may be very important.

[2] Indeed, one want to avoid applying negation over success paths, which is equivalent to applying the negation going through some high level gates to the leaves, this is computationally demanding. The application of the delete term is a also an approximation.

[3] Recall that, there a main assumption regarding to the basic events that are considered in the PSA they are all considered as independent. Therefore, we won't consider any dependency between these events even if in some contexts there may be explicit dependecies that may be considered as such (for instance **ccf!** (**ccf!**)) or implicit dependencies (for instance when dealing with some external hazards which is not the scope of this document).

```
E1,E2
E2,E3
E1,E4
```

`E2,E4,E3` is droped for non minimality.

In this category of algorithms, there are many strategies for solving the MFTs and all these algorithms use approximations in different steps of the algorithms. This include the processing of the search tree for cutset identification (many subtrees are prunned due to the thresholds), the quantification of the top event probability (for instance using approximation of the Sylvestre-Poincarré development), and the treatment of nont-coherent fault trees resulting from the transformation of the event trees into the MFT.

4.2 BDD algorithms

The use of BDD can be found in different steps; in generating cutsets or in summing prime impliquant probabilities. In the first, the whole MFT is solved using a construction of the corresponding BDD and this does not require approximations to generate the set of prime impliquants. The only approximation in such algorithms is when one needs to compute the sum of the probabilities of the prime impliquants which is common to the other approaches. However, the use of BDDs may be found during the quantitative processing of the probabilities of the set of prime impliquants. Many algorithms use BDDs to store the relevant set of prime impliquant (a quite small set of the top of the list) and thus generate a more acurate top probability of the global list avoiding the poor aproximations in the presence of non rare events [10], [45], [68] (which typically occurs when dealing with some external or internal hazards).

4.3 Parallelization efforts

In [62], a parallel algorithm was introduced to leverage the computational power of General-Purpose Graphics Processing Units (GPGPU), enabling the handling and solving of complex fault trees more efficiently and accurately. In this work, a novel gate expansion method was introduced, converting complex gates into simpler ones that are easier to compute in parallel. Moreover, the probabilities of failure are represented in log scale to accommodate very small numbers, enhancing computational accuracy and efficiency.

Splitting models is a practice to reduce the computational complexity, by reducing the scope of the studied consequence analysis cases to some but not all event trees, and take advantage of parallel processes to accelerate analysis. This is done using a naive heuristic that mainly consists in splitting the network corresponding to the MFT into different connex components. In [35], the experience showed that such naive procedure may be relevant for some models without many bridged event trees. However, in some more complex models (typically level-2 models[4]) this strategy may not be peformant.

[4] While level 1 models are about assessing the core damage frequency, level 2 models are dedicated to analyse the containment response to the accident and assesses the likelihood and magnitude of potential radioactive releases to the environment. The initiators of level 2 models are consequences of level-1 event trees and therefore may be more complex and may incorporate many bridged event trees.

5. Approximations within quantum computing algorithms

There are many challenges associated with solving problems using quantum computers. On one side, building and operating quantum computers with significant capabilities regarding their performance, reliability and efficiency[1] is still a hard task. On the other side, solving "exactly" hard problems in this Noisy Intermediate Scale Quantum (NISQ) era is still limited to very small instances. Approximations may be used in both directions to enhance these two problems. Indeed, for the first aspect, approximations are used to model and correct gate errors, ensuring that the overall quantum computation remains accurate, and in noise models to represent the realistic behavior of quantum devices. Moreover, in the process of quantum circuit synthesis, approximations may be used to map high-level quantum gates to a limited set of native gates available on a quantum processor. This involves finding an approximation that preserves the essential characteristics of the desired gate while using the available hardware efficiently.

For the second aspect, other types of approximations are related to the problems being solved and there are many ways to reduce these problems to other more tractable variants using approximations (in the classical or mathematical sense) to solve or approach the desired solutions. For instance, in variational quantum algorithms [28] [1] [2], like the Variational Quantum Eigensolver (VQE) [55], parameterized quantum circuits are used to approximate solutions to specific problems. The variational nature allows the algorithm to iteratively adjust parameters to improve the approximation until a satisfactory solution is found.

In this work, we aim to give an overview of approximation-like algorithms to solve QPSA. We are focusing on the following types of approximations:

- Approximating the space solution by a more reduced space where all the relevant solutions may live.
- Approximating the quantitative metrics such as the probability or the frequency of some event or outcome (get P such that $P - P_{top} \leq \epsilon$ for some ϵ sufficiently small).
- Approximating the quantum algorithms using, for instance, circuit synthesis, interpolation or parallelization of quantum algorithms. Recall that in many hybrid quantum classical algorithms, these techniques are used to limit the use of the quantum routines to small instances and composing the solutions using classical artifacts or in the case of distributed quantum computing make links between qubits of different nodes using different techniques (see section 5.3).

Different algorithms to deal with QPSA were identified in the deliverables D6.8 [34] and D6.18 [33] of the NEASQC project. In this section, we give an overview of approximation algorithms that may be used or adapted to solve QPSA.

5.1 Approximations in circuit synthesis and cutting strategies

In the case of QPSA, one can take advantage of all the generic approaches that are followed to reduce the CNOT count or cutting strategies to reduce the circuit noise and errors [13], [54], [14].

Approaches of cutting circuits into small ones can have two main advantages:

1. Making possible the implementation of large circuits and hence allowing scaling.
2. Small circuits are more robust to noise effects, which, in the case of "good approximations", can help solve instances that connot be solved classically.

A quantum circuit can be represented as a unitary matrix. The objective of the synthesis techniques is to reduce the noise or errors generated in state-preparation, measurement and qubit state decoherence. The main idea is to break the unitary representing the circuit into a succession of simpler unitaries which are close "approximations". The main issue of this approach is th composition of a global solution from partial ones and we will see this is a generic pattern for many research directions in this matter.

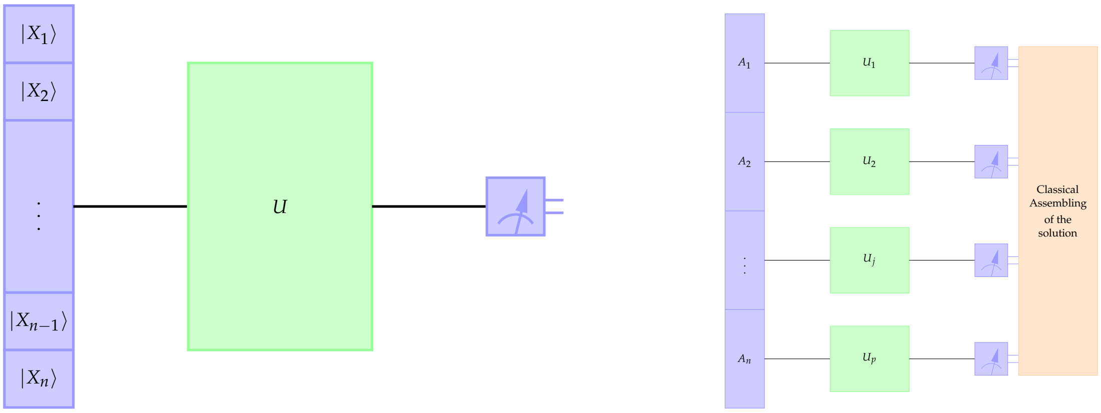
[1] Many metrics were developped to assess these capabilities (Qubit Fidelity or Coherence Time ([50]), Gate Error Rate ([9]), Entanglement Fidelity ([37]), Quantum Volume ([17]), Logical Qubit Error Rate ([9]), Gate Set Uniformity [61], Qubit Connectivity [65], Calibration Time [64]).

5.2 Interpolation and parallelization

One of the well known procedures to deal with quantum algorithms in small-scale quantum devices is to use hybrid approaches. In [49], two approaches were proposed to break down a quantum algorithm into many small fragments with two heuristics; the first is called parallelization and consists in identifying independent smaller instances of a problem that can be solved in parallel. The other is called interpolation and consists in a decomposition of the whole matrix of a given circuit to smaller unitaries that can be run on sub-registers to approximate the original circuit.

Consider a quantum algorithm that can solve some variant of QPSA and assume we have a QPSA problem (it could be a static or a dynamic one). If we consier the algorithm such that there exists a unitary U such that given a register $|x\rangle$ it is transformed in $f(|x\rangle) = U\,|x\rangle$.

The interpolation would be to break down the unitary U in such a way to apply many unitaries U_i to sub-registers (see figure 6). These unitaries should be the same unless the circuit depends on the form of the instance. This interpolation could be done in an exact manner, but could also be approximated in a way to meet the requirements of the probabilistic approximation criteria (see section 1).



(a) We start with a Unitary with a global register

(b) In the case of the interpolation, we have a breakdown of the register to independant registers on which are applied small unitaries and we get fragments of the solutions that have to be assembled classically into a global solution.

***Figure 6**: Breaking a quantum circuit using interpolation approach [49]*

In the case of QPSA, for example for the static part, there are many instances that can be good candidates for this approach. Recall that a preliminary step of solving fault trees is to compress modules and then work with these modules as elementary events before generating what we call modularized cutsets (or modularized prime impliquants). Moreover, depending on the nature of the fault trees, in the case of independence of different parts of the MFT (which in reliability studies is not that frequent though), one can split it into small parts that have to be solved using this interpolation.

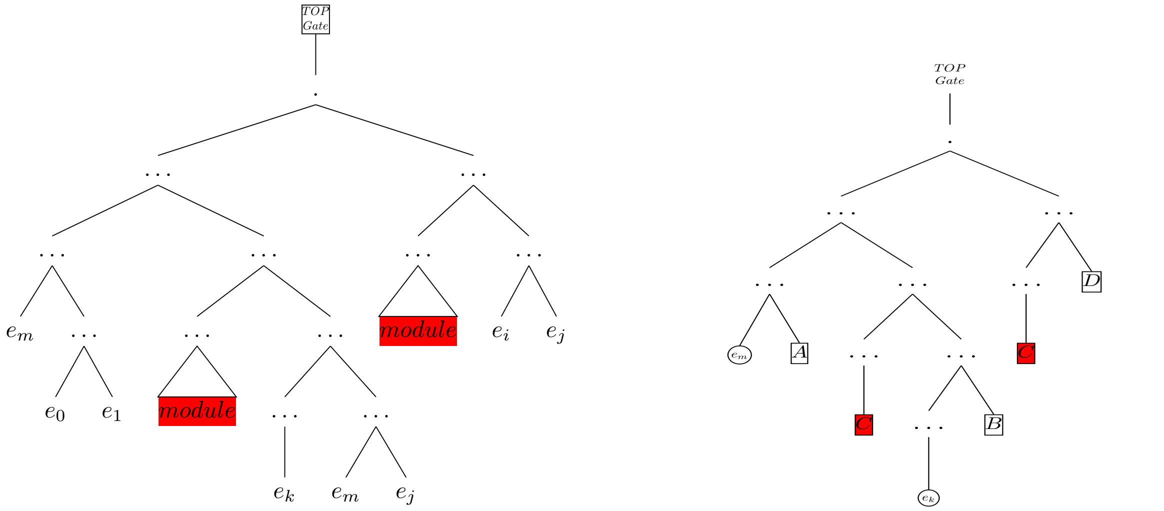
In another case (cf. figure 7a), we can use modules[2] that can be compressed into elementary events and then solve the modules in addition to the compact fault tree. The resulting modularized cutsets are then assembled to get the demodularized cutsets

The second approach introduced in [49], is the parallelization. In this case, one needs to split the instance into small instances that can be solved independently. Therefore, the register is divided in many small registers that are submitted as inputs of different circuits that perform the solving of the small instances.

In [35], a spliting strategy was developed to breakdown MFTs using the model cartograhy[3] (see Figure 8). These fault trees can be solved in parallel and their corresponding cutsets can be added together (sometimes with minimalization see figure 10).

[2] A module refers to a distinct, self-contained component or subset within the fault tree that represents a specific subsystem or part of a system being analyzed for reliability and safety. These modules are used to simplify and manage the complexity of fault trees, especially when dealing with large and intricate systems.

[3] This was implemented in the Andromeda PSA tool (see. [36]).



(a) Initial tree with modules

(b) The correponding tree with module compression

***Figure 7**: Breaking a fault tree into the intersection of its compressed form and the resulting modules*

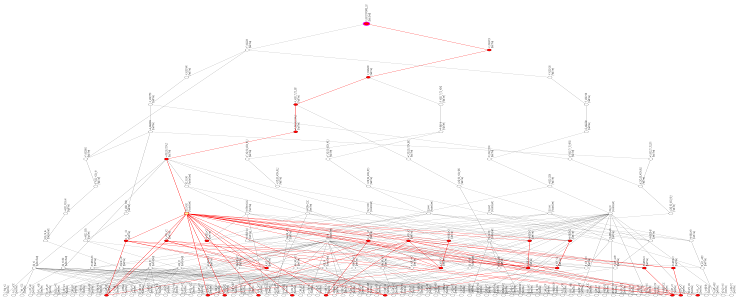


***Figure 8**: Model cartography in the form of a network , the model which includes event trees and fault trees is splitted in a coherent manner so that each structure (event trees, or fault tree) has all its childs in the splitted subtree.*

In the following example, the MFT could be splitted into three subtrees corresponding to the gates B, C and D, after modularizing the gates $D1$ and $D4$ which turn to be equal.

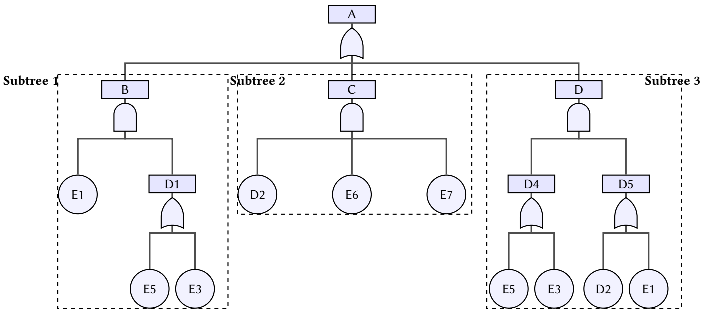
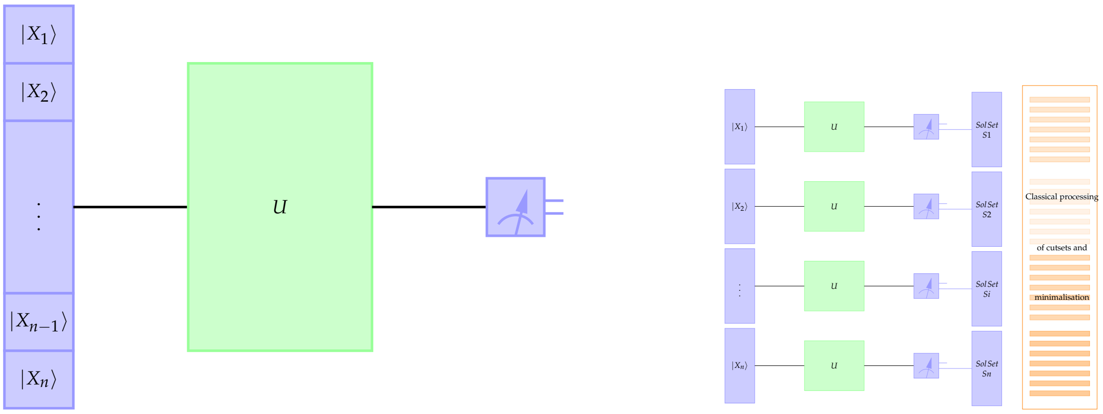


***Figure 9**: The original MFT is splitted into 3 sub trees that may be computed in parallel, their cutsets are then summed up*

In [49], Murça et al. applied the interpolation approach to the problem of general NAND trees (to which our problem could be quickly reduced) and showed that it was better than the parallelization approach using a query model. They used a hybrid approach with an algorithm based on Quantum Singular Value Transformation (QSVT) while in [4] a search algorithm (without any consideration of depth limitation) in top of a quantum walk, which is a clear improvement of the previous approaches for this problem (cf.[26], [38], [11], etc.)



(a) We start with a Unitary with a global register

(b) In the case of the parallelization, we have a breakdown of the register to independant registers corresponding to independant instances, each of which is solved using the unitary U to get a set of solutions. The complete list of cutsets is then assemled using a union of all the solutions sets. A minimalisation process is then applied to filter non-minimal one if needed.

***Figure 10**: Breaking a quantum circuit using parallelization approach [49]*

5.3 Distributed quantum computing

Beyond these hybrid quantum/classical approaches for handling small instances, distributed quantum computing (see [53, 23, 27, 19]) with promising early implementations [18]), represents another promising avenue for QPSA research.

In [69] and [56], distributed versions of Grover's algorithm are proposed. In [69], Zhou et al. consider a distributed Long algorithm [48], which is an improved version of Grover's algorithm [30] that finds the marked/target element with a probability of 100%. They divide the search problem into $\lceil n/2 \rceil$ parts that have to be run in a distributive manner with an iteresting depth reduction which do not increase when n do $(8(n \ mod \ 2) + 9)$.

In [56] a distributed algorithm was proposed to solve Boolean CNF formulae. The algorithm uses a quantum counting algorithm on distributed machines to find the number of solutions to subfunctions f_i of the Boolean intial formula f. These subfunctions are used to reduce the search space of the Boolean function and are defined as follows:

For any $i \in [0, 2^n - 1]$, $y_i \in \{0,1\}^k$ is identified with the binary representation of i, and then define Boolean function $f_i : \{0,1\}^n \to \{0,1\}$ as follows: For any $x \in \{0,1\}^{n-k}$, $f_i(x) = f(xy_i)$.

From lemma 1 of [56], if we consider $f : \{0,1\}^n \to \{0,1\}$, then given $n > k \geq 1$, for any integer $i \in [0, 2^{k-1}]$, denote $a_i = |\{x \in \{0,1\}^{n-k} : f_i(x) = 1\}|$ and $t_a = \lceil 2\pi\sqrt{a} + 11 \rceil$. Then a quantum counting algorithm can output an integer number a'_i such that $a_i \in \{a'_i - t_a, a'_i - t_a + 1, \ldots, a'_i + t_a\}$ with $\lceil \sqrt{2^{n-k}} \rceil$ queries to f_i, and the success probability is at least $\frac{8}{\pi^2}$. In particular, if a $i = 0$ then $a'_i = 0$ is determined with certainty, and if a $i = 2^{n-k}$ then $a'_i = 2^{n-k}$ is determined with certainty.

Once a f_i is not constant to zero, one can use Grover's algorithm to search a solution $x \in \{0,1\}^{n-k}$ such that $f_i(xy_i) = 1$.

The philosphy of this algorithm looks similar to the approach in [22], where Dunjko et al. presented a hybrid algorithm which can sufficiently reduce the problem in question and make it possible to solve 3-Boolean Satifiability Problem (SAT) instances of size $n \gg M$ on a quantum computer with M qubits.

The idea is to consider a quantum computer with $M = cn$ qubits where $c \in (0,1)$ is an arbitrary constant. The algorithm solves the 3-SAT problem with n variables in a time of the order of $O^*(2^{(\gamma_0 - f(c) + \epsilon)n})$ where $f(c) > 0$ is a constant and ϵ can be arbitrarily small. The proposed algorithm is based on the divide & conquer concept, but it is rather a divide & quantum. Indeed, it consists of considering a covering of an instance $x \in \{0,1\}^n$ with balls[4] $B_r(x)$

[4] For metric spaces, a ball is a region bounded by a hypersphere or a sphere representing the neigborhood of the center within some radius according to the corresponding metric. In this context, the hamming distance is considered as a metric.

centered on x (i.e. the set of binary strings y with a Hamming distance[5] less than r). This algorithm reduces r at each step, as soon as r is small enough we can then use a quantum algorithm to solve the formula in the corresponding r-balls.

5.4 Grover and approximate counting

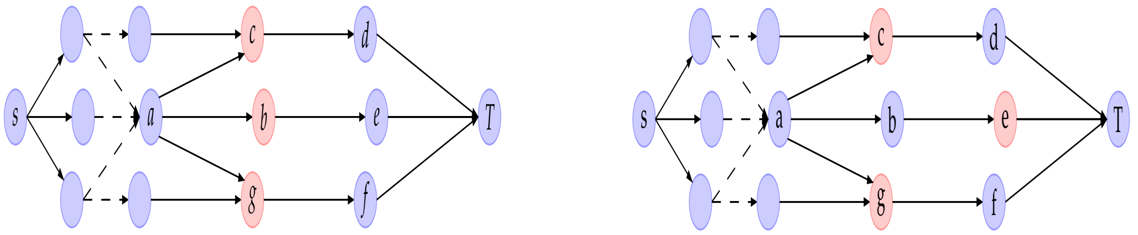
In the D6.8 deliverable [33], we showed that one can use a Grover and Counting algorithm to solve the PSA-problem. Indeed, given a number M of solutions (cutsets) one can easily apply grover algorithm to find all the cutsets with a quadratic speedup.

If we consider approximate counting instead of counting, we can approximate M to within a factor of $1 + \epsilon$, minimizing the number of queries.

Approximate counting aims to estimate the count $M = \#\{x : f(x) = 1\}$ of a given function $f : [N] \rightarrow \{0, 1\}$ to within a factor of $1 + \epsilon$, minimizing the number of queries. The quantum approach allows to estimate M with $O(\mathsf{min}(\sqrt{N/\epsilon}, \sqrt{N/M}/\epsilon))$ queries, improving efficiency over classical methods.

5.5 Via vertex Separators

One of the main approaches we studied in this project is about solving the fault tree analysis problem using a reduction to st-connectivity [67]. In the MOCUS-like algorithms that use approximations [41], [59], [58], some approximation steps correspond to neglecting trees or modules with neglected probabilities. This corresponds to ignoring some vertex separators with very low probabilities in the corresponding st-graph or at least neglecting elementary cutsets that may be generated starting from the st-graph cutsets. Recall that when dealing with st-graphs, in our approach we do compress many failure modes in the same node and thus the graph cutsets are cutsets with aggregated failures. For instance, if the nodes A,B and C have respectively the failure modes $(a_i)_i$,$(b_j)_j$ and $(c_k)_k$, then the cutset $A.B.C$ is a compression of the combination of the $(a_i.b_j.c_k)_{i,j,k}$. Therefore, the approximations that are mainly made correspond to the deletion of the $(a_i.b_j.c_k)$ whith very low probabilities (lower than some threshold).



(a) case 1 with a first cut

(b) case 2 with a neglected cut for instance due the fact that its probability is too low regarding some threshold or relatively to the other cuts

***Figure 11**: Cuts in an st-graph*

Even if some approximations can be applied to reduce the complexity of the algorithms in the case of an st-graph, this is mainly done prior to submitting the instance to the quantum algorithm. Indeed, the classical compression of the graph is performed to reduce its size and in our case without loss of generality.

5.6 Cooperative approach

The cooperative approach, introduced by Cheng and Tao in [15] for solving 3-SAT problems, combines the evolutionary algorithm framework with a Grover search algorithm.

The idea is to use combined registers of quantum and classical variables to limit the action of the quantum search algorithm to few quantum variables by the use of what Cheng and Tao call *auxiliaries*. Auxiliaries are dedicated

[5] The Hamming distance (named after Richard Hamming) between two strings or vectors of equal length is the number of positions at which the corresponding symbols are different. It is one of several string metrics for measuring the edit distance between two sequences. For instance, the Hamming distance between 1011101 and 1001001 is 2, and 3 between 2173896 and 2233796 (cf. Wikipedia).

to classically prepare candidate assignments to search a solution of the complete formula using a Grover search algorithm.

The process of selection of candidate assignments is done within a framework with different strategies Genetic Huill-Climbing Algorithms for Satisfiability (GenSAT) and a modified version of GenSAT algorithm (GSAT), while the selection process of the variables to be considered quantumly is driven by the appearance number of the variables in the formula. Indeed, it is demonstrated in [15] that selecting the variables that have smaller appearance as qubit-variables has higher success probability on finding a solution than selecting the variables with larger appearance. Figure 12 shows the general circuit of the algorithm.

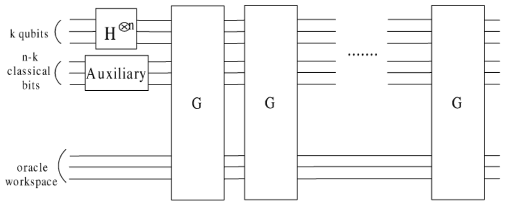


***Figure 12**: Quantum circuit with combined quantum and classical bits. The classical bits are picked at random following one of the gensat strategies and the quantum bit selection driven by the appearance number.*

In this work, different strategies were followed to prepare the auxiliaries. The experimentation showed that the cooperative quantum search algorithm with GENSAT has the best performance in terms of query complexity. Moreover, the optimal configuration (the best number of quantum bits regarding the number of classical variables) for the algorithm is suggested by mathematical analysis.

To consider approximation in this approach would be to involve filtering solutions of cutsets with probabilities below the threshold in the phase of guessing candidates. Indeed, when generating the auxiliaries, one can consider filtering low frequency cutsets (see figure 13). This can also be completed with a classical post treatment, once all the cusets are identified.

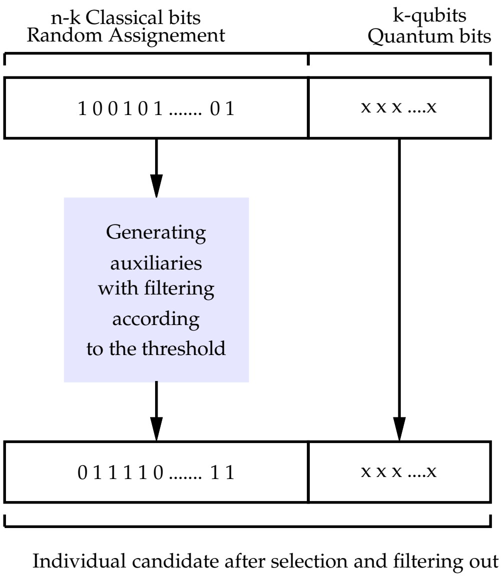


***Figure 13**: When generating auxiliaries, the filtering process is used to avoid generating cutsets with low frequency/probability (Adapted from [15]).*

5.7 State space method and quantum walks

In the literature, different approaches to risk assessment are considered as dynamic[6] ([47], [20], [21],[63], [39]). When dealing with dynamic models, the physical behavior of a complex system and its components is taken into account, in particular how their reliability would evolve due to degradation of a component performance, changes in its operation modes, accidents and other phenomena that would increase the probability of a certain component to fail at, or after, a given time.

In these approaches, the system has different possible *state configurations*, expressed in terms of combinations of failed/operating states of each single component of the system, and the *transition probabilities* to move from a state to another. This method is called *State Space Model* which produces huge models event for medium size real life systems.

In [33], a quantum algorithm was presented to deal with the problem of finding the sequences that lead a certain undesired event. In particular, in the hybrid version (cf. [33] section 6.2.7), an approximation was used to find sequences with a probability below a given threshold. Since the hybridation of the algorithm is based on a recursive search, the removal of the partial sequences with lower probabilities/frequencies has a great impact on prunning parts of the search space. These approximations are quite similar to what is used in the classical case. However, we still take advantage of finding the subsequences using the quantum walk.

[6] These approaches may have different definitions of what they consider as a dynamic framework. Some of the aspects they may have in common are consideration of events order (to better consider passive redundancy and standby systems) and the possibility to consider recoveries and time.

6. Conclusion

In the light of classical approximation algorithms, this NEASQC deliverable was intended to give an overview of the approximation algorithms that can be applied in the quantum framework to solve large scale instances of PSA. There are many classes of algorithms that can be used in this direction. All these algorithms can take advantage of the hybridation and parallelisation or distributive modes.

While, in a large extent, approximation can be found in the hardware level and the area of circuit synthesis and circuit cutting (section 5.1), there can be other ways to approximate QPSA. Indeed, one can approximate the space of the solutions using interpolation and parallelization which are different ways to split quantum circuits. This interpolation can be tuned to match the structure of the MFTs to take advantage of the topology of the trees regarding splitting and composition strategies.

Distributed computing (cf. section 5.3) can also be another interesting direction. In addition to the results showed by [69] in a general case and in [56] for SAT the problem of solving fault trees may be of interest in this approach due to the nature of shared variables by the gates of the so called MFT. Indeed, one can consider these variables as link variables (between distributed nodes) to break the combinatorial complexity that is generated by these variables through the trees.

Other aproaches can also be adopted to leverage approximation algorithms, for instance, cooperative quantum/classical algorithm by Cheng and Tao in [15].

In the dynamic case, there is already an approximated algorithm presented in the NEASQC deliverable D.6.18 [33].

7. Acknowledgements

I gratefully acknowledge the valuable suggestions and constructive criticism provided by the reviewers, which have significantly enhanced the quality of a first version of this deliverable. I would also like to thank Ms Rola Saidi, Emilie Caleca and Mr François Sauvaget for a carreful reading of a first draft of this document and many suggestions. Finally, I'm grateful to Joseph Mickael for his strong support during this project.

Bibliography

[1] D. S. Abrams and S. Lloyd, “Simulation of many-body fermi systems on a universal quantum computer,” *Physical Review Letters*, vol. 79, no. 13, p. 2586, 1997.

[2] D. Aharonov, A. Kitaev, and N. Nisan, “Quantum circuits with mixed states,” in *Proceedings of the thirtieth annual ACM symposium on Theory of computing - STOC ’98*, ser. STOC ’98. ACM Press, 1998. [Online]. Available: http://dx.doi.org/10.1145/276698.276708

[3] Y. Aharonov, L. Davidovich, and N. Zagury, “Quantum random walks,” *Phys. Rev. A*, vol. 48, pp. 1687–1690, Aug 1993. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.48.1687

[4] A. Ambainis, A. M. Childs, B. W. Reichardt, R. Spalek, and S. Zhang, “Any AND-OR Formula of Size N can be Evaluated in time $N^{1/2+o(1)}$ on a Quantum Computer,” in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS’07)*, 2007, pp. 363–372.

[5] A. Ambainis, “Quantum Search Algorithms,” *SIGACT News*, vol. 35, no. 2, pp. 22–35, 2004.

[6] ——, “New Developments in Quantum Algorithms,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, pp. 1–11.

[7] A. Andris, “Quantum Walks and Their Algorithmic Applications,” *International Journal of Quantum Information*, vol. 01, no. 04, pp. 507–518, 2003.

[8] Arboost Technologies, *Aralia User Manual.* Arboost, 2004.

[9] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[10] O. Bäckström and D. Ying, “A presentation of the MCS BDD algorithm in the risk spectrum software package,” in *9th International Conference on Probabilistic Safety Assessment and Management 2008, PSAM 2008*, 2008.

[11] H. Buhrman, R. Cleve, and A. Wigderson, “Quantum vs. classical communication and computation,” in *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, 1998.

[12] E. Campos, S. E. Venegas-Andraca, and M. Lanzagorta, “Quantum tunneling and quantum walks as algorithmic resources to solve hard K-SAT instances,” *Scientific Reports*, 2021.

[13] D. Camps and R. Van Beeumen, “Approximate quantum circuit synthesis using block encodings,” *Physical Review A*, vol. 102, no. 5, Nov. 2020. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.102.052411

[14] Y. A. Chen and X. S. Gao, “Quantum Algorithm for Boolean Equation Solving and Quantum Algebraic Attack on Cryptosystems,” *Journal of Systems Science and Complexity*, 2022.

[15] S. T. Cheng and M. H. Tao, “Quantum cooperative search algorithm for 3-SAT,” *Journal of Computer and System Sciences*, 2007.

[16] S. A. Cook, “The complexity of theorem-proving procedures,” in *Proceedings of the Annual ACM Symposium on Theory of Computing*, 1971.

[17] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, “Validating quantum computers using randomized model circuits,” *Physical Review A*, vol. 100, no. 3, Sep. 2019. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.100.032328

[18] T. Darras, B. E. Asenbeck, G. Guccione, A. Cavaillès, H. Le Jeannic, and J. Laurat, “A quantum-bit encoding converter,” *Nature Photonics*, vol. 17, no. 2, pp. 165–170, 2023.

[19] M. G. Davis, J. Chung, D. Englund, and R. Kettimuthu, “Towards distributed quantum computing by qubit and gate graph partitioning techniques,” 2023.

[20] F. Di Maio, C. Picoco, and E. Zio, “A repairable dynamic event tree framework for the safety assessment of a steam generator of a nuclear power plant,” in *PSAM 2016 - 13th International Conference on Probabilistic Safety Assessment and Management*, 2017.

[21] F. Di Maio, A. Rai, and E. Zio, “A dynamic probabilistic safety margin characterization approach in support of Integrated Deterministic and Probabilistic Safety Analysis,” *Reliability Engineering and System Safety*, 2016.

[22] Y. Dunjko, V. Ge, I. C. J., V. Dunjko, Y. Ge, and J. I. Cirac, “Computational Speedups Using Small Quantum Devices,” *Physical Review Letters*, vol. 121, no. 250501, 2018.

[23] M. Elkin, H. Klauck, D. Nanongkai, and G. Pandurangan, “Can quantum communication speed up distributed computation?” in *Proceedings of the 2014 ACM symposium on Principles of distributed computing*, 2014, pp. 166–175.

[24] EPRI, “FTREX User Manual Version 1.4,” Palo Alto, CA, 2008.

[25] C. A. Ericson, “Fault tree analysis: A history,” in *17th International System Safety Conference*, 1999.

[26] E. Farhi and S. Gutmann, “Quantum computation and decision trees,” *Phys. Rev. A*, vol. 58, pp. 915–928, Aug 1998. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.58.915

[27] D. Ferrari, A. S. Cacciapuoti, M. Amoretti, and M. Caleffi, “Compiler design for distributed quantum computing,” *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–20, 2021.

[28] R. P. Feynman, “Simulating physics with computers,” *International Journal of Theoretical Physics*, vol. 21, no. 6, pp. 467–488, 1982.

[29] J. F. Groote and H. Huttel, “Undecidable Equivalences for Basic Process Algebra,” *Information and Computation*, vol. 115, no. 2, pp. 354–371, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0890540184711011

[30] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proceedings of the Annual ACM Symposium on Theory of Computing*, 1996.

[31] J. Guan, Q. Wang, and M. Ying, “An HHL-based algorithm for computing hitting probabilities of quantum walks,” *Quantum Information and Computation*, 2021.

[32] M. Hibti, “What if we revisit evaluation of PSA models with network algorithms?” ANS PSA 2013 International Topical Meeting on Probabilistic Safety Assessment and Analysis, 2013.

[33] M. Hibti and A. Zaiou, “D6.18: QPSA Quantum Walks and Markov Algorithms,” NEASQC Project, Tech. Rep., 2023.

[34] M. Hibti, A. Zaiou, Y. Bennani, and B. Matei, “D6.8: State of the art of SAT and PSA solvers in the light of quantum computing,” NEASQC Project, Tech. Rep., 2022.

[35] Hibti M., Friedlhuber T., Abdellatif M., “Model Cartography Diving in Complex PSA Models,” *Proceedings of the 30th European Safety and Reliability Conference and 15th Probabilistic Safety Assessment and Management Conference*, 2020. [Online]. Available: http://dx.doi.org/10.3850/978-981-14-8593-0{_}4815-cd

[36] Hibti M., Friedlhuber T. and Rauzy A., “Overview of The OPEN {PSA} Platform,” *PSAM/ESREL 2011*, 2012.

[37] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, “Quantum entanglement,” *Rev. Mod. Phys.*, vol. 81, pp. 865–942, Jun 2009. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.81.865

[38] P. Høyer, M. Mosca, and R. De Wolf, “Quantum search on bounded-error inputs,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2003.

[39] Hu, Y.-S., Modarres, M., “Evaluating system behavior through Dynamic Master Logic Diagram (DMLD) modeling,” *Reliability Engineering and System Safety*, vol. 64, no. 2, pp. 241–269, 1999.

[40] Hung-Yau Lin Sy-Yen Kuo Fu-Min Yeh, “Minimal Cutset Enumeration and Network Reliability Evaluation by Recursive Merge and BDD,” in *Proceedings of the Eighth IEEE International Symposium on Computers and Communication (ISCC’03)*, IEEE, Ed., vol. 03, 2003, pp. 1346–1530.

[41] E. B. H. J. B. Fussell and N. H. Marshall, “{MOCUS} – A Computer program To Obtain Minimal Sets From Fault Trees,” Aerojet Nuclear Corp, Tech. Rep., 1974.

[42] J. B. Fussell and W. E. Vesely, “A New Methodology For Obtaining Cut Sets For Fault Trees,” *Transactions American Nuclear Society*, vol. 15, no. 1, pp. 262–263, 1972.

[43] S. Jeffery and S. Kimmel, “Quantum Algorithms for Graph Connectivity and Formula Evaluation,” *CoRR*, vol. abs/1704.0, 2017. [Online]. Available: http://arxiv.org/abs/1704.00765

[44] W. S. Jung, “ZBDD algorithm features for an efficient Probabilistic Safety Assessment,” *Nuclear Engineering and Design*, 2009.

[45] W. S. Jung and J. Riley, “Strength of ZBDD algorithm for the post processing of huge cutsets in probabilistic safety assessment,” in *PSAM 2014 - Probabilistic Safety Assessment and Management*, 2014.

[46] M. E. S. b. P. C. K. Pande, “Computerized Fault Tree Analysis: TREEL and MICSUP,” Univ. of California, P, 1975.

[47] Labeau, P.E., Smidts, C., Swaminathan, S.,, “Dynamic reliability: towards an integrated platform for probabilistic risk assessment.” *Reliability Engineering and Systems Safety*, vol. 68, pp. 219–254, 2000.

[48] G.-L. Long, “Grover algorithm with zero theoretical failure rate,” *Physical Review A*, vol. 64, no. 2, p. 022307, 2001.

[49] M. Murça, D. Magano, and Y. Omar. (2023) Making the cut: two methods for breaking down a quantum algorithm.

[50] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. New York, NY, USA: Cambridge University Press, 2011.

[51] R. Niskanen, “Reachability Problem for Polynomial Iteration Is PSPACE-complete,” in *Reachability Problems*, M. Hague and I. Potapov, Eds. Cham: Springer International Publishing, 2017, pp. 132–143.

[52] Olivier Nusbaumer, Wolfgang Kruger and Enrico Zio, “Analytical Solutions of Large Fault Tree Models using BDD: New Techniques and Applications,” in *Probabilistic safety assessment and management: PSAM 10*, Seattle, USA, 2010.

[53] R. Parekh, A. Ricciardi, A. Darwish, and S. DiAdamo, “Quantum algorithms and simulation for parallel and distributed quantum computing,” 2022.

[54] T. Patel, E. Younis, C. Iancu, W. de Jong, and D. Tiwari, “Robust and resource-efficient quantum circuit approximation,” 2021.

[55] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, “A variational eigenvalue solver on a photonic quantum processor,” *Nature Communications*, vol. 5, no. 1, Jul. 2014. [Online]. Available: http://dx.doi.org/10.1038/ncomms5213

[56] D. Qiu, L. Luo, and L. Xiao, “Distributed grover’s algorithm,” *Theoretical Computer Science*, vol. 993, p. 114461, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397524000768

[57] A. Rauzy, “Notes on Computational Uncertainties in Probabilistic Risk/Safety Assessment,” *Entropy*, vol. 20, no. 3, 2018. [Online]. Available: http://www.mdpi.com/1099-4300/20/3/162

[58] Rauzy, Antoine, “An Open-PSA Fault Tree engine,” *PSAM/ESREL 2011*, 2012.

[59] Relcon AB., *RiskSpectrum Professional, RSAT users Manual.* Relcon AB., 2003.

[60] SAIC and EPRI, *CAFTA user’s manual, version 2.0*, ser. EPRI (Series). The Institute, 1989. [Online]. Available: http://books.google.com/books?id={_}IwEGwAACAAJ

[61] C. M. Shappert, J. T. Merrill, K. R. Brown, J. M. Amini, C. Volin, S. C. Doret, H. Hayden, C.-S. Pai, K. R. Brown, and A. W. Harter, “Spatially uniform single-qubit gate operations with near-field microwaves and composite pulse compensation,” *New Journal of Physics*, vol. 15, no. 8, p. 083053, aug 2013. [Online]. Available: https://dx.doi.org/10.1088/1367-2630/15/8/083053

[62] F. Sihombing and M. Torbol, “Parallel fault tree analysis for accurate reliability of complex systems,” *Structural Safety*, vol. 72, pp. 41–53, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167473017300875

[63] N. Siu, “Risk assessment for dynamic systems: An overview,” *Reliability Engineering and System Safety*, 1994.

[64] C. Tornow, N. Kanazawa, W. E. Shanks, and D. J. Egger, “Minimum quantum run-time characterization and calibration via restless measurements with dynamic repetition rates,” *Physical Review Applied*, vol. 17, no. 6, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1103/PhysRevApplied.17.064061

[65] P. Yuan, J. Allcock, and S. Zhang, “Does qubit connectivity impact quantum circuit complexity?” 2023.

[66] A. Zaiou, “Quantum machine learning approaches for graphs and sequences : application to nuclear safety assessment,” Ph.D. dissertation, 2022, thèse de doctorat dirigée par Mateï, Basarab Informatique Paris 13 2022. [Online]. Available: http://www.theses.fr/2022PA131048

[67] A. Zaiou, Y. Bennani, M. Hibti, and B. Matei, “Quantum approach for vertex separator problem in directed graphs,” in *Artificial Intelligence Applications and Innovations*, I. Maglogiannis, L. Iliadis, J. Macintyre, and P. Cortez, Eds. Cham: Springer International Publishing, 2022, pp. 495–506.

[68] J. Zhong, J. Tong, and X. He, “An approach to use BDD during the fault tree editing and analyzing,” in *Proceedings of the 8th International Conference on Probabilistic Safety Assessment and Management, PSAM 2006*, 2006.

[69] X. Zhou, D. Qiu, and L. Luo, “Distributed exact grover’s algorithm,” *Journal of High Energy Physics*, vol. 2023, no. 3, p. 001, 2023.