N° d'Ordre :
EDSPIC :

Université Sorbonne Paris Nord
ÉCOLE DOCTORALE GALILÉE

Ph.D. Thesis
by

Ahmed ZAIOU

for the dgree of
Doctor of Computer Science

Quantum Machine Learning Approaches for Graphs and Sequences

Application to Nuclear Safety Assessment

defended on 9th december 2022 in front the following jury:

Thesis supervisors :

Younès Bennani	Professor, Université Sorbonne Paris Nord
Basarab Matei	Associate professor HDR, Université Sorbonne Paris Nord
Mohamed Hibti	Expert, EDF Lab Saclay

Reporters :

Cyrille Bertelle	Professor, Université du Havre
Michel Verleysen	Professor, Ecole Polytechnique de Louvain

Examiners :

Stefano Guerini	Professor, Université Sorbonne Paris Nord
Maria Malek	Associate professor HDR, CY Tech Cergy Paris Université
Nicoleta Rogovschi	Associate professor HDR, Université Paris Cité
Ali Yahyaouy	Professor, Sidi Mohamed Ben Abdellah University

"If you are not completely confused by quantum mechanics, you do not understand it."

John Wheeler

"I think I can safely say that nobody understands quantum mechanics."

Richard Feynman

Résumé

Dans cette thèse, notre objectif est de fournir des solutions moins complexes en utilisant des algorithmes purement quantiques ainsi que des algorithmes d'apprentissage automatique quantiques pour traiter des problèmes dans le domaine des Études Probabilistes de Sécurité (EPS) avec un temps raisonnable. Nous abordons les deux aspects du problème des EPS, statique et dynamique. Pour le problème statique, où nous sommes intéressés à trouver toutes les combinaisons d'événements de base du système qui peuvent générer des accidents graves, nous proposons d'obtenir ces combinaisons d'événements de base par un algorithme quantique, en utilisant des graphes orientés au lieu de chercher toutes les solutions d'un problème SAT. Notre contribution est un algorithme quantique qui utilise un nombre linéaire de qubits et grâce à un filtre classique, nous pouvons trouver toutes les combinaisons d'événements de base qui peuvent générer ces accidents. Dans le cas dynamique, où nous sommes intéressés à trouver toutes les séquences accidentelles d'un système, notre principal intérêt est le traitement de ces séquences. Dans le cas classique, afin de trouver toutes ces séquences, nous utilisons le graphe d'état du système et nous cherchons tous les chemins entre l'état courant et tous les états critiques. Comme ce problème est NP-complet, nous proposons une solution quantique pour trouver tous ces chemins. Nous proposons deux algorithmes quantiques, tous les deux basés sur la philosophie des marches quantiques. Le premier algorithme permet de trouver tous les chemins entre un sommet source et plusieurs sommets destination dans un graphe orienté a-cyclique. Cet algorithme utilise N qubits et M portes afin de trouver tous les chemins. Le second est une version hybride du premier, il est capable de traiter de grands graphes même avec un nombre réduit de qubits. Une autre contribution consiste une approche quantique de l'algorithme Dynamic Time Warping (DTW) afin de calculer la similarité entre ces séquences, ainsi qu'une version capable de trouver la meilleure correspondance entre les séquences en utilisant des sous-séquences dont la longueur varie dynamiquement. Nous proposons également une stratégie d'apprentissage pour les Modèles de Markov Cachés Quantiques (QHMM) afin de générer des scénarios accidentels à partir de n'importe quel état initial du système et également pour gérer le système en temps réel. Nous proposons enfin une version améliorée de k-means quantique. La complexité de chaque itération de la version classique de k-means est de $\mathcal{O}(K \times M \times N)$. Dans notre cas, le calcul de toutes les distances entre les observations et les centres des clusters avec un seul circuit quantique, et l'utilisation de l'algorithme de recherche quantique de Grover, nous permet de réduire la complexité à $\mathcal{O}(\log(K \times M \times N))$. Une autre version de l'algorithme de k-means équilibrés quantiques est aussi proposé en utilisant le quantique adiabatique. Enfin, nous proposons une version quantique de l'algorithme Convex-NMF qui est plus rapide que la version classique. Nous concluons cette thèse par une application de nos approches proposées sur un système réel dans le domaine des EPS.

Abstract

In this dissertation, our goal is to provide less complex solutions using pure quantum algorithms as well as quantum machine learning algorithms, to deal with problems in the field of Probabilistic Safety Assessment (PSA) within a reasonable time. We address both aspects of the PSA problem, static and dynamic. For the static issue, where we are interested in finding all the combinations of basic events of the system that can generate severe accidents (which could be the meltdown of the core of the nuclear power plant), we propose to obtain these combinations of basic events by a quantum algorithm, using directed graphs instead of looking for all solutions of a SAT problem. Our main contribution is a quantum algorithm, that uses a linear number of qubits and, using a classical filter, we can find all those combinations of basic events that can generate these accidents. In the dynamic issue, where we are interested in finding all the accidental sequences of a system, our main interest is the treatment of these sequences, wherein the classical case, in order to find all these sequences we use the state graph of the system and we look for all the paths between the current state and all the critical states. This problem is NP-complete, therefore we propose a quantum solution to find all these paths. We propose two quantum algorithms, both based on the philosophy of quantum walks. The first algorithm allows us to find all the paths between a source vertex and several destination vertices in a directed a-cyclic graph (the state graph of the system). This algorithm uses N qubits and M gates in order to find all the paths. The second one is a hybrid version of the first one, it is able to process large graphs even with a reduced number of qubits. Another contribution consists of a quantum approach to the Dynamic Time Warping (DTW) algorithm, in order to calculate the similarity between these sequences, as well as, a version able to find the best match between sequences by using sub-sequences with dynamically varying lengths. We propose also a learning strategy for Quantum Hidden Markov Models (QHMMs) to generate accidental scenarios from any initial state of the system and also to manage the system in real time. We propose finally a version of quantum k-means. The complexity of each iteration of the classical version of k-means is $\mathcal{O}(K \times M \times N)$. In our case, computing all the distances between observations and cluster centers with a single circuit and using Grover's quantum search algorithm allows us to reduce the complexity to $\mathcal{O}(\log(K \times M \times N))$. Another version of the quantum Balanced k-means algorithm is proposed using quantum annealing. Finally, we propose a quantum version of the convex-NMF algorithm that is faster than the classical version. We conclude this dissertation with an application of our proposed approaches to a real system in the PSA field.

Acknowledgements

First and foremost, i would like to thank my thesis director, Prof. Younes Bennani, my co-supervisor Dr. Basarab Matei and my professional supervisor Dr. Mohamed Hibti for their unwavering support throughout this doctoral journey, and for giving me the necessary tools to succeed in my scientific work.

My sincere thanks also go to Professors Cyrille Bertelle, Michel Verleysen, Stefano Guerrini, Ali YAHYAOUY, Maria MALEK and Rogovschi Nicoleta for having devoted time to analyze this work and for the very rich and instructive exchanges that we had during the defense.

I would like to express my gratitude to my colleagues of the University Sorbonne Paris Nord for their help, their advice and especially for their conviviality and their research spirit.

I would like to express my sincere thanks to all the I21 team of the PERICLES department of EDF Lab Paris Saclay and to all the people who, by their words, their writings, their advices and their criticisms, guided my reflections and accepted to meet me and to answer my questions during my research. I would also like to thank EDF RD for funding this thesis and for giving me the opportunity to get involved in this field of research.

I want to thank my wonderful parents who have always been there for me. Also, I would like to thank my sisters and brothers for their encouragement.

Finally, I would like to thank all the people who contributed to this success and who helped me during this research work.

Ahmed Zaiou

Contents

List of Figures

List of Tables

Dedicated to my parents . . .

Avant Propos

Les ordinateurs quantiques et l'informatique quantique ont récemment attiré une attention considérable dans l'industrie et dans les milieux universitaires. La commercialisation d'ordinateurs quantiques "efficaces" et le fait que les technologies de l'information basées sur les principes de la physique quantique deviennent une réalité ont poussé un certain nombre d'acteurs à explorer ce domaine et ses conséquences sur la manière dont de nombreux problèmes complexes peuvent être résolus et l'impact qu'il peut avoir sur différentes activités.

Dans le domaine des *Noisy Intermediate-Scale Quantum* (NISQ)[1], un certain nombre de résultats encourageants ont montré l'intérêt d'explorer de nombreux problèmes complexes, et différentes approches ont été spécifiquement introduites pour tirer parti de l'accélération quantique lorsque cela est possible, même avec des ordinateurs quantiques à petite échelle, en utilisant une approche hybride. Certains problèmes peuvent cependant être résolus efficacement non pas avec des machines quantiques universelles[2], mais en utilisant la simulation quantique ou le recuit quantique.

Les algorithmes d'apprentissage automatique ont prouvé leur puissance et leur utilité dans le monde à travers leur domaine d'application. Ces algorithmes se retrouvent dans presque tous les secteurs, comme la médecine, la finance, le marketing, l'environnement, la sécurité et bien d'autres domaines très intéressants.

L'un des défis les plus importants des applications d'apprentissage automatique est la complexité de leurs algorithmes. La majorité d'entre eux ont une complexité NP-hard ou NP-complet, notamment les algorithmes suivants : K-means [Alo+09], Réseau de neurones [BR93] et arbre de décision [LR76]. En outre, avec l'explosion des données dans le monde au cours des dernières années, le problème de la complexité est devenu très important, ce qui a motivé les chercheurs à se concentrer de plus en plus sur la recherche de la meilleure approche pour traiter ces énormes bases de données le plus rapidement possible. La complexité de ces algorithmes et la taille des bases de données dans le monde d'aujourd'hui font de l'apprentissage automatique l'un des domaines qui nécessitent une accélération quantique.

Le domaine des Études Probabilistes de Sûreté (EPS) est un candidat potentiel pour l'exploration de l'informatique quantique. Pour les centrales nucléaires, il s'agit d'un problème présentant de nombreuses complexités dues à différents aspects

[1] Le terme "intermediate-scale" fait référence à la taille des ordinateurs quantiques actuels, qui sont suffisamment grands pour surpasser les ordinateurs classiques dans certains problèmes appropriés. Le terme "Noisy" fait référence au fait que nous ne disposons pas encore de technologies matures pour contrôler les qubits pendant les "longs" calculs, ce qui entraîne des erreurs ou du "bruit" de petite à grande taille.

[2] Il existe plusieurs types de machines de calcul quantique, notamment le modèle de circuit quantique, parmi lesquels la machine de Turing quantique (universelle) et l'ordinateur quantique adiabatique.

: la nature des systèmes en question avec leurs dimensions techniques et socio-organisationnelles, la complexité des modèles sous-jacents aux différentes représentations du système et de sa dynamique, et en particulier sa complexité informatique qui peut empêcher d'utiliser efficacement ces modèles pour la prise de décision en temps opérationnel. En outre, la complexité de calcul est l'un des problèmes majeurs concernant ce candidat. C'est le cas aussi bien pour les approches statiques conventionnelles (traitant uniquement de la logique des événements sans aucune considération du temps ni des aspects dynamiques) que pour les approches dynamiques (considérant les phénomènes physiques, l'ordre des événements et leur chronologie).

Dans cette thèse, nous nous sommes concentrés sur la proposition de solutions en utilisant des algorithmes purement quantiques ainsi que des algorithmes d'apprentissage automatique quantiques afin de traiter des problèmes dans le domaine des EPS, à la fois pour la problématique statique et dynamique. Nous proposons des algorithmes purement quantiques pour traiter les problèmes fondamentaux du domaine EPS, avec ses deux grands aspects, statique et dynamique. Pour la problématique statique, dans lequel nous sommes intéressés à trouver des combinaisons d'événements de base du système qui peuvent générer des accidents graves, nous proposons comment obtenir ces combinaisons d'événements de base par un algorithme quantique, en utilisant des graphes dirigés au lieu de chercher toutes les solutions d'un problème SAT. En ce qui concerne la problématique dynamique, nous nous concentrons sur le traitement des scénarios de défaillance, où nous proposons une solution quantique pour trouver ces scénarios de défaillance et créer une base de données de scénarios pour chaque système. Nous proposons deux algorithmes quantiques, le premier permet de trouver tous les chemins entre un sommet source et plusieurs sommets de destinations dans un graphe orienté a-cyclique, le second est une version hybride du premier afin de traiter de grands graphes même avec un nombre réduit de qubits. Ces chemins représentent les scénarios de défaillance du système et peuvent être appelés des séquences accidentelles. Pour les traiter correctement, nous proposons notre approche quantique de l'algorithme Dynamic Time Warping (DTW) afin de calculer la similarité entre deux séquences, et nous proposons également une version capable de trouver la meilleure correspondance entre les séquences en utilisant des sous-séquences dont la taille varie dynamiquement. L'utilisation de ces deux versions quantiques de DTW nous permettent de classifier ces séquences en prenant en compte les sous-séquences dans chaque séquence avec une variation dynamique de la taille des sous-séquences. Nous proposons également une stratégie d'apprentissage des Modèles de Markov Cachés Quantiques pour générer des scénarios accidentels à partir de n'importe quel état initial du système et aussi pour detecter les séquences probables et non probables. En plus de cela, nous traitons plusieurs problèmes dans le domaine de l'apprentissage automatique quantique, nous commençons par le calcul de distance avec des circuits quantiques, et comme conséquence de ce travail, nous proposons notre version améliorée de k-means quantique. Une autre version de l'algorithme k-means équilibré quantique est proposée en utilisant le recuit quantique. Finalement, nous proposons une version quantique de l'algorithme Convexe-NMF qui est plus rapide que la version classique. Enfin, nous terminerons cette thèse par une application de nos approches proposées dans un système dans le domaine des EPS.

Guide pratique de la thèse

Cette thèse est organisée comme suit :

Chapitre 1 : Études Probabilistes de Sûreté

Les Études Probabilistes de Sûreté (EPS) est un domaine connu pour étudier la sûreté des centrales nucléaires et également pour évaluer la nature des défaillances et la performance des systèmes installés dans ces centrales. Ce premier chapitre donne d'abord quelques généralités sur ce domaine, comment il est apparu dans le monde, ainsi que son histoire. De plus, le problème général et les objectifs sont présentés, ainsi que quelques méthodes et algorithmes permettant de répondre à ces problèmes dans le monde de l'informatique classique.

Chapitre 2 : Informatique Quantique : Concepts de Base et État de l'Art

L'informatique quantique occupe aujourd'hui un grand intérêt, grâce à ses résultats en termes de complexité et d'accélération. Afin de comprendre comment leurs algorithmes fonctionnent et pourquoi ils sont plus efficaces que les algorithmes classiques, dans ce chapitre nous présentons les notions de base du calcul quantique, telles que les notions mathématiques de base de la physique quantique, les systèmes quantiques, soit à un qubit, à deux qubits ou à n qubits, et nous montrons comment nous pouvons manipuler les qubits dans chacun de ces cas. En plus de ces notions, nous présentons les classes de la complexité classique et nous ajoutons les classes de la complexité quantique. Les algorithmes quantiques généraux qui montrent une accélération quantique sont également présentés dans ce chapitre. De plus, il contient un état de l'art sur quelques algorithmes d'apprentissage automatique quantique, supervisés et non supervisés.

Chapitre 3 : Une approche quantique du séparateur de sommets pour les graphes orientés

Le principal défi dans le domaine des EPS consiste à identifier les causes des accidents qui peuvent être provoqués dans le système étudié. La recherche de ces causes, ou en d'autres termes, ces combinaisons de défaillances qui peuvent générer une conséquence inacceptable, qui peut être un accident grave et qui peut aller jusqu'à la fusion du cœur de la centrale nucléaire, est un défi très important à résoudre et il est également très nécessaire de connaître toutes les possibilités. Afin de trouver ces combinaisons d'événements de base, nous utilisons des graphes orientés et nous recherchons tous les séparateurs de sommets qui peuvent arrêter le flux entre la source et le terminal du graphe. Ainsi, au cours de ce chapitre, nous décrivons en premier lieu comment nous pouvons représenter le diagramme de fiabilité d'un système par un graphe orienté. Ensuite, nous abordons le problème du séparateurs des sommets d'un graphe orienté qui n'a pas encore été résolu par un algorithme quantique précis. Nous proposons un algorithme quantique pour résoudre ce problème et nous montrons également que cet algorithme est facilement applicable dans les cadres existants des ordinateurs quantiques et qu'il est également faisable sur le plan informatique. Pour le démontrer, nous effectuons un cas d'étude dans lequel nous utilisons un petit graphe à titre d'exemple, puis nous expliquons chaque itération de notre algorithme en indiquant les résultats trouvés dans un ordinateur quantique après chaque itération.

Chapitre 4 : Approches quantiques pour le traitement des séquences

Le traitement des séquences est un défi très populaire dans plusieurs domaines tels que : les données vidéo, audio et graphiques. Pour notre problématique en EPS, il est utilisé pour traiter des scénarios de défaillance de centrales nucléaires. Dans ce chapitre, nous abordons ce défi de manière quantique à travers trois questions : Comment trouver tous les scénarios de défaillance possibles d'un système ? Comment calculer la similarité entre eux et comment les classifier ? Comment créer un modèle génératif à partir de ces scénarios ? Afin de répondre à ces questions, nous représentons les états du système par un graphe orienté, chaque sommet de ce graphe représente un état du système et chaque arête représente une des deux actions : "réparation" ou "défaillance" d'un élément de base du système. L'état critique est représenté dans le graphe par un sommet marqué. Afin de trouver les scénarios qui peuvent être la cause pour atteindre cet état critique, nous recherchons tous les chemins entre le sommet qui représente l'état actuel et le sommet marqué (l'état critique) dans le graphe. Nous explorons le graphe en utilisant des marches quantiques, est une stratégie quantique permettant de réduit la complexité de l'exploration du graphe, et aussi ne laisse pas la chance influencer le choix de trouver ou non le sommet désiré comme dans le cas des marches aléatoires. Grâce aux marches quantiques, nous pouvons traverser les graphes de manière parallèle et aborder tous les chemins possibles en même temps. Ainsi, nous proposons notre approche basée sur les marches quantiques pour trouver tous les chemins entre un sommet source et tous les sommets marqués dans le graphe. Nous proposons également une approche hybride pour traiter les grands graphes avec le nombre de qubits disponibles dans l'ordinateur quantique utilisé. De cette façon, nous pouvons répondre au premier défi qui consiste à trouver les scénarios de défaillance d'un système. Ces deux approches que nous avons proposées sont testées et comparées à l'approche classique de marche aléatoire sur 6 graphes de taille différentes.

Pour la deuxième question, mesurer la distance entre les objets est l'une des tâches essentielles dans le domaine de l'apprentissage automatique et dans divers autres domaines du traitement des données. Lorsque les objets sont des séries temporelles, nous utilisons le Dynamic Time Warping (DTW) pour mesurer la distance entre eux. Donc, nous proposons notre approche quantique de la méthode DTW et nous proposons aussi une approche quantique pour le cas de l'utilisation des sous-séquences. Ces deux approches sont testées et comparées avec l'approche classique et aussi l'approche quantique proposée par [Fel+20].

En considérant les scénarios de défaillance comme des séquences, afin de créer un modèle génératif à partir d'un ensemble de données de séquences, nous utiliserons des Modèles de Markov Cachés Quantiques (QHMMs). Ceux-ci nous permettront d'apprendre des modèles capables de nous fournir davantage de scénarios et également de détecter les scénarios probables et non probables.

Chapitre 5 : Estimation de la distance pour le clustering basé sur les prototypes quantiques

Le calcul de la distance entre les observations et la recherche d'un élément dans une liste donnée sont les deux principales étapes de presque tous les algorithmes classiques d'apprentissage automatique. Ce sont également les étapes les plus gourmandes en mémoire et en temps pour trouver les résultats finaux. C'est pourquoi, dans ce chapitre, nous allons répondre aux questions suivantes : Comment estimer les distances entre les différentes données et les centroïdes ? Comment rechercher le

centroïde le plus proche d'une donnée ? En plus de répondre à ces deux questions, nous proposons notre version quantique de l'indice de Davies-Bouldin, et nous proposons une version améliorée de l'algorithme *k*-means quantique.

Chapitre 6 : Clustering en utilisant le recuit quantique

L'un des plus grands défis que nous avons rencontrés en combinant les tailles des systèmes traiter dans le domaine des EPS et le progrès des ordinateurs quantiques aujourd'hui est que, d'une part, nous avons des systèmes avec des très grands tailles à traiter, et d'autre part, nous n'avons que de petits ordinateurs quantiques. La question qui se pose ici est la suivante : quelles sont les orientations que nous pouvons prendre pour rendre possible l'utilisation de petits ordinateurs quantiques pour les grands systèmes ? L'une des solutions possibles consiste à diviser les graphes qui représentent le système en plusieurs graphes plus petits et à les traiter séparément. Plusieurs algorithmes d'apprentissage automatique peuvent être utilisés pour diviser ces graphes en plusieurs petits graphes. La taille des modèles que nous souhaitons traiter avec des ordinateurs quantiques dans nos problèmes est très grande, et de plus, ils sont représentés par des graphes dirigés. Le traitement de ces graphes par des algorithmes d'apprentissage automatique nécessite une représentation matricielle, habituellement nous utilisons la matrice de connexion entre les sommets du graphe, ce qui nous donne une grande matrice carrée. Pour pouvoir traiter ces matrices avec une certaine facilité, il est nécessaire d'en réduire la dimension. La deuxième question est donc la suivante : comment pouvons-nous réduire les dimensions de ces matrices ?

Pour répondre à la première question, nous n'avons dans notre cas qu'une seule contrainte, la taille des clusters trouvés ne doit pas dépasser la taille maximale du système. Cette taille peut être donnée en une seule requête à un ordinateur quantique. L'algorithme qui permet de spécifier les tailles des clusters en fonction de la taille de la base de données est l'algorithme *k*-means équilibré. Pour la seconde question, nous utilisons la version convexe de l'algorithme de factorisation de matrices non négatives (Convex-NMF). Dans ce chapitre, nous nous concentrons donc sur les deux algorithmes : *k*-means équilibré et Convex-NMF. Nous améliorons la version quantique de l'algorithme *k*-means équilibré dans le recuit quantique pour trouver des résultats optimaux. Nous proposons une version quantique du Convex-NMF, pour trouver le minimum global de la fonctionnelle $\|X - XWG\|_F^2$ pour une matrice à valeur réelle X.

Chapitre 7 : Cas d'application : Système de refroidissement des piscines de combustible

Dans ce chapitre, nous considérons un système dans le domaine des EPS, le système de refroidissement des piscines de combustible, comme une application et nous appliquons les algorithmes que nous avons proposés dans les chapitres précédents pour traiter l'analyse de la sûreté de ce système. Nous utilisons le premier algorithme pour trouver les combinaisons d'événements de base qui peuvent générer des accidents graves, puis nous utilisons notre algorithme pour trouver les scénarios de défaillance de ce système. Ces scénarios sont classifiés par *k*-NN en utilisant QDTW comme distance. Finalement, la base de données est utilisée pour apprendre un Modèle de Markov Caché Quantique afin de créer un modèle génératif des scénarios.

Introduction

Quantum computers and quantum computing attracted recently huge attention in the industry and academic sides. The commercialization of "effective" quantum computers and the fact that information technologies based on principles of quantum physics are becoming a reality pushed several actors to explore the field and its consequences on the way many complex problems can be solved and the impact that it may have on different activities.

In this *Noisy Intermediate-Scale Quantum* (NISQ) [3] era, many encouraging results showed the interest of exploring many complex problems, and different approaches were specifically introduced to take advantage of the quantum speedup when possible even with small-scale quantum computers using a hybrid approach. Some problems, however, could be solved efficiently not in universal quantum machines [4] but using quantum simulation or quantum annealing.

Machine learning algorithms have proven their power and usefulness in the world through their application domain. These algorithms can be found in almost every sector, such as medicine, finance, marketing, environment, security, and many other very interesting fields.

One of the most important challenges of machine learning applications is the complexity of their algorithms. The majority of them have NP-hard or NP-complete complexity, including the following algorithms: K-means [Alo+09], Neural Network [BR93] and decision tree algorithm [LR76]. Furthermore, with the explosion of data in the world over the past few years, the problem of complexity has become very important, which has motivated researchers to focus more and more on finding the best approaches to process these huge datasets as quickly as possible. The complexity of these algorithms and the size of the datasets in the world today make Machine Learning one of those areas that require quantum speedup.

Probabilistic Safety Assessment (PSA) is a candidate for quantum computing exploration. For nuclear power plants, it is a problem with many complex issues due to different aspects: the nature of the systems at hand with their technical and socio-organizational dimensions, the complexity of the models underlying the different representations of the system and its dynamics, and particularly its computational complexity which may prevent from efficiently using such models for decision making in operational times. Moreover, computational complexity is one of the major issues regarding this candidate. It is the case for both the conventional static approaches (dealing only with the logic of the events without any consideration of

[3] The term “intermediate-scale” refers to the size of nowadays quantum computers, they are large enough to outperform classical computers in some suitable problems (see. [Ped+19a]). The term “Noisy” refers to the fact that we still do not have mature technologies to control qubits during "long" computations, which results in small to large errors or noise.

[4] There are several types of quantum computing machines including the quantum circuit model among which, quantum Turing machine (universal), and adiabatic quantum computer.

time or any dynamic aspects) and dynamic approaches (considering the physical phenomena, the order of events, and their timing).

In this dissertation, we focused on providing solutions using pure quantum algorithms as well as quantum machine learning algorithms, to deal with problems in the PSA field, both static and dynamic problems. We propose pure quantum algorithms to deal with the fundamental problems of the PSA field, with its two main aspects, static and dynamic. For the static problem, in which we are interested in finding combinations of basic events of the system that can generate serious accidents, we propose how to obtain these combinations of basic events by a quantum algorithm, using directed graphs instead of looking for all solutions of an SAT problem. Regarding the dynamic problem, we focus on the processing of failure scenarios, where we propose a quantum solution to find these failure scenarios and create a dataset of scenarios for each system. We propose two quantum algorithms, the first one allows us to find all paths between a source vertex and several destination vertices in a Directed Acyclic Graph (DAG). The second one is a hybrid version of the first one, to deal with large graphs even with a reduced number of qubits. These paths represent the failure scenarios of the system and can be called accidental sequences. To handle them properly, we propose our quantum approach to the Dynamic Time Warping (DTW) algorithm, to compute the similarity between two sequences, and we also propose a version able to find the best match between sequences using subsequences whose size varies dynamically. The use of these two quantum versions of DTW allows us to classify these sequences by taking into account the subsequences in each sequence with a dynamic variation of the size of the subsequences. We propose a learning strategy for Quantum Hidden Markov Models to generate accidental scenarios from any initial state of the system, and also to detect probable and non-probable sequences. In addition to that, we deal with several problems in the field of quantum machine learning, we start with the computation of distances with quantum circuits, and as a consequence of this work, we propose our improved version of quantum k-means. Another version of the balanced quantum k-means algorithm is proposed using quantum annealing. Lastly, we propose a quantum version of the Convex Non-negative Matrix Factorization (Convex-NMF) algorithm that is faster than the classical version. Finally, we will end this dissertation with an application of our proposed approaches in a system in the PSA field.

Guide to the Dissertation

This dissertation is organized as follows:

Chapter 1: Probabilistic Safety Assessments

Probabilistic Safety Assessments (PSA) is a field known for studying the safety of nuclear power plants, evaluating the nature of failure and also the performance of the systems installed in them. In this first chapter, we start by giving some generalities about this field and how it appeared in the world with some history. We present the general problem, the objectives, and some methods and algorithms to solve the problems in the world of classical computer science.

Chapter 2: Quantum Computing: Basics and State of the Art

Quantum computing is taking a large space today, thanks to its results in terms

of complexity and speedup. To understand how the algorithms work and why they are more efficient than classical algorithms, in this chapter, we present the basic advice of quantum computing, such as the basic mathematical notions of quantum physics, quantum systems, either by one qubit, two qubits or n qubits and we show how we can handle qubits in each of these cases. In addition to these notions, we present the classes of classical and quantum complexity. General quantum algorithms that show quantum speedup are presented in this chapter. Also, it contains state-of-the-art machine learning algorithms, both supervised and unsupervised algorithms.

Chapter 3: A Quantum Vertex Separator Approach for Directed Graphs

The first problem in the PSA field is the identification of the causes of accidents that can be generated in the studied system. The search for these causes, or in other words, these combinations of failures that can generate an unacceptable consequence, which can be a serious accident, and it can go up to the fusion of the core of the nuclear power plant, is a very important challenge to solve, and it is very necessary to know all the possibilities. To find these combinations of basic events, we use directed graphs, and we look for all the vertex separators that can stop the flow between the source and the terminal of it. In this chapter, firstly, we describe how we can represent the reliability diagram of a system by a directed graph. After that, we address the problem of vertex separators of a directed graph, which has not yet been solved by an accurate quantum algorithm. We provide a quantum algorithm to solve this problem, and we show that it is easily applicable in the existing frameworks of quantum computing, and that is also computationally feasible. To show this, we make a study case, where we use a small graph as an example, then we explain each iteration of our algorithm by indicating the results found in the quantum computer after each iteration.

Chapter 4: Quantum Approaches for Sequences Processing

Sequence processing is a very popular challenge in several fields, such as video, audio, and graphic data. For our problem in the PSA field, it is used to process failure scenarios of nuclear power plants. In this chapter, we approach this challenge in a quantum manner through three questions: how to find all possible failure scenarios of a system? How to calculate the similarity between them and how to classify them? How to create a generative model from them? In order to answer these questions, we will represent the states of the system by a directed graph, each vertex of this graph represents a state of the system, and each edge represents one of two actions: "repair" or "failure" of a basic component of the system. The critical state is represented in the graph by a marked vertex. In order to find the scenarios that can be the cause to reach this critical state, we look for all the paths between the vertex that represents the current state in the graph and the marked vertex (the critical state). We explore the graph by using quantum walks, which is a quantum strategy that reduces the complexity of exploring the graph and doesn't let luck influence the choice of finding or not the desired vertex as in the case of Random Walks. Thanks to quantum walks, we can traverse the graphs in a parallel way, and approach all possible paths at the same time. So, we propose our quantum walk-based approach, to find all the paths between a source vertex and all the marked vertices in a Directed Acyclic Graph (DAG). Also, we propose our hybrid approach to handle large graphs with the number of qubits available in the used quantum computer. In this way, we can address the first challenge of finding the failure scenarios of a system. These two approaches are tested

and compared with the classical random walk approach on 6 graphs of different sizes.

For the second question, measuring the distance between objects is one of the essential tasks in the field of machine learning and others fields of data processing. When the objects are time series, we use Dynamic Time Warping (DTW) to measure the distance between them. So, we propose our quantum approach to the DTW method, and we propose a quantum approach for the case of using sub-sequences. These two approaches are tested and compared with the classical approach and the quantum one proposed by [Fel+20].

By considering the failure scenarios as sequences, to create a generative model from a sequence dataset, we will use Quantum Hidden Markov Models (QHMMs) to learn models that can provide us more scenarios, and also detect probable and no-probable scenarios.

Chapter 5: Distance Estimation for Quantum Prototypes Based Clustering

Distance calculation between observations and finding an element in a given list are the two main steps in almost all classical machine learning algorithms. They are also the most memory and time consuming steps to find the final results. That is why in this chapter, we will answer the questions: how to estimate the distances between the different data and centroids? How to search for the closest centroid to a given data? In addition to answering these two questions, we propose our quantum version of the Davies–Bouldin index, and we propose an improved version of the quantum K-means algorithm.

Chapter 6: Clustering using Quantum Annealing

One of the biggest challenges that we have encountered in combining system sizes in the PSA field and the advances of quantum computers today is, on the one hand, we have very large systems to deal with, and on the other hand, we only have small quantum computers. So, the question here is: what are the directions that we can take to make the use of small quantum computers for large systems feasible? One of the possible solutions is to divide the graphs that represent the system into several smaller graphs and treat them separately. Several machine learning algorithms can be used to divide these graphs into several small graphs. The size of the models that we wish to process with quantum computers in our problems is very large, and they are represented by directed graphs. The processing of these graphs by machine learning algorithms requires a matrix representation, usually, we use the adjacency matrix of the graph, which gives us a large square matrix. To be able to process these matrices with some ease, it is necessary to reduce the dimension. So the second question here, is how we can reduce the dimension of these matrices.

To answer the first question, in our case, we have only one constraint, the size of the clusters found must not exceed the maximum size of the system that can be given in a single request to a quantum computer. The algorithm that allows specifying the sizes of the clusters according to the size of the database is the Balanced K-means algorithm. For the second question, we use the convex version of the Non-negative Matrix Factorization algorithm (Convex-NMF). So, in this chapter, we focus on two algorithms: The balanced K-means clustering algorithm and Convex-NMF. We improve the quantum version of the Balanced K-means algorithm in the quantum

annealing to find the best results. We propose a quantum version of the Convex-NMF, to find the global minimum of the functional $\|X - XWG\|_F^2$ for a real valued matrix X.

Chapter 7: Application case: Fuel Pool Cooling System

In this chapter, we take a system in the PSA field, the Fuel Pool Cooling System, as an application and we apply our algorithms proposed in the previous chapters to process the safety analysis of this system. We use the first algorithm to find the combinations of basic events that can generate serious accidents. After that, we use our algorithm to find the failure scenarios of this system. These scenarios are clustered by k-NN using QDTW as a distance. Finally, these datasets are used to learn Quantum Hidden Markov Models (QHMMs) to create generative models of the scenarios.

Chapter 1

Probabilistic Safety Assessments

Probabilistic Safety Assessments (PSA) are increasingly used internationally as a means of assessing and improving the safety of nuclear and non-nuclear installations. The first PSA was performed on the nuclear units of SURRY[1] and PEACH BOTTOM[2] in the USA from 1972 to 1975. It resulted in the WASH 1400 report, also known as the RASMUSSEN1 report [BI80]. This PSA was initiated by the U.S. Atomic Energy Commission, the forerunner of the Nuclear Regulatory Commission (NRC). It was based on preliminary work done by the UKEA in England and presented in 1967 by Dr. Farmer to the International Atomic Energy Agency (IAEA).

Then, the report that came out was NUREG-1150 [Ort+91], a report of about 3000 pages. In this report, the event tree method was used to link system fault trees to the initiating events and the core meltdown criteria. The use of event trees was a response to the difficulties encountered in analyzing an installation using fault trees alone.

A German study [Ker80], similar to the RASMUSSEN report, aimed to evaluate the individual and collective risks resulting from the operation of nuclear power plants and to compare them with other natural and industrial risks. These two studies have shown the importance of Loss Of Coolant Accidents (LOCA) due to small breaks in Pressurized Water Reactors (PWR). In addition, the main lessons of the RASMUSSEN report are that the consequences of a core meltdown would remain limited (no deaths, evacuation area less than 20 ha...).

This latter report was criticized by Professor Kendall, which led the NRC to initiate a critical analysis of WASH 1400 in 1977. The result of this study is summarized in the following:

- Underestimation of the uncertainties and their propagation,
- Underestimation of common cause failures (CCFs),
- Lack of completeness of the studied accidental sequences,
- Not taking into account the recovery of human errors,
- but a recommendation for the increasing use of predictive analytics techniques in nuclear safety: there will be a before and after WASH 1400.

The TMI accident occurred in 1979[3]. The WASH 1400 had already demonstrated the importance of accidents originating from small breaches in the main primary circuit,

[1] A nuclear power plant located in Surry County, Virginia.
[2] A nuclear power plant situated on the Susquehanna River, 90 km south of Harrisburg.
[3] Partial meltdown of the reactor core, with almost no effects outside the plant site.

while the reference accident, considered the most serious, was the rupture of the Tank-Pressurizer Connection.

In 1980, the NRC launched its work on the introduction of probabilistic safety goals into US regulations. At the same time, all the US administrations concerned with nuclear safety (NRC, DOE, and EPRI) launched the development of a guide for the use of probabilistic methods for the PRA Procedures Guide or NUREG-2300 [Hic83]. This report defines the three levels of study for a PSA:

- **PSA level 1:** The objective is to identify and analyze the accident scenarios, in order to quantify the frequency of reactor core meltdown. Unlike PSA Levels 2 and 3, supporting physical studies are not explicitly incorporated into PSA Level 1, which is based on a functional representation,
- **PSA level 2:** The objective is the analysis of the physical processes related to the progression of the accident following the degradation of the core and the failure modes of the confinement, in order to quantify the frequency of release of radioactive products from the confinement,
- **PSA level 3:** The objective is the analysis of the transport of radioactive products in the environment and the socio-economic consequences, in order to quantify the safety risk. The WASH 1400 is a PSA level 3.

Any intermediate model between a PSA level 1 and a PSA level 2 is called a PSA level 1+. Practically speaking, a PSA level 1+ model complements the level 1 model in such a way as to:

- Grouping the sequences leading to the core meltdown according to their characteristics, in particular, to distinguish the sequences leading to high-pressure meltdown, and the sequences with confinement bypass and/or, determining the status of the systems related to the confinement function, or the possible severe accident management functions according to the characteristics of the releases.
- Building an interface between a PSA level 1 and a PSA level 2.

Figure 1.1 shows these three levels of analysis.

Since 1990, in France, several PSA level 1 studies have been published for each level by the operator EDF, and the technical support of the Safety Authority (IRSN); they are regularly updated as part of safety reviews after analysis of all the elements resulting from feedback, and unit operation (reliability data, analysis of incidents, latest batches of unit modifications, stabilized set of PSA procedures).

1.1 PSA Level 1

PSA Level 1 consists of an evaluation of the design and operation of the plant and focuses on the sequences that can lead to the failure of the core [Inta]. Figure 1.2 schematizes the main steps of a PSA level 1, and Figure 1.3 describes the structure of a PSA in a Boolean structure. It should be noted that these different steps are highly dependent on each other and that the process is essentially iterative.

The international consensus for modeling methods is the use:

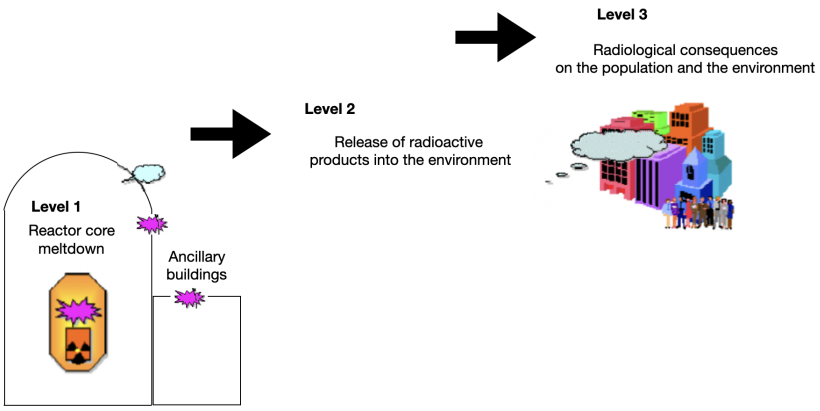


FIGURE 1.1: The three levels of PSA

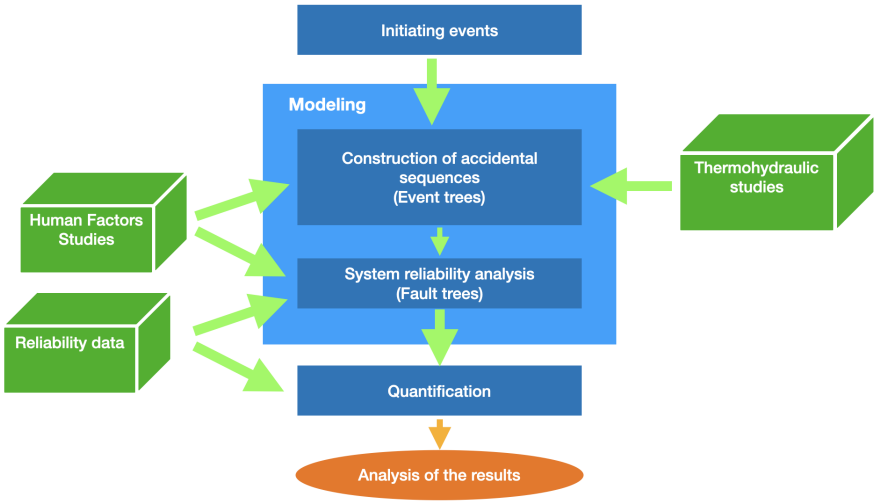


FIGURE 1.2: Steps of development of the PSA level 1

- Event Tree method, which is a logical scheme for defining accident sequences starting from an initiating event, considering the success or failure of the systems implemented to stop the progression of the accident, for modeling the sequence of scenarios from the occurrence of an initiating event, to the achievement or not of the core meltdown. This method was used for the first time for the realization of the WASH 1400: it allows to elaborate and evaluate the sequences of events leading to the core meltdown, also called accidental sequences;

- Fault Trees, which is a logical scheme allowing linking by a deductive method, the failure of the system to the elementary events likely to cause it, for the modeling of systems, unavailability, and Common Cause Failures (CCF). These fault trees are associated with events in the event trees. This method appeared in 1961-62 and is widely used in many industrial fields (chemistry, aeronautics, space, nuclear).

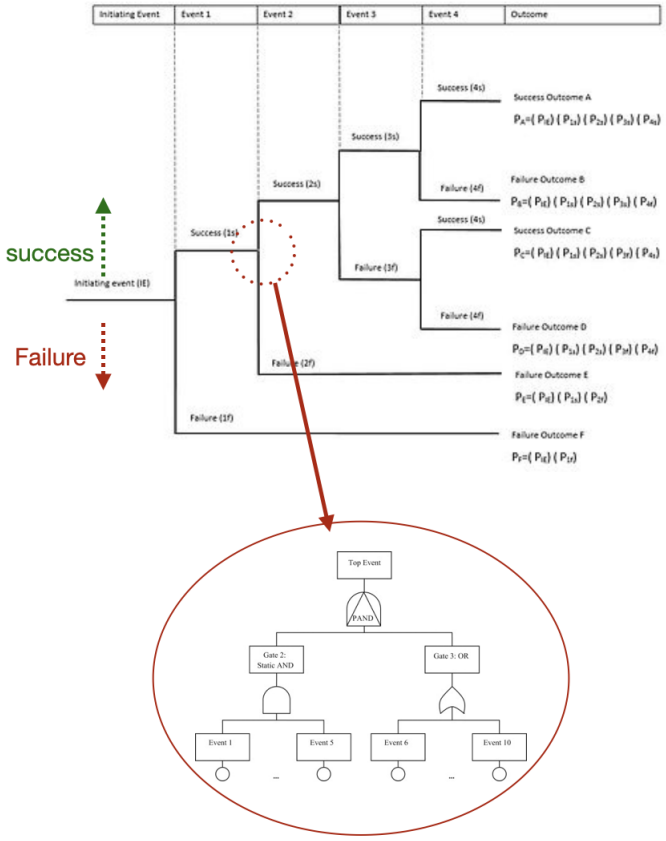


FIGURE 1.3: Structure of a PSA level 1

1.2 PSA Level 2

The objective of PSA Level 2 is to evaluate the frequency and level of releases to the environment resulting from severe accidents leading to the melting of the core [Intb].

Practically, these studies often follow a PSA Level 1 carried out beforehand (the probability of the core meltdown is therefore assumed to be known, as well as the main sequences leading to it). A preliminary work of interfacing allows us to define the different degraded states of the installation (IDE) from the results of the PSA Level 1. The IDEs associated with their frequency of occurrence constitute the input data for the PSA Level 2.

A PSA Level 2 is an analysis of the physical processes related to reactor core meltdown accidents and containment failure modes (response to induced loads on structures, transport of radioactive materials to the environment, and corium progression). This analysis is carried out from a single generic event tree for all IDEs. All the paths of this tree constitute the sequences of the PSA Level 2, for which the probability of occurrence, the inventory, and the number of radioactive materials released into the environment (or source term) are determined.

1.3 PSA Level 3

In addition to the aspects analyzed in PSA Level 2, PSA Level 3 also deals with the dispersion of radionuclides in the surrounding environment and with potential

environmental and health effects [Intb]. Once the PSA Level 2 has been carried out, a PSA Level 3 or Probabilistic Risk Assessment (PRA) must be carried out. It takes into account meteorological and population data, as well as the characteristics of agriculture around the site. Therefore, a PSA Level 3 cannot be standardized like PSA Levels 1 and 2.

The WASH 1400 went so far as to estimate the number of potential deaths. The interest of these studies lies in the evaluation of the cost (of a nuclear accident) - benefit (of the use of nuclear energy) ratio for society. There is currently no PSA level 3 studies in France. In the rest of this document, PSA refers to a PSA Level 1.

1.4 Methods of operational safety

In this section, we present some of the above-mentioned methods for analyzing operational safety. We present the Reliability Block Diagram technique, Fault Tree method, Dynamic Reliability Diagram method, Petri Net, and Markov Chains.

1.4.1 The Reliability Block Diagram technique

The Reliability Block Diagram (RBD) technique is a schematically based technique for showing how components contribute to the success and failure of an overall system. This method is widely used because of its simplicity and practicality. This model consists of having an input point, an output point, and a set of blocks connected in parallel or series. Each block represents a physical component, that functions normally. If one of the components fails, it behaves like an open switch. The diagram remains functional, if there is a path connecting the start point to the endpoint otherwise the diagram will be non-operational [Tal+09; RH03]. In Figure 1.4, we present an example of a reliability diagram.

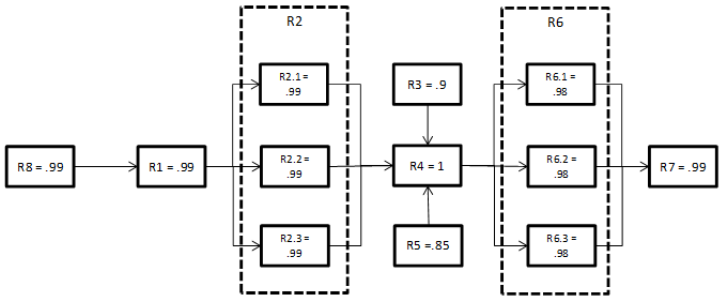


FIGURE 1.4: Reliability diagram

The main advantage of this approach is its simplicity. Engineers from a variety of professions may utilize and comprehend it with ease. However, this approach also has a significant drawback: a lack of knowledge about how the system behaves.

1.4.2 Fault Tree method

The Fault Tree (FT) method is a deductive approach (top-down), i.e. from the most general to the most detailed. It has gone through several stages since its birth until now. First, this method was implemented in 1962, the birth of this method in the Bell Telephone Company by Watson. Then, in 1965, Haasl succeeded in formalizing these

construction rules. Then Wasley started to put the basis of quantitative evaluation in the 1970s. Finally, in 1992, by coding Binary Decision Diagrams (BDD), Madre and Rauzy achieved high computational efficiency [PE10]. The goal of this method is to identify the possible combinations that cause the adverse event.

The FT is composed of three types of events: leaf events, intermediate events, and undesirable events. These events are linked together by logic gates that define the relationship and subsequently the nature of the possible failure.

This method has several advantages, such as the ease of editing because it is a graphical method that uses symbols and the ease of inserting a modification. In addition, it is very simple to understand and read the FT, even for people who are not familiar with operational safety. Also, as already mentioned, this method presents a deductive approach that allows the intuitive construction of the tree. An example of a fault tree structure is shown in Figure 1.5.

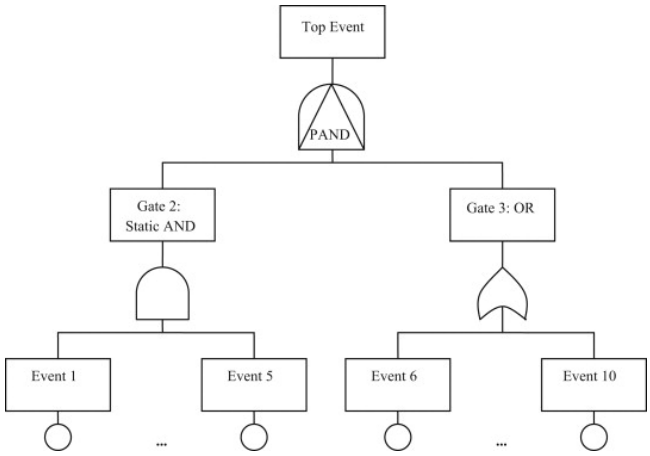


FIGURE 1.5: Fault Tree

Like any other method, the fault tree method also has some drawbacks. Indeed, FT doesn't allow the evaluation of the operational availability of repairable systems. In addition, it is necessary to redo the construction of the tree in case of system evolution.

FT allows for both qualitative and quantitative analysis. Qualitative analysis is used to identify the necessary and sufficient combination(s) of leaf events giving rise to the adverse event. The most common techniques used in qualitative fault tree analysis are Minimal Cut Set (MCS), Minimal Path Set (MPS), and Common Cause Failure (CCF). A Cut Set (CS) is a combination of component failures that cause the system to fail. An MCS is a cut set that, if one element is removed from the set, the rest is no longer a cut set. The Path Set (PS) is the opposite of CS. The PS is a combination of components that, if they do not fail, the system remains functional. An MPS is a set of paths that, if one element is removed from the set, the rest is no longer a defined path. CCF is dependent and residual failures in which two or more failure events exist at the same time due to the same cause shared. The identification of the latter helps the designer to identify the weak points of the system [RS15].

Quantitative analysis is performed to calculate the probability of the adverse event.

There are two different approaches to quantitative analysis: discrete-time quantitative analysis and continuous-time quantitative analysis. Discrete-time quantitative analysis is an approach that considers the entire life of the system as a singular event. In other words, each component can fail only once in a fixed time. In contrast, quantitative continuous-time analysis is an approach that considers the evolution of system failures over time. It is generally characterized by a probability function.

The fault tree can also be transformed into a success tree. The latter shows how to prevent the undesired event from occurring. The conditions applied to the success tree guarantee that the undesirable event does not occur. Therefore, the success tree is a valuable tool that gives equivalent information to the fault tree but from a success point of view.

1.4.3 Dynamic Reliability Block Diagram method

Given the limitations of RBD in analyzing dynamic and complex systems, the Dynamic Reliability Block Diagram (DRBD) method, which is the derivative of RBD, allows the resolution of these systems. Indeed, a DRBD is obtained by extending RBD with new tools that allow dynamic modeling and behavioral dependence between components [XXR09].

1.4.4 Petri Net

In the presence of failures, the Petri net [Mur89] allows the dynamic modeling of repairable systems. It is mainly developed for the study of dynamic systems. These systems go from one state to another for each transition of the components (repair, failure). The Petri net is a bi-part graph that contains circles and rectangles. The circles represent the states and the rectangles represent the transitions. The states are connected by two arcs and a transition. The Petri net allows the search for non-reachable states, blockages, loops, causes of expectations, and conflicts. It allows the sequential analysis of a system. This network is easier to master than other methods. But, for a very large number of states (thousands), the network will be complex and the generation of a Markov graph is almost impossible.

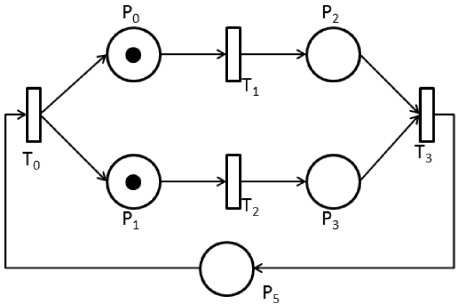


FIGURE 1.6: Petri Net

1.4.5 Markov chains

Markov chains are widely used in the literature to model the dysfunctional behavior of systems and are recommended by several industrial standards. For this type of modeling, dysfunctional behavior is seen as a stochastic process verifying the Markov

property. This hypothesis is valid for many cases of non-systematic material failures and is translated, among other things, by not taking into account the aging phenomena. The hypothesis seems more difficult to justify for repairs, but it is nevertheless accepted for the sake of the homogeneity of the models, and the ability to carry out numerical calculations.

Despite this assumption, the Markov chain remains a tool allowing behavioral modeling of incomparable precision compared to the formalism of the first category. Moreover, it is defined in a mathematical framework that allows the development of efficient analysis techniques. Unfortunately, the representation of each state being explicit, this type of modeling is subject to the classical problem of combinatorial explosion. The manual construction of such models for large systems is impossible. Therefore, Markov chains are either built manually to describe very specific local behaviors or generated automatically from another model, to facilitate quantitative analysis.

A Markov chain modeling the dysfunctional behavior of two components A and B, which can only be repaired in the order of their failures, is shown in Figure 1.7.

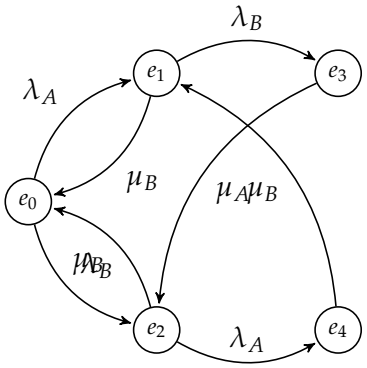


FIGURE 1.7: Example of a Markov chain for two components

1.5 Issues and motivations

In France, since 1990, EDF has allowed a systematic investigation and division of the undesirable event (also called a critical event) into several initiating events (which may cause the undesirable event). Then, for each of these initiators, the PSA identifies the accidental sequences that go through the failure or success of the backup missions put in place to reduce the risk and bring the installation back to a safe state called acceptable consequence (AC). By studying the nuclear installation as an integrated system, including both technical and human aspects, PSA contributes to risk management, identifies accident sequences, determines how often they may occur, and studies for each scenario the set of potential consequences.

1.5.1 Construction of a PSA

The first step in the construction of a PSA is the identification of the list of initiating events that can cause the critical event, which is the core meltdown. An initiator can be an internal accident (explosion, fire, etc), or an external accident (flood, tornado, etc) affecting the facility site.

The second step in a PSA is to perform a Qualitative Sequence Analysis (QSA), which models all possible scenario sequences based on the evolution of the facility's response to the initiator's effects. Each scenario begins with the initiator under study and runs through the successes and failures of the safety systems, and human actions put in place to minimize the effects of the initiator.

These accidental scenarios are then translated into event trees (one tree for each initiator, with potential references to other initiators). The failures of system missions are studied through the analysis of fault trees, while those of human actions are point values injected in the PSA model, but which come from the results of human reliability analyses. Regarding the sequences that lead to unacceptable consequences, their frequencies are calculated and the cut lists are established using the RiskSpectrum software.

Note that the fault tree analysis allows us to model the causes of the failure of a system in detail. This type of analysis is used to identify the potential causes of failure of a system through a cascade of logical gates (OR, AND,..) of component failures. It also allows us to calculate the probability of failure of each mission because a fault tree can be seen as a Boolean expression linking the undesirable event to the basic events. It has been demonstrated that any Boolean expression admits a unique representation (canonical form) in the form of a union of minimal cuts. The minimal cuts are the conjunction of several basic events. Thus, the probability of each minimal cut is calculated, then, the probability of the undesirable event is estimated from the probabilities obtained by the application of the formula of Poincaré according to the following formula:

$$P(UE) = \sum_{i=0}^{n} P(C_i) - \sum_{i<j} P(C_i, C_j) + \sum_{i<j<k} P(C_i, C_j, C_k) - \cdots + (-1)^n P(C_1, C_2, \ldots, C_n) \tag{1.1}$$

such that UE is the Undesired Event, and C_i are the minimum cuts.

In practice, the cross basic probabilities are often low, so we are satisfied with the first term of the Poincaré formula:

$$P(UE) = \sum_{i=0}^{n} P(C_i) \tag{1.2}$$

1.5.2 Motivation

In the industry, the running time or the time needed to obtain the results plays a very important role, it can prevent the delay of a very important decision, such as the shutdown of the electricity production in nuclear power plants. For this reason, finding methods or technologies to respond quickly and precisely to our problem, is a field of research still open to improving the running time situation. Today, although being a powerful tool for calculation and risk analysis, PSA has several limitations.

In the static part, PSA calculations are based on boolean formulas, which must be reduced to a disjunctive normal form. This type of calculations are known to be NP-hard [BW96] [FS87]. To overcome these limitations, several truncations are performed to obtain results. A truncation can establish a probability threshold or

a cut length threshold. Then, it eliminates the cuts that are below the chosen threshold.

In the dynamic case, it is a question of exploring accidental sequences (with dynamic behaviors linked to physical phenomena and kinetic aspects linked to driving sequences, system recovery, and the order of solicitation of backup systems) which are similar to the paths of very large graphs that can include circuits and make the probabilistic evaluation of undesirable events very complicated.

On the other hand, quantum Computing [Jae07] is becoming increasingly a solution for complex problems that are hard to solve with classical computers [Pre12]. This computational ability has been demonstrated theoretically by several papers, which suggests very interesting algorithms such as Grover's algorithm [Gro96], Shor algorithm [Sho99a] and several others such as [Ped+19b; Zho+20]. In this dissertation, we look at our problem in the context of quantum computing to propose solutions that can be efficient in terms of computation time and computational complexity.

1.6 Conclusion

Some of the probabilistic safety assessment principles employed in this dissertation are presented in this chapter. Several justifications for using quantum computing to address issues in the PSA sector.

Chapter 2

Quantum Computing: Basics and State of the Art

Quantum computing [NC02] has attracted enormous interest in recent years, and attracted many researchers in different disciplines. This excitement followed the two main revolutionary algorithms introduced by Grover and Shor. The first algorithm introduced by Grover [Gro96], manages to reduce the complexity of finding an element in an unstructured dataset of size N to $O(\sqrt{N})$, the second by Shor [Sho99a], which can break the RSA code in a polynomial time. One of the main goals of this new research area is to solve problems that can't be solved in the classical framework to break the computational complexity of many hard problems, and sometimes find good shortcuts and new approaches to solving them. In 2012, John Preskill introduced the term "quantum supremacy", a concept to describe that quantum computers can do some things that classical computers can't [Pre12]. Various algorithms were then proposed to achieve this goal (sycamore [Ped+19b], Chinese [Zho+20]). Others have shown significant acceleration compared to the best results of classical computers. In this chapter, we describe in detail what a quantum system is according to the mathematical notions, what is a system of one qubit, two qubits, and n qubits, and how we manipulate them in each case. Then, we discuss what is a quantum circuit, we explain what is the quantum annealing computation with the QUBO (Quadratic Unconstrained Binary Optimization) problem, as well as how to solve an optimization problem through the D-wave system. After that, we talk about the computational complexity in the quantum case. Finally, We give a state-of-the-art on quantum algorithms and then describe some quantum algorithms in the field of machine learning.

2.1 Mathematical View of a Quantum System

The quantum state of a quantum system is completely described by a wave function associated with a state vector (we call it by ket ψ) $|\psi\rangle$, which is described by a unitary column vector. The latter belongs to a complex space having a scalar product, called state space. This is a Hilbert space which we will denote **H** and describe as follows.

Definition 1 A Hilbert space is a Banach space having the norm $||.||$ deriving from a Scalar or Hermitian product $\langle .,. \rangle$ according to the formula $||\omega|| = \sqrt{\langle \omega, \omega \rangle}$. It is the generalization into any dimension of a Euclidean or Hermitian space. Following the theorem of M. Fr'echet, J. von Neumann and P. Jordan, a Banach space (respectively named vector space) is a Hilbert space (respectively pre-Hilbert space) if and only if its norm checks the equality:

$$\|\omega_1 + \omega_2\|^2 + \|\omega_1 - \omega_2\|^2 = 2(\|\omega_1\|^2 + \|\omega_2\|^2) \tag{2.1}$$

This means the satisfaction of the rule of the parallelogram, the sum of the squares of the sides of a parallelogram is equal to the sum of the squares of the diagonals.

The evolution of a quantum system is described by a unitary transformation. Let $|\psi\rangle$ be the state of a system at time t_1 and $|\psi\rangle'$ its state at t_2, then there exists a unitary transformation **U** such that:

$$|\psi\rangle' = \mathbf{U}\,|\psi\rangle \tag{2.2}$$

where **U** is a function that allows to change of the state of the system at time t_1 to the state of the system t_2. **U** is Hermitian if all its eigenvalues are real and, if it is Hermitian, it is measurable and it is observable. Otherwise, only the name of the operator is appropriate.

The evolution of the operator associated with a quantum system is governed by the Schrodinger equation:

$$i\hbar\frac{d\,|\psi\rangle}{dt} = \mathbf{H}\,|\psi\rangle \tag{2.3}$$

the operator **H** here is the Hamiltonian of the system, it is the observable associated with the total energy of the system. Moreover, $\hbar = \frac{h}{2\pi}$ where h is Planck's constant. We consider the eigenvalue equation of an observable O:

$$O\,|\psi_n\rangle = \lambda_n\,|\psi_n\rangle \tag{2.4}$$

with $|\psi_n\rangle$ as complete base. We can then define $|\psi\rangle$ on this base:

$$|\psi\rangle = \sum_n c_n\,|\psi_n\rangle \tag{2.5}$$

The probability of measuring λ_n is $|c_n|^2$. The result is random: $\lambda_1, \lambda_2, \lambda_3$, ... except in the case where the state $|\psi\rangle$ is of the form $|\psi_n\rangle$ (in which case, $|c_n|^2 = 1$ and the measurement gives the value λ_n).

The space of states of a composed physical system is the direct product of the spaces of states associated with the subsystems that compose it. Technically, if a subsystem i is represented by $|\psi_i\rangle$, and the total system is constituted by n kets numbered from 1 to n, then the state of the system is:

$$|\psi_1\rangle \otimes \cdots \otimes |\psi_i\rangle \otimes \cdots \otimes |\psi_n\rangle \tag{2.6}$$

2.2 One qubit system

Before starting to explain how this qubit is used in quantum computing, we start by addressing the question: what is a qubit?

2.2.1 Qubit

Let's begin with the word qubit, this term came after the combination of the two words Bit and Quantum, also it can be called Quantum bit, the qubit represents the quantum analog of the classical bit. It is the " computer description " of the state of a particle, an elementary quantum system. In contrast to the classical bit that can take only one value at a time, the qubit can exist in a superposition of states.

2.2.2 Superposition

The principle of superposition in quantum mechanics specifies that a quantum state can have several values for a certain observable quantity. This is because the state - any state - of a quantum system is described by a vector in a space called Hilbert space. This vector, like any vector of any vector space, can be decomposed into a linear combination of other vectors according to a given basis. In quantum mechanics, a given observable corresponds to a given basis of the Hilbert space. Therefore, if we are interested in the position of a particle, the positional state must be described as a sum of a finite number of vectors, each vector representing a specific position in space. The norm of each of these vectors represents the probability that the particle is present at a given position.

Example 1 Example An electron can simultaneously take two orbits of an atom.

Using Dirac notation, the concept of Qubit has been formally introduced in the context of quantum information transmission theory as a two-level system, whose state can be written as a superposition of two fundamental states: $|0\rangle$ and $|1\rangle$.

$$|\psi\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \tag{2.7}$$

Here β_0 and β_1 are two complex numbers $(\beta_0, \beta_1) \in \mathbb{C}^2$, called the amplitudes of the two classical states $|0\rangle$ and $|1\rangle$. Where the following normalization condition is verified:

$$|\beta_0|^2 + |\beta_1|^2 = 1 \tag{2.8}$$

When we measure the state $|\psi\rangle$, we observe $|0\rangle$ or $|1\rangle$, with the two probabilities β_0 and β_1 respectively. Moreover, the measurements are not reversible: the state of the system becomes $|0\rangle$ or $|1\rangle$ and it is impossible to return to the values of β_0 and β_1.

2.2.3 Handling a qubit

Before visualizing the Bloch sphere, it is necessary to introduce some formality. We consider a Hilbert space **H** of dimension 2 with an orthogonal basis $\{|0\rangle, |1\rangle\}$. This basis must be orthogonal in order to distinguish the states during the measurement. The base is also normalized because the kets are normalized. It is obvious, from the dimension of **H**, that the operations that will be performed on its elements are modeled in 2×2 matrices.

Let's assume that we have a generic operator:

$$\mathbf{U} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{2.9}$$

and we have two elements of the space **H** represented as follows:

$$|\psi\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle \tag{2.10}$$

and

$$|\psi\rangle' = \beta_0' |0\rangle + \beta_1' |1\rangle \tag{2.11}$$

with the effect of the operator $\mathbf{U}$ on the qubit $|\psi\rangle$ produces as a result the state $|\psi\rangle'$ as follows:

$$|\psi\rangle' = \mathbf{U}\,|\psi\rangle \tag{2.12}$$

which can be calculated as a matrix multiplication as follows:

$$\begin{pmatrix}\beta_0'\\ \beta_1'\end{pmatrix} = \begin{pmatrix}a & b\\ c & d\end{pmatrix}\begin{pmatrix}\beta_0\\ \beta_1\end{pmatrix} \tag{2.13}$$

$$= \begin{pmatrix}a\beta_0 + b\beta_1\\ c\beta_0 + d\beta_1\end{pmatrix} \tag{2.14}$$

According to the fact that the sum of the probabilities (amplitude normalization condition) must always be equal to 1, we must have:

$$|\beta_0|^2 + |\beta_1|^2 = |\beta_0'|^2 + |\beta_1'|^2 = 1 \tag{2.15}$$

This is the only condition to be satisfied and it corresponds to a unitary $\mathbf{U}$, that is, by definition, such that:

$$\mathbf{U}^\dagger\mathbf{U} = \mathbf{U}\mathbf{U}^\dagger = I_2 \tag{2.16}$$

where I_2 is the 2×2 unit matrix and $\mathbf{U}^\dagger$ is the conjugate transpose of $\mathbf{U}$. This condition implies that any unitary transformation is reversible, so any calculation, in the absence of measurements, is reversible. One consequence is that the determinant of $\mathbf{U}$ is equal to $e^{i\alpha_0}$ where $\alpha_0 \in \mathbb{R}$, which greatly simplifies the set of transformations to be considered.

In order to describe the unitary transformation that can be applied to a qubit, we start by representing the qubit using the Bloch sphere. The angles θ and ϕ correspond to the polar spherical coordinates of $|\psi\rangle$.

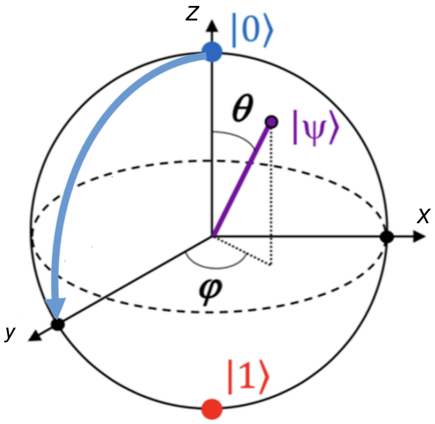


FIGURE 2.1: Bloch sphere

In this representation, we can establish the basic rotations around the X, Y, and Z axes using Pauli operators (We also say Pauli gates), which are defined as follows.

Pauli-X (X)

For the rotation around the X axis, we use the operator: $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

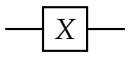


FIGURE 2.2: X gate

Pauli-Y (Y)

For the rotation around the Y axis, we use the operator: $\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$

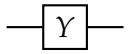


FIGURE 2.3: Y gate

Pauli-Z (Z)

For the rotation around the Z axis, we use the operator: $\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

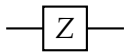


FIGURE 2.4: Z gate

also those operators can be named by X, Y, and Z respectively.

General quantum gate $U(\theta, \phi, \lambda)$

The most general quantum gate is represented by the following matrix:

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2) \end{pmatrix} \tag{2.17}$$

with the three parameters θ, ϕ and λ, we can make any relation of $|\psi\rangle$ with respect to X, Y, and Z axis.

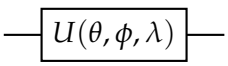


FIGURE 2.5: $U(\theta, \phi, \lambda)$ gate

The identity is defined by the operator: $I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

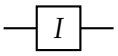


FIGURE 2.6: I_2 gate

Hadamard gate

The most important example of these unitary transformations is the Hadamard gate because it is the way to exploit the quantum superposition. It is defined as follows:

$$H = \frac{1}{\sqrt{2}}(\sigma_x + \sigma_z) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.18}$$

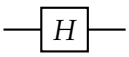


FIGURE 2.7: H gate

This gate allows to take a known basic quantum state, $|0\rangle$ or $|1\rangle$, in a superposition of states $\beta_0 |0\rangle + \beta_1 |1\rangle$ or $\beta_0 |0\rangle - \beta_1 |1\rangle$. It also allows if we apply it twice ($H^2 = I_2$), to keep any state (because it is a unitary gate).

2.3 Two qubits system

The switch to a two-qubit system implies that we are using two qubits to build our problem or perform our calculation. Following this passage, we simply start by asking the following questions:

- What states can be handled by a two-qubit system?
- How does the connection between these two qubits work?
- How we can apply transformations on these two qubits?

A fundamental property, purely quantum, appears under certain conditions as soon as we consider a system with 2 qubits: entanglement.

2.3.1 Entanglement

The entanglement is a quantum phenomenon in which two particles or more share the same properties. When one of them is measured, the other entangled particles instantly take the same value, independently of the distance between them. Mathematically, it allows us to define the following formalism, let be two distinct Hilbert spaces $\mathbf{H}_1$ and $\mathbf{H}_2$ with 1 qubit, provided with two orthogonal bases:

$$B_{\mathbf{H}_1} = \{|0_{\mathbf{H}_1}\rangle, |1_{\mathbf{H}_1}\rangle\} \text{ and } B_{\mathbf{H}_2} = \{|0_{\mathbf{H}_2}\rangle, |1_{\mathbf{H}_2}\rangle\} \tag{2.19}$$

The total system then consists of the following base:

$$\{|0_{\mathbf{H}_1}\rangle \otimes |0_{\mathbf{H}_2}\rangle, |0_{\mathbf{H}_1}\rangle \otimes |1_{\mathbf{H}_2}\rangle, |1_{\mathbf{H}_1}\rangle \otimes |0_{\mathbf{H}_2}\rangle, |1_{\mathbf{H}_1}\rangle \otimes |1_{\mathbf{H}_2}\rangle\} \tag{2.20}$$

by reducing the notations we find $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$.

In order to create the matrices that act on two qubits, we must first define the tensor product.

Definition 1 Let A and B be two matrices where:

$$A = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}, \quad B = \begin{pmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{pmatrix} \tag{2.21}$$

The tensor product of A and B is defined as follows:

$$A \otimes B = \begin{pmatrix} a_{00}B & a_{01}B \\ a_{10}B & a_{11}B \end{pmatrix} \tag{2.22a}$$

$$= \begin{pmatrix} a_{00}b_{00} & a_{00}b_{01} & a_{00}b_{00} & a_{01}b_{01} \\ a_{00}b_{10} & a_{10}b_{11} & a_{00}b_{10} & a_{01}b_{11} \\ a_{00}b_{00} & a_{01}b_{01} & a_{00}b_{00} & a_{01}a_{01} \\ a_{10}a_{10} & a_{11}a_{11} & a_{00}a_{10} & a_{01}a_{11} \end{pmatrix} \tag{2.22b}$$

In the same way, for the vectors $|\psi_1\rangle = \beta_1 |0\rangle + \beta_2 |1\rangle$ and $|\psi_2\rangle = \beta'_1 |0\rangle + \beta'_2 |1\rangle$ the tensor product is defined as follows:

$$|\psi_1\psi_2\rangle = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \otimes \begin{pmatrix} \beta'_1 \\ \beta'_2 \end{pmatrix} = \begin{pmatrix} \beta_1\beta'_1 \\ \beta_1\beta'_2 \\ \beta_2\beta'_1 \\ \beta_2\beta'_2 \end{pmatrix} \tag{2.23}$$

A priory, we could decompose a very large complex quantum system into tensor products of smaller and simpler subsystems. However, this is far from always being the case, and this is as soon as we consider two qubits: let $|\psi_{\mathbf{H}_1}\rangle$ and $|\psi_{\mathbf{H}_2}\rangle$ be two qubits, a quantum state with two qubits is said to be entangled if it is not of the form $|\psi_{\mathbf{H}_1}\rangle \otimes |\psi_{\mathbf{H}_2}\rangle$. The most famous example is the "Bell states", which are defined as follows:

$$|B_0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{2.24}$$

$$|B_1\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{2.25}$$

$$|B_2\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \tag{2.26}$$

$$|B_3\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \tag{2.27}$$

This means that the 2 qubits are maximally correlated, maximally correlated meaning that the correlation does not depend on anything. In the classical case, when two values are correlated, the correlation function associated with them depends on various parameters, for example, some decrease when the distance between two variables increases. On the other hand, in the quantum case, the correlation does not depend on any parameter. This property is very surprising because if we make a measurement in $\mathbf{H}_1$, and we find for example the value 0, everything will happen as if we colleagues of $\mathbf{H}_2$ knowing that we took this value to switch instantly in the same state, and this even if it is millions of light years away!

2.3.2 Handling two qubits

Now, let's look properly at the possible manipulations on 2 qubits. Even if, in general, any unitary matrix of dimension 4×4 can be a unitary transformation on two qubits, unitary operations on 2 qubits most often follow from the following question: **if c then U** . Indeed, as this one revealed great importance in classical logic, the idea came to define a quantum counterpart, known under the name **Controlled-U** and noted **C-U**. Its action is the following: we consider a control qubit, $|c\rangle$, and a target qubit, $|\psi\rangle$. If the control qubit is at 0, (i.e. in the state $|0\rangle$), the target qubit keeps its state; otherwise ($|1\rangle$), the operation U is applied to the target qubit.

C-U gate

The **C-U** gate allows to operate the U gate on the second qubit if the first qubit is equal to $|1\rangle$, otherwise, the second qubit is left unchanged. The matrix of this gate is represented as follows:

$$\textbf{C-U} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix} \tag{2.28}$$

And for a state $|\psi\rangle = \beta_1 |00\rangle + \beta_2 |01\rangle + \beta_3 |10\rangle + \beta_4 |11\rangle$, the **C-U** transformation is:

$$\textbf{C-U}(|\psi\rangle) = \beta_1 |00\rangle + \beta_2 |01\rangle + \beta_3 |1\rangle \otimes U |0\rangle + \beta_4 |1\rangle \otimes U |1\rangle \tag{2.29}$$

The gate U here can be changed by any Pauli gate. Graphically, we show the gate **C-U** as follows:

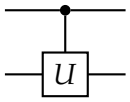


FIGURE 2.8: **C-U** gate

C-Not gate (C-X gate)

The **C-Not** gate (Controlled Not, knowing that it can be noted **C-X**) allows to operate a Not gate (**X** gate) on the second qubit, if the first qubit is equal to $|1\rangle$, otherwise, the second qubit is left unchanged. The first qubit is usually called the control qubit. The second one is usually called target qubit. This quantum gate is used in particular to realize the quantum entanglement between the control and the target qubits. The matrix of this gate is represented as follows:

$$\textbf{C-Not} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.30}$$

And for a state $|\psi\rangle = \beta_1 |00\rangle + \beta_2 |01\rangle + \beta_3 |10\rangle + \beta_4 |11\rangle$, the **C-Not** transformation is:

$$\textbf{C-Not}(|\psi\rangle) = \beta_1 |00\rangle + \beta_2 |01\rangle + \beta_3 |11\rangle + \beta_4 |10\rangle \tag{2.31}$$

Graphically, we show the gate **C-Not** as follows:

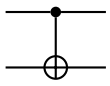


FIGURE 2.9: **C-Not** gate

SWAP gate

The **SWAP** gate, is the gate that allows to swap of the states of two qubits. This gate is represented graphically in Figure 2.10 and mathematically with the following matrix:

$$\mathbf{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.32}$$

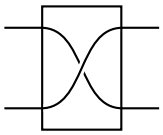


FIGURE 2.10: **SWAP** gate

It can be built also with three **C-Not** gates as in Figure 2.11.

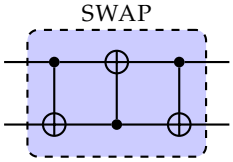


FIGURE 2.11: **SWAP** gate with three **C-Not**

For a state $|\psi\rangle = \beta_1 |00\rangle + \beta_2 |01\rangle + \beta_3 |10\rangle + \beta_4 |11\rangle$, the **SWAP** transformation is:

$$\mathbf{SWAP}(|\psi\rangle) = \beta_1 |00\rangle + \beta_2 |10\rangle + \beta_3 |01\rangle + \beta_4 |11\rangle \tag{2.33}$$

2.4 n qubits system

n qubits system is a system where we use n qubits entangled between them. Mathematically, let us suppose that we have n Hilbert spaces $\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_n$, of one qubit for each one, each $\mathbf{H}_i$ provided by the orthogonal base:

$$B_{\mathbf{H}_i} = \{|0_{\mathbf{H}_i}\rangle, |1_{\mathbf{H}_i}\rangle\} \tag{2.34}$$

the total system of these bases is a base of 2^n different states.

2.4.1 Quantum computing with oracles

A quantum oracle or we can sometimes call it a black box, is a unitary transformation that represents a boolean function. Suppose that we have a boolean function, $f : \{0,1\}^n \leftarrow \{0,1\}^m$, with n and m are the numbers of inputs and outputs respectively. We can describe this function by using a quantum oracle O, where the oracle O acts on the input as follows:

$$O(|x\rangle \otimes |y\rangle) = |x\rangle \otimes |y \oplus f(x)\rangle \tag{2.35}$$

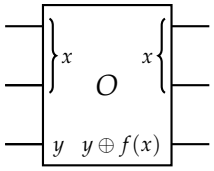


FIGURE 2.12: The general structure of the Oracle O

2.5 Quantum circuit

A quantum circuit is a computational routine consisting of coherent quantum operations on quantum states, combined with real-time classical computation. It is an ordered sequence of quantum gates, measurements, and resets, all of which can be conditioned by data from classical computation in real time.

A set of quantum gates is said to be universal if any unitary transformation of quantum data can be efficiently approximated arbitrarily as a sequence of gates in the set. Any quantum program can be represented by a sequence of quantum circuits and non-concurrent classical calculations.

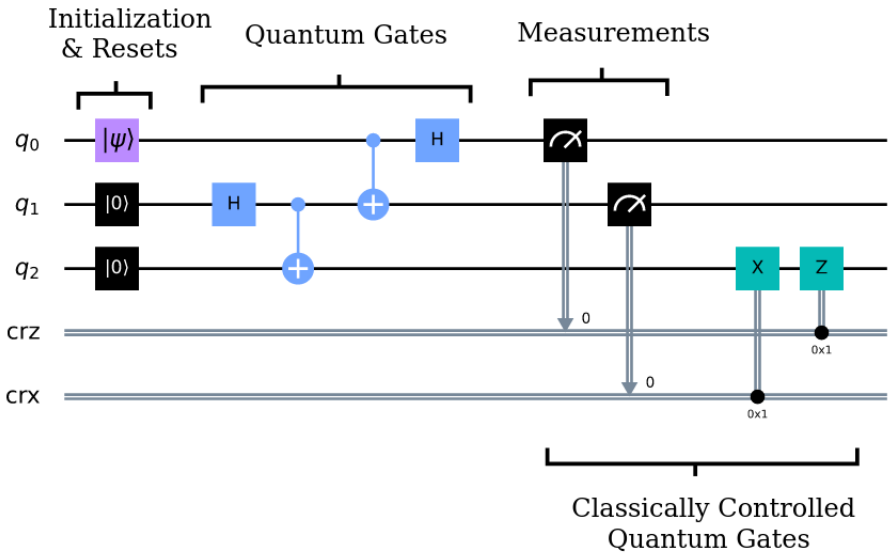


FIGURE 2.13: Example of a quantum circuit composed of three qubits and two classical bits.

There are four main components in the quantum circuit 2.13:

- **Initialization and reset:** we need to start our quantum computation with a well-defined quantum state. This is achieved using the initialization and reset operations. The resets can be performed by a combination of single-qubit gates, and concurrent real-time classical computation that monitors whether we have successfully created the desired state through measurements. The initialization of q0 into the desired state $|\phi\rangle$ can then follow by applying single-qubit gates.
- **Quantum gates:** we apply a sequence of quantum gates that manipulate the three qubits according to the needs of the objective algorithm. In this case, we only need to apply single-qubit Hadamard (H) and two-qubit C-Not gates.
- **Measurements:** we measure two of the three qubits. A classical computer interprets the measurements of each qubit as classical outcomes (0 and 1) and stores them in the two classical bits.

2.6 Quantum annealing

The quantum annealing is used to find the global minimum of an objective function. It has been proposed for the first time by B. Apolloniet all in [ADFCB88] [ACDF89] and was later reformulated in [KN98][Fin+94]. It uses quantum physics to search for low energy states, in order to find the optimal combination that solves the original optimization problem. It uses quantum bits (Qubits), that can be in state $|0\rangle$, or state $|1\rangle$, or both states at the same time, which gives a new strategy for representing and processing data. This quantum physics strategy provides sufficient power and speedup to process very complex optimization problems.

The company D-wave announced in 2012 the launch of the first computer for quantum annealing with 128 qubits. This adiabatic quantum computer prepares a Hamiltonian, in other words, prepares a quantum system with several connected qubits. At the beginning of the treatment, these qubits are superposed. Then, the computer will make this Hamiltonian evolve in an adiabatic way in order to find the resolution of the problem. Currently, we can go up to 2000 qubits, thanks to the D-wave 2000Q quantum computer. The latter is made available by D-wave in open source, it contains the necessary tools for quantum annealing and allows to solve the Quadratic Unconstrained Binary Optimization problem (QUBO) in a hybrid way on quantum processors and classical hardware architectures with the software Qbsolv.

The D-Wave quantum annealer handles QUBO problems natively [McG14]. It starts with a set of superposed qubits, each qubit having the same probability of $|0\rangle$ and $|1\rangle$ states. After a few microseconds, we find the classical states, which correspond to the minimum energy of the problem, or a state very close to it.

In order to use this computer, we simply need to transfer the problem to a QUBO and perform the embedding to give the problem as input to the D-wave 2000Q in order to find the global minimum.

2.6.1 QUBO problems

The Quadratic Unconstrained Binary Optimization (QUBO) problem became a very generic model, which can be used to represent a large range of combinatorial optimization problems. It's an NP-complete problem and can be solved by a quantum computer after being embedded in the chimera graph of the quantum computer.

The generic QUBO problem has the following form:

$$\sum_{b} \psi(b) q_b + \sum_{b<b'} \psi'(b,b') q_b q_{b'} \tag{2.36}$$

with $\psi(b) \in \mathbb{R}$ are the linear coefficients, $\psi'(b,b') \in \mathbb{R}$ are the quadratic coefficients of the problem and $q_b, q_{b'} \in \mathbb{B}$ for all $i,j \in [0,n]$ where $\mathbb{B} = \{0,1\}$ and $0 \leqslant j \leqslant i \leqslant n$. n is the number of binary variables of the problem.

The problem can be formulated using matrix notation as follows:

$$\min_{q \in \mathbb{B}^n} q^T \Psi q \tag{2.37}$$

with $\Psi \in \mathbb{R}^{n \times n}$ is a symmetric $n \times n$ matrix containing the coefficients $\psi(b)$ and $\psi'(b,b')$, and q it's a binary vector.

2.6.2 D-Wave Systems

D-Wave Systems is a Canadian private company founded in 1999. Its main activity is the creation and distribution of quantum computers. Indeed, it is known for being the first company in the world that sold quantum computers. As mentioned before, D-Wave Systems follows the Quantum Annealing approach to design its supercomputers. Their two last supercomputers are: "D-Wave 2000Q", which was released in January 2017 with 2048 qubits in a Chimera architecture; and "Advantage", released in 2020 with 5640 qubits in a Pegasus architecture [Zbi+20].

1. **Ocean Software**

   D-Wave Systems(D-Wave for short) has its own APIs[1], IDE[2], and software tools in general to use their computers. The documentation is constantly evolving according to the new functionalities that are introduced. D-Wave offers an online service called Leap that enables users to connect to D-Wave supercomputers remotely and run their own programs on them. Each user has at least 1 minute per month of quantum computation time for free, which has been more than enough for the development of this research project. Although all the implementation of the code can be done with online tools, D-Wave also offers the possibility to download their Git Hub repository, and all the software necessary to use the tools locally with a connection to Leap service. Ocean software includes multiple solvers and samplers as well as connection protocols for D-Wave supercomputers. This makes it possible to model a QUBO problem in this case and run it on their supercomputers easily using the different solvers.

2. **Steps of a problem solving**

[1] Application Programming Interface
[2] Integrated Development Environment

In order to get a general idea about how a problem is solved with D-Wave's quantum supercomputers, let's explain the steps that are followed: First, the mathematical problem has to be modeled into a QUBO. It is usually modeled into a QUBO instead of an Ising because it is easier and more intuitive for the user. QUBO is more orientated to computer science problems, while Ising is traditionally used in statistical mechanics. Secondly, either manually by the users or automatically by Ocean software tools, the problem is translated into an Ising formulation. Once the problem has been modeled, it can be represented as a graph where vertices are qubits and weighted edges between them are couplers with different strengths (edges weights). This graph has to be embedded into the QPU architecture, which is often represented as a bigger graph of the same kind. After embedding the problem, each qubit and coupler receives a weight that will influence the result according to the problem formulation. Once the problem is properly setted and adjusted in the QPU, the Quantum Annealing process starts. The device uses Quantum Annealing to sample from low-energy eigen states of the Hamiltonian and seek the minimum of the resulting energy landscape, which is the solution for the problem. Some problems require a post-processing of the results. These can include several sampling to get better statistical probability about which is the optimal result. The solution may need to be adapted to a meaningful expression (for example, adding the constant terms of the function gives sense to the final result).

3. **QPU Architectures**

   QPU architectures are essential to compute different problems. Quantum problems need to be embedded in a grid of physical connected qubits. The layout design determines the QPU architecture. To understand this architecture let's define some essential terms. Quantum processors have a set of qubits Q. There is also a set of couplers C that connects some pairs of qubits. The QPU architecture is represented as a graph formed by small cells (sub-graphs) which are connected between them.

   In the Chimera topology, whose cells are $4x4$ bipartite graphs that are connected by external couplers. D-Wave 2000Q QPU supports a Chimera graph $C16$, that is to say, it has a grid of $16x16$ cells of 8 qubits each. The graph of that specific QPU has 2048 physical qubits $(16 \times 16 \times 8)$ with $6,016$ couplers that connect them. Chimera qubits have a degree of 6 (are coupled to 6 different qubits) and a nominal length of 4 (qubits are connected to 4 other qubits of the same cell). In the Figure 2.14 we show a C3 Chimera graph.

4. **Embedding and qubit chains**

   Knowing that the mathematical function is correctly modeled as a graph, this graph must be embedded in the QPU topology. This process is called "minor embedding". The objective of this process is to find a sub-graph (a minimal graph) of the QPU topology which represents the problem graph. The objective is to assign one physical qubit to each variable of the problem paying attention to the connection between them. However, as QPU topologies Chimera and Pegasus are not fully connected graphs, it is necessary to associate multiple physical qubits to one logical qubit in order to fit the problem graph into a topology graph. This association of different physical qubits to represent ate the

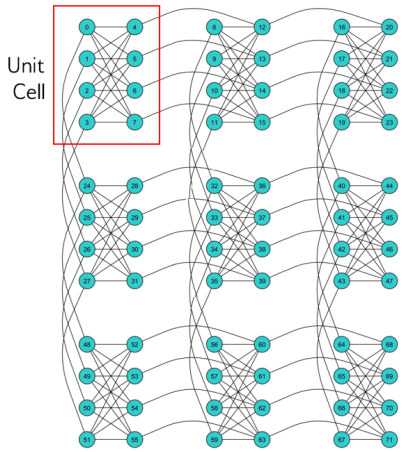


FIGURE 2.14: A C3 Chimera graph

same logical value is called a "qubit chain". The objective of qubits chains is to force different qubits to have the same value and preserve problem consistency.

5. **Solvers and Samplers**

   According to D-Wave's description, solvers are resources that run problems. Some of these solvers are an interface to use QPU, whereas other solvers leverage CPU and GPU resources. Solvers are a set of functions (an API) that facilitates developing and running problems on D-Wave's supercomputers. Solvers run processes called "samplers". These processes sample from low energy states of the problem's objective function.

2.7 Quantum Complexity Theory

Before defining quantum algorithms, we start by defining the quantum complexity, and its classes. To do this, we start by defining the classes of classical complexity in the following:

- **P:** Defined for decision problems that are solved efficiently in polynomial time with the number of data to be processed N (N is the problem size). The problems in this class are said to be easy to solve (e.g., searching in lists, finding a path in a directory, etc.).

  The computational time required to solve problems in this class is simply proportional to N^M with M an integer that does not depend on the problem and does not require significant computational resources (at least for problems in the same class).

- **NP:** This class is defined for problems in which it is easy to check the validity of a solution efficiently (a valid solution can be obtained in polynomial time) but not always solved efficiently. In theory, some of the problems associated with this class are more complex than the previous ones, and they have at best an exponential computation time when the method used is simply to test all possibilities. In practice, this type of problem is adapted to quantum computers since these computers are theoretically able to handle 2^N combinations.

- **NP-Complet:** The class of problems is defined as a subset of the class NP. One of the reasons why we think that $P \neq NP$ is the existence of this class.

Indeed, if only one problem in NP-complete is solved in polynomial time, then all problems in NP can be solved in polynomial time so $P = NP$. But no polynomial algorithm has ever been discovered for any problem in NP-complete. NP-complete problems are, in a sense, the most difficult problems to solve in NP. In this class, more than 3000 NP-complete problems are identified, including the SAT problem. This problem is defined as a Boolean formula composed of variables $x_1, \ldots, x_n$ and connectors (and, or, not).

- **NP-Hard:** This class of problems corresponds to the optimization counterpart of NP-complete problems. Currently, the vast majority of practical problems belong to this class, including fundamental problems in many related disciplines (scheduling, Sudokus, etc.) [AB09]. There is a catalog of NP-hard properties [GJ79] that includes a large number of problems from 1979 and several websites maintain the most recent optimization problems [CKH95]. This class is also used for optimization problems whose objective is to find a global minimum (or maximum) in a large set of solutions.

In the context of the theory of complexity, the class of problems known as NP is particularly important. Of these problems, we can distinguish those with polynomial complexity named P. Moreover, a particularly important subset of NP problems is called NP-complete [Big86] and has the property that any problem in NP can be transformed into a problem in NP-complete in polynomial time. In Figure 2.15 we show the classification of the different classes.

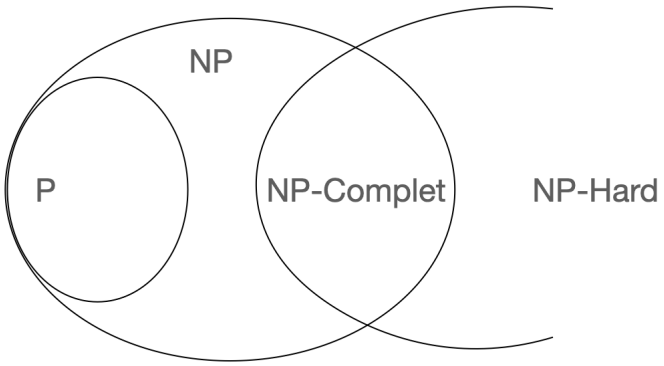


FIGURE 2.15: Classes of complexity

In the quantum part, there are also specific complexity classes for quantum algorithms. We can then add a classification of problems by level of difficulty for quantum computers even if the correspondence with the above classes is still clear.

- **Bounded error Quantum Polynomial time (BQP):** Introduced in the 90's when the first quantum algorithms appeared, this class is intended for problems that can be handled in polynomial time by a quantum computer. Theoretically, under certain conditions, there would be a correspondence between BQP problems and P problems. The first analyses have shown that the P-class of polynomial problems is well within the BQP class [BV97]. It is in this class that many of the well-known quantum algorithms like Grover and Shor are defined. These problems are at the center of current research on quantum algorithms and their uses. Indeed, the question is whether $P \neq BQP$? It is already established that $P \subseteq BQP$ which itself is a subset (strict or not) of NP.

The representation of the complexity classes becomes as follows:

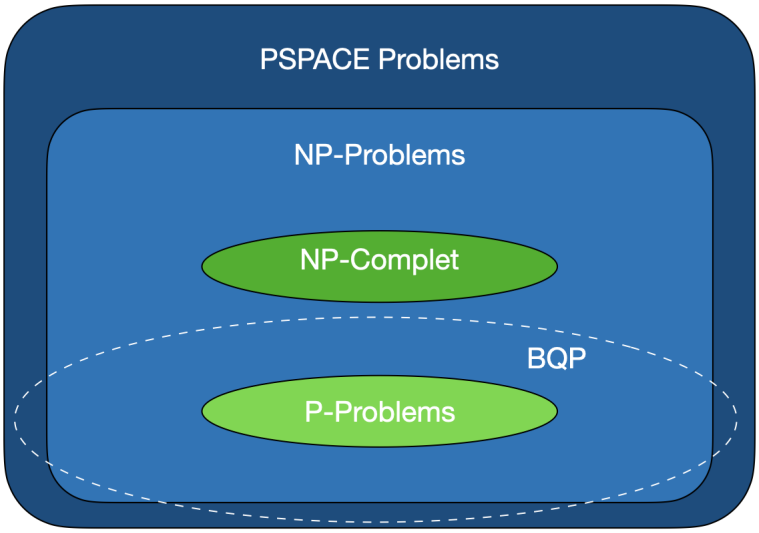


FIGURE 2.16: Classes of complexity with quantum complexity

2.8 Quantum algorithms

In this section, we review the state of the art of the most well-known quantum algorithms.

2.8.1 Deutsch-Jozsa algorithm

We start with the Deutsch-Jozsa algorithm and define its problem [DJ92]. Suppose that we have $N = 2^n$ Boolean variables, such that $x \in \{0,1\}^N$ with the two possibilities:

1. **Constant:** the value of x_i stays always the same, all x_i have the same value 0 or 1,
2. **Balanced:** the number of x_i that have a value of 1 is equal to the number of x_i that have a value of 0,

The main objective of Deutsch-Jozsa's algorithm is to answer the question: x is balanced or constant?

In order to answer this question, the steps of the Deutsch-Jozas algorithm are the followings:

- **Step 1:** Following the Deutsch-Jozas algorithm, we start by initializing n qubits to the state $|0\rangle^n$ as follows:

$$|\psi\rangle = |\psi_0 \dots \psi_{n-1}\rangle = |0\rangle^{\otimes n} \tag{2.38}$$

- **Step 2:** After that, we apply a Hadamard transformation on each qubit $|\psi_i\rangle$ to create a uniform superposition as a result:

$$H^{\otimes n}|\psi\rangle = H^{\otimes n}|0\rangle^n \tag{2.39}$$
$$= \frac{1}{\sqrt{2}^n} \sum_{\psi_i \in \{0,1\}^n} |\psi_i\rangle \tag{2.40}$$

Now, we have a superposition containing all the elements of x. Then, we define the following oracle O_x which represents the following function:

$$f(\psi_0 \dots \psi_{n-1}) = 0 \text{ or } 1, \text{ where } \psi_k = 0 \text{ or } 1, \forall k \tag{2.41}$$

We apply this oracle to the results found in the previous step to find the following:

$$O_x H^{\otimes n}|\psi\rangle = O_x H^{\otimes n}|0\rangle^n \tag{2.42}$$
$$= \frac{1}{\sqrt{2}^n} \sum_{\psi_i \in \{0,1\}^n} (-1)^{x_i}|\psi_i\rangle \tag{2.43}$$

- **Step 3:** After the oracle we add another call of Hadamard's on all the qubits to find the following outputs:

$$H^{\otimes n} O_x H^{\otimes n}|\psi\rangle = H^{\otimes n} O_x H^{\otimes n}|0\rangle^n \tag{2.44}$$
$$= \frac{1}{2^n} \sum_{\psi_i \in \{0,1\}^n} (-1)^{x_i} \sum_{\psi_j \in \{0,1\}^n} (-1)^{i.j}|\psi_j\rangle \tag{2.45}$$

with $\psi_i.\psi_j = \sum_{k=1}^{n} \psi_{ik}\psi_{jk}$, since $\psi_i.0^n = 0, \forall \psi_i \in \{0,1\}^n$, we can observe that the amplitude of the state $|0\rangle^n$ at the end of the experiment is:

$$\frac{1}{2^n} \sum_{i \in \{0,1\}^n} (-1)^{x_i} = \begin{cases} 1 & \text{if } x_i = 0 \text{ for all } i \\ -1 & \text{if } x_i = 1 \text{ for all } i \\ 0 & \text{if } x \text{ is balanced} \end{cases} \tag{2.46}$$

Finally, if we observe the state $|0^n\rangle$ at the end, we can conclude that x is constant and balanced otherwise. Therefore, the Deutsch-Jozsa problem can be solved with only one quantum query and O(n) other operations. The general circuit of the Deutsch-Jozsa algorithm is shown in Figure 2.17.

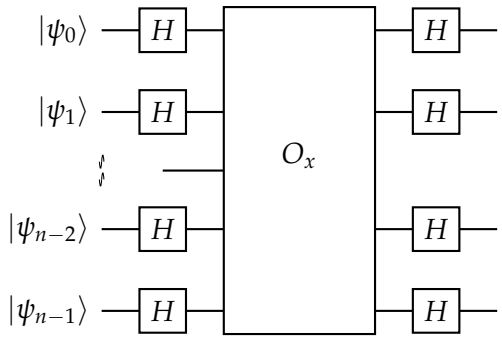


FIGURE 2.17: Deutsch-Jozsa Algorithm

2.8.2 Quantum Fourier transform

Fourier transformation has been very powerful in many areas of science, in addition, to be an elegant tool, it often transformed a difficult problem into an easier one.

Similarly to its classical version, the quantum Fourier transform acts on a quantum state $\sum_{i=0}^{N-1} x_i |i\rangle$ and maps it to the quantum state $\sum_{i=0}^{N-1} y_i |i\rangle$ where

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=1}^{N-1} x_j w_N^j k \tag{2.47}$$

which is the discrete Fourier transform of the amplitude x_k. This transformation is unitary and can be implemented as the dynamics for a quantum computer (cf. [NC11] p. 217). Moreover, it maps the computing basis states $|k\rangle_{k\in\mathbf{Z_d}}$ to another basis as follows:

$$|w_k\rangle = \mathtt{F}\,|k\rangle = \frac{1}{\sqrt{d}} \sum_{l=0}^{d-1} e^{2\pi i k l/d}\,|l\rangle \tag{2.48}$$

The new basis $|w_k\rangle_{k\in\mathbf{Z_d}}$ is called the quantum Fourier basis.

2.8.3 Quantum phase estimation

Quantum phase estimation is one of the important routines of quantum computing as it is convened in many quantum computing algorithms (e.g. state separation [Zha+19], accelerating variational quantum eigensolver [WHB19], tensor principal component analysis [Has20], ...etc).

Given a unitary operator U with $|\psi\rangle$ an eigenvector and $e^{2\pi i\theta}$ its corresponding eigenvalue ($U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$) since θ is not known a priory, the quantum phase estimation algorithm will serve to estimate it.

The quantum phase estimation procedure uses two registers. The first register contains t qubits[3] initially in the state $|0\rangle$ we call this the counting register. The second register will serve to store the eigenvector.

In the computational basis, we store numbers in binary form using the states $|0\rangle$ and $|1\rangle$, but in the Fourier Transform basis, we store numbers using different rotations around the Z-axis. This is used by the quantum phase estimation algorithm to write the phase of U (in the Fourier basis) to the t qubits in the counting register, then again using the inverse quantum Fourier transform to go back to the computational basis to be measured.

Recall that to represent a number in a binary form we use the following scheme:

$$a_0 \times 2^0 + a_1 \times 2^1 + a_2 \times 2^2 + a_3 \times 2^3 + a_4 \times 2^4 + ... + a_n \times 2^n$$

Where coefficients $a_0, a_1, a_2, \ldots$ are bits and, the power i corresponding to a bit a_i is called its weight.

[3] The choice of t depends on the number of digits of accuracy we wish to have in our estimate for θ, and with what probability we wish the phase estimation procedure to be successful (cf. [NC11]).

For instance, we can encode 5 as in Figure 2.18.

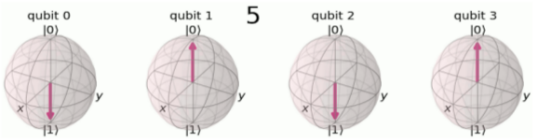


FIGURE 2.18: 5 as a binary string

There is another way of coding numbers using "phases". For instance, we can encode 5 as in Figure 2.19.

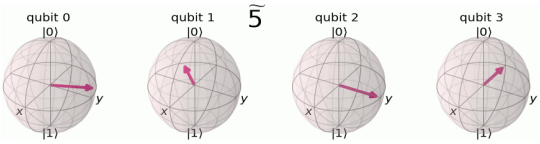


FIGURE 2.19: 5 as a phase superposition

Indeed, to count to a number, x between 0 and 2t, we rotate this qubit by $x2t$ around the z-axis. For the next qubit we rotate by $2x2t$, then $4x2t$ for the third qubit.

Therefore, in the Fourier basis, the topmost qubit completes one full rotation when counting between 0 and 2t.

When we use a qubit to control the U gate, the qubit will turn (due to kickback) proportionally to the phase $e^{2i\pi\theta}$. We can use successive C-U gates to repeat this rotation an appropriate number of times until we have encoded the phase θ as a number between 0 and 2t in the Fourier basis.

Then we use $QFT^\dagger$ to convert this into the computational basis before measurement (Cf. Figure 2.20).

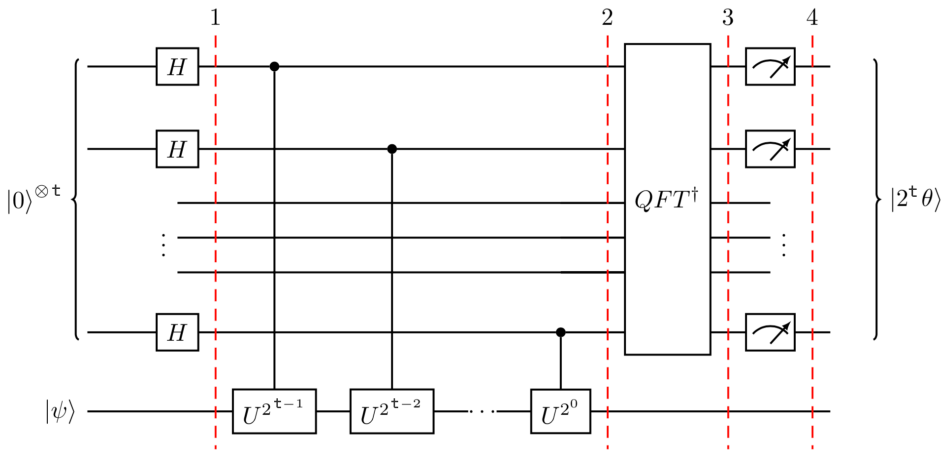


FIGURE 2.20: Quantum Phase Estimation Circuit (src Qiskit documentation)

2.8.4 Grover search

Grover's search algorithm allows to search for an element in a set of unstructured data. In Figure 2.21 it is considered to begin the algorithm in the starting state $|\alpha\rangle$ and the objective is reaching the final state $|\beta\rangle$.

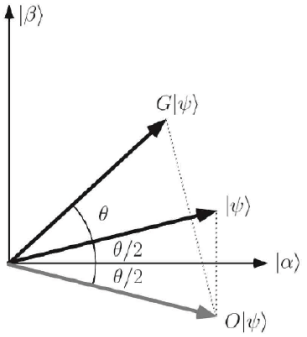


FIGURE 2.21: Geometrical representation of the effect of Grover search. the state $|\beta\rangle$ is assumed to be the solution of a given problem.

Each step of the algorithm consists in 2 rotations, which practically is implemented by working on the amplitudes of the state. In particular, the aim is the maximization of the "good" state's amplitude. The method has been generalized in [Bra+00] and called *quantum amplitude amplification*; maximize such amplitude, first, a rotation around the initial state is done by a given angle $\frac{\theta}{2}$ and then a second rotation, that will be performed by the so called *Grover operator* which will rotate around the new state $|\psi\rangle$ by a full angle θ. The key parameter here is the angle of rotation θ: it is needed to be the largest in order to reach with the smallest amount of operations the final state but *without* exceeding it. The problem is indeed that an excessive rotation would lead the amplitude to decrease, thus decreasing the probability of having a good result.

If we consider $|\alpha\rangle$ as the superposition of vectors that are not solutions of the problem we deal with, and $|\beta\rangle$ the superposition of the vectors that are solutions of our problem. A Grover iteration is the space spanned by $|\alpha\rangle$ and $|\beta\rangle$ is a rotation of an angle θ (cf. Figure 2.21). This rotation can be described as a matrix with eigenvalues $e^{i\theta}$ and $e^{i(2\pi-\theta)}$.

Definition 2 The Quantum Oracle is a black box used to estimate a function using qubits. It allows to transform a system from a quantum state $|x\rangle$ to a state $|f(x)\rangle$, by the evolution of quantum states.

$$O\,|x\rangle = |f(x)\rangle$$

As in the case of quantum phase estimation, Grover search is based on the possibility of building oracles to implement the U^{2x} (or at least polynomial time subroutines to solve some yes/no questions). There are many ways to implement the oracle effects [NC11]. In the *Quantum Learning Machine* of Atos *QLM* the "phase oracle" that changes the vector signs are considered.

The relationship between the angle θ and the number of solutions M is determined by the formula $sin^2(\theta/2) = M/N$. (N being the number of all potential candidates).

2.8.5 Quantum walk

Quantum walks are the correspondence of the classical random walks in quantum mechanics. Their main difference is that quantum walks do not converge to some limiting distribution. Thanks to quantum interference, quantum walks can propagate much faster than classical random walks. The current literature offers an explicit and complete introduction to quantum walks [ADZ93], [Kem03], [VA12], [BCA03]. The concept of quantum walking was first proposed by Aharonov et al [ADZ93] in 1993. Kempe [Kem03] presents two types of quantum walks including discrete time quantum walks and continuous time quantum walks. We will present a simple example in a one-dimensional space to help readers quickly understand the basic ideas of discrete-time quantum walks in the next subsection and a comparison between quantum and classical walks in a graph in the next.

Generality of quantum walk

We define a space $\mathcal{H} = \mathcal{H}_p \otimes \mathcal{H}_c$ for one-dimensional quantum walks [Kem03]. $\mathcal{H}_p$ denotes the Hilbert space that is encompassed by the positions of the particle. For a one-dimensional Hilbert space, it can be represented by:

$$\mathcal{H}_p = \{|i\rangle : i \in \mathbf{Z}\} \tag{2.49}$$

For an N-dimensional Hilbert space, it can be represented by:

$$\mathcal{H}_p = \{|i\rangle : i = 0 \dots N-1\} \tag{2.50}$$

where $|i\rangle$ is a particle located at position i. $\mathcal{H}_c$ denotes the coin space that is covered by two basic states $\{|\uparrow\rangle, |\downarrow\rangle\}$ (spin up and spin down respectively).

The unitary operation S defines the conditional translation on the space $\mathcal{H}$:

$$S = |\uparrow\rangle\langle\uparrow| \otimes \sum_i |i+1\rangle\langle i| + |\downarrow\rangle\langle\downarrow| \otimes \sum_i |i-1\rangle\langle i| \tag{2.51}$$

where $i \in \mathbf{Z}$, $\otimes$ is the tensor product that separates two degrees of freedom, spin, and space, and will allow us to visualize more clearly the resulting correlations between these two degrees of freedom [Kem03]. S can realize the following equations:

$$S(|\uparrow\rangle \otimes |i\rangle) = |\uparrow\rangle \otimes |i+1\rangle \tag{2.52}$$

$$S(|\downarrow\rangle \otimes |i\rangle) = |\downarrow\rangle \otimes |i-1\rangle \tag{2.53}$$

This means that the particle jumps to the right if it has a spin up, and to the left, if it has a spin down.

One of the most frequently used unitary transformations is the Hadamard transformation H [Kem03]. Here is an example of H:

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.54}$$

Hadamard's walk on Z is:

$$|\uparrow\rangle \otimes |0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle) \otimes |0\rangle \tag{2.55}$$

$$\xrightarrow{S} \frac{1}{\sqrt{2}}(|\uparrow\rangle \otimes |1\rangle + |\downarrow\rangle \otimes |-1\rangle) \tag{2.56}$$

Then the one-step quantum walk transformation can be defined as follows:

$$U = S(C \otimes I). \tag{2.57}$$

Where C is a unitary transformation allowing the rotation of the spin in $\mathcal{H}_c$. A quantum walk of t steps is defined as the transformation U^t.

Quantum walk on a graph

The quantum walks on a graph G, is introduced in [Mon05], where the space $\mathcal{H}_p$ has a dimension equal to the number of vertices of the graph G. And for each vertex v_i, we have a space $\mathcal{H}_c$ of d dimension where d is the number of edges coming out of the vertex v_i. The matrix C is to operate the spins in $\mathcal{H}_c$ and S is to apply the walk in each vertex. Finally, the unitary operation U allows to apply a single walk in the graph G. With this strategy, we can propagate in the graph at each time t in all directions, contrary to the random walk, we must choose each time a destination in a random way.

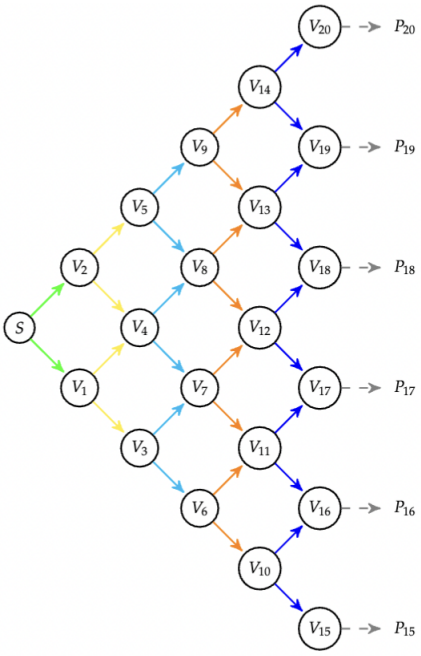


FIGURE 2.22: Quantum walks

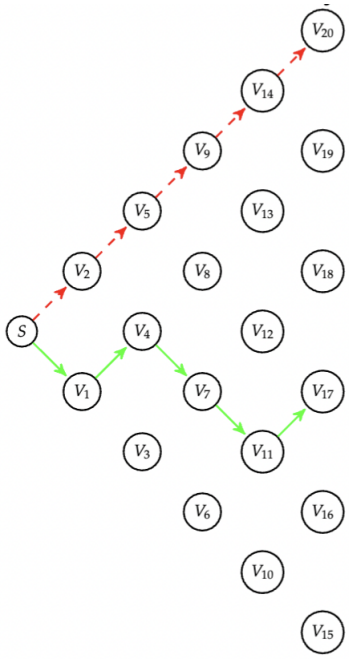


FIGURE 2.23: Random walks

In Figures 2.22 and 2.23, we show the general difference between the quantum and random walks. In Figure 2.22, we show the quantum walks in the graph with different colors in each arc, each one represents a quantum walk applied by a unit operation U_t where $t = 0, \ldots, 4$. In this example, after five walks we will find the probability of reaching each vertex $v_i, i = 15, 16, \ldots, 20$ with the probability of the states that represent the vertices in the output superposition. With this propagation strategy on the graph. We are assured to have walked through all paths between the

vertex s and the destinations, because, at each time t, all the available destinations are processed at the same time, instead of choosing one at each time randomly.

On the other hand, in Figure 2.23, we begin with the source s, and at each instant t, we throw a coin, if we find pile we move to the right and if face we move to the left. With the random walk strategy, at each call of the algorithm, we find a path, in order to find the probability of reaching each vertex, we run the algorithm a number of times and calculate the probability of finding each one. With this approach, we are not sure that we have processed all possible paths between s and the destination vertices. For example, in order to get the red path of Figure 2.23, we need to find 5 times successively the face, the probability of finding this is very small, and if we increase the size of the graph, the probability of finding these types of paths is very low.

2.8.6 Other algorithms

1. **Shor's Factoring Algorithm [Sho99b]**: Shor's algorithm aims at solving the factorization problem, which is used in most of our computer security systems. Specifically, given a number N, how can we find two prime numbers p and q such that $N = p \times q$? Today, this problem becomes impossible to solve for numbers with more than 500 digits. Peter Shor demonstrated in 1994 that with a quantum computer, it would be possible to factor numbers that have thousands or millions of digits. Shor uses the link between the factorization problem and the search for a period in a functional. By using quantum superposition, he shows how to make quantum states resonate around the period of the function to find it, and then use this result to factor quickly.

2. **HHL algorithm [Llo10]**: The HHL algorithm is used to solve linear equations. Under some assumptions, this algorithm can solve linear equations with an exponential speedup compared to the classical algorithm.

2.9 Quantum Machine Learning

In this section, we give a brief overview of Quantum Machine Learning algorithms, both in the supervised and unsupervised categories, as well as a description of the quantum version of the PCA algorithm, and Quantum Neural Networks.

2.9.1 Quantum K-means algorithm

We start with the most famous algorithm in the field of unsupervised machine learning, K-means. It allows to partition the data into K clusters, based on the similarity between data, where each cluster contains the most similar data. The classical version of this algorithm requires three steps in each iteration:

1. Calculate the similarity between each individual data and each center,

2. Assign each individual data to the nearest cluster,

3. Update the centers.

The complexity of this algorithm according to the version of Lloyds [SMR08] is $O(NM)$, with N as the dimension of data and M as the number of observations in the dataset.

The quantum version of this algorithm has been published in the paper [Ker+19], which proposes an approach with an exponential speed-up according to the number of observations in the dataset, compared to the classical K- means algorithm. This quantum version requires a mechanism to compute the similarity between the data in a quantum manner, and also a method to search for the nearest cluster in a quantum manner for the assignment.

Before showing the algorithm directly, we start by explaining the procedure to calculate the similarity between two data in the next subsection, and in order to search the nearest center we use Grover's algorithm already described in 2.8.4.

SwapTest

The SwapTest circuit (see Figure 5.1) has three inputs: a control qubit and two registers, the first register $|\alpha\rangle$ to represent the first vector α and the second $|\beta\rangle$ for the vector β. This small circuit 5.1, allows to compute the overlap $\langle\alpha|\beta\rangle$ by measuring the control qubits. It has been used to compute the similarity between two vectors for the first time in the paper [ABG06]. To clearly understand how this circuit allows us to find the similarity between to vectors we describe in the following mathematical analysis.

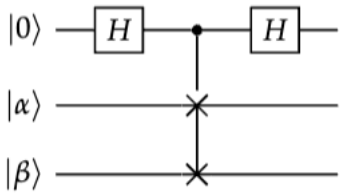


FIGURE 2.24: SwapTest Circuit

Let us suppose that we have two vectors α and β represented by the two quantum states $|\alpha\rangle$ and $|\beta\rangle$ with n qubits for each one. The mathematical progress of the circuit is as follows:

1. As input of the circuit we have these two quantum states $|\alpha\rangle$, $|\beta\rangle$, and a control qubit initialized to the state $|0\rangle$ as follows:

$$|\psi_0\rangle = |0, \alpha, \beta\rangle \tag{2.58}$$

2. After the initialization, we apply the Hadamard gate to the first control qubit which gives:

$$|\psi_1\rangle = (H \otimes \mathbb{I}^{\otimes n} \otimes \mathbb{I}^{\otimes n}) |\psi_0\rangle = \frac{1}{\sqrt{2}}(|0, \alpha, \beta\rangle + |1, \alpha, \beta\rangle) \tag{2.59}$$

3. We apply the swap gate on the two registers $|\alpha\rangle$ and $|\beta\rangle$ under the control with $|1\rangle$ of the first qubit:

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0, \alpha, \beta\rangle + |1, \beta, \alpha\rangle) \tag{2.60}$$

4. We apply a second Hadamard to the control qubit:

$$|\psi_3\rangle = \frac{1}{2}|0\rangle\left(|\alpha,\beta\rangle + |\beta,\alpha\rangle\right) + \frac{1}{2}|1\rangle\left(|\alpha,\beta\rangle - |\beta,\alpha\rangle\right) \tag{2.61}$$

5. We measure the control qubit, and the probability of measuring the state $|0\rangle$ is computed as follows:

$$\begin{aligned} P(|0\rangle) &= |\frac{1}{2}\langle 0|0\rangle\left(|\alpha,\beta\rangle + |\beta,\alpha\rangle\right) + \frac{1}{2}\langle 0|1\rangle\left(|\alpha,\beta\rangle - |\beta,\alpha\rangle\right)|^2 && (2.62)\\ &= \frac{1}{4}|(|\alpha,\beta\rangle + |\beta,\alpha\rangle)|^2 && (2.63)\\ &= \frac{1}{4}(\langle\beta|\beta\rangle\langle\alpha|\alpha\rangle + \langle\beta|\alpha\rangle\langle\alpha|\beta\rangle + \langle\alpha|\beta\rangle\langle\beta|\alpha\rangle + \langle\alpha|\alpha\rangle\langle\beta|\beta\rangle) && (2.64)\\ &= \frac{1}{2} + \frac{1}{2}|\langle\alpha|\beta\rangle|^2 && (2.65) \end{aligned}$$

According to the probability $P(|0\rangle)$, we can decide that $|\alpha\rangle$ and $|\beta\rangle$ are orthogonal if $P(|0\rangle) = 0.5$ and identical if $P(|0\rangle) = 1$.

Lloyd et al in the paper [LMR13] demonstrate how we can calculate the Euclidean distance using SwapTest, the method is named DistCalc and described as follows:

1. We encode each vector x in a quantum state as [NDW16] [Mot+04]:

$$|x|^{-1}x \rightarrow |x\rangle = \sum_{i=1}^{N}|x|^{-1}x_i|i\rangle \tag{2.66}$$

   with this encoding we use only $log_2 N$ qubits with N is the dimension of x, and the state $|x\rangle$ is normalized because $\langle x|x\rangle = |x|^{-2}x^2 = 1$.

2. In order to use SwapTest, we initialize the two states as follows:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0,\alpha\rangle + |1,\beta\rangle) \tag{2.67}$$

$$|\phi\rangle = \frac{1}{\sqrt{Z}}(|\alpha|\,|0\rangle + |\beta|\,|1\rangle) \tag{2.68}$$

   with $Z = |\alpha|^2 + |\beta|^2$

3. We use SwapTest and calculate the Euclidean distance as follows:

$$|a-b|^2 = 2Z|\langle\phi|\psi\rangle|^2 \tag{2.69}$$

Algorithm

Using SwapTest and Lloyd et al's method to compute the distance between the data and also with Grover's algorithm shown in section 2.8.4, we can illustrate the general Quantum K-means algorithm as follows:

2.9.2 Quantum K-medians algorithm

The K-medians algorithm is the same as the K- means algorithm except on the side of updating the centroids, instead of calculating the mean to find the new centroids in

Algorithm 1: Quantum K- means algorithm

Input : A set of data $X = \{x_1, x_2, \ldots, x_M\}$, where each data is of dimension N. The number of clusters K

Initialization Initialized K centers $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^N$ in a random way or with any initialization used in the classical case of the K- means algorithm.

while *Not converged,* $||\mu_j^{s+1}\mu_j^s|| < \nu$ **do**

1. For each data $x_i, i = 1, \ldots, M$, calculated K distances $d_{i,k} = ||x_i - \mu_k||$ using the quantum method proposed by Lloyd.

2. Used the GroverOption algorithm to find the closest cluster to each data:

$$c_i := arg\{\min_k ||x_i - \mu_k||^2\} \tag{2.70}$$

3. Update the centroids by calculating the average data of each cluster. The new centroids become as follows:

$$\mu_j = \frac{1}{|C_j|} \sum_{i \in C_j} x_i \tag{2.71}$$

where C_j is the set of data of the cluster j and $|C_j|$ is the number of data belonging to the cluster j.

end

each iteration, we calculate the median between data. To do that, the paper [ABG07] allows us to present the following method to calculate the median in a quantum way.

Let $\{x_1, x_2, \ldots, x_M\}$ be a set of M points of dimension N, the median of this set is represented by the element x_i that minimizes the distance with all other data of this set. So to find the median, we compute the following distance for each data x_i:

$$d_i = \sum_{j=1}^{M} ||x_i - x_j|| \tag{2.72}$$

and we take as median the element x_i which has the smallest d_i.

Then, the method to calculate the median quantically, called MedianCalc, is represented by two instructions:

1. Calculate the distances d_i using DistCalc,

2. Use GroverOption to find the smallest distance and the median.

Now, we have the method of calculating the median in a quantum way, it only remains to present the general quantum algorithm K-medians:

The complexity of this algorithm is identical to that of the quantum K-means algorithm, except that here there is an additional quadratic gain for the calculation of the median.

Algorithm 2: Quantum K-medians algorithm

Input : A set of data $X = \{x_1, x_2, \ldots, x_M\}$, where each data is of dimension N. The number of clusters K

Initialization: Choose K centers $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^N$ in a random way from the training set X.

while *Not converged,* $\mu_j^{s+1} \neq \mu_j^s$ **do**

1. For each data $x_i, i = 1, \ldots, M$, calculated K distances $d_{i,k} = ||x_i - \mu_k||$ using the quantum method proposed by Lloyd.
2. Used the GroverOption algorithm to find the closest cluster to each data:
$$c_i := arg\{\min_k ||x_i - \mu_k||^2\} \tag{2.73}$$
3. Update the centroids by calculating the median of each cluster using MedianCalc.

end

2.9.3 Quantum Support Vector Machines

In the part of supervised machine learning, the most known quantum algorithm until today is the quantum version of the Support Vector Machines (SVM) algorithm. It allows us to find a hyperplane that allows us to separate the classes between them with a maximum margin with the data points. This hyperplane is used for future data classification as a frontier decision.

This quantum version of SVM is proposed in the paper [Ang+03], and it allows to use Grover's algorithm to minimize the following objective function:

$$\min_{\alpha^{(i)}}\{\frac{1}{2}\sum_{i,j}\alpha^{(i)}\alpha^{(j)}y^{(i)}y^{(j)}K(x^{(i)}, x^{(j)}) - \sum_{i=1}^{M}\alpha^{(i)}\} \tag{2.74}$$

with $K(x^{(i)}, x^{(j)})$ is the kernel function. The general algorithm is 3.

2.9.4 Quantum Principal Component Analysis

The Principal Component Analysis (PCA) algorithm allows to reduce the dimension of the data from N to R. The paper [LMR14], proposes a quantum approach to this algorithm with an exponential complexity faster than the classical approach. The quantum version of this algorithm is based on the quantum phase estimation algorithm presented in the section 2.8.3. The steps of this algorithm are as follows:

1. Encode the data using the method proposed in [NDW16] [Mot+04]: for each data x of X:
$$x \rightarrow |x\rangle \tag{2.77}$$
2. Representation of the covariance/correlation matrix as a density matrix
$$\rho = \frac{1}{M}\sum_{i=1}^{M}|x^{(i)}\rangle\langle x^{(i)}| \tag{2.78}$$

Algorithm 3: Quantum SVM algorithm

Input : A set of data $X = \{x_1, x_2, \ldots, x_M\}$, where each data is of dimension N.

Initialization:

Initialized all parameters and the kernel function

Start:

1. Encoding the data, for each data x we use the binary representation as follows:

$$x \rightarrow a = (a_1, a_2, \ldots, a_k)^T \tag{2.75}$$

with $a_i = \{1, 0\}, \forall i = 1, \ldots, k$. After that, we use this direct binary encoding as a quantum state:

$$|a_1, a_2, \ldots, a_k\rangle \tag{2.76}$$

2. Encode the objective function as a quantum oracle and use the Grover algorithm presented in section 2.8.4 to find the optimal solution.

3. Use the approach introduced in [LMR14] to construct an unitary matrix $U = e^{i\rho t}$ from the matrix ρ.

4. Use the unitary matrix $U = e^{i\rho t}$ and try to find eigenvectors and eigenvalues by the algorithm Quantum Phase Estimation.

2.9.5 Quantum Neural Networks

Classical Neural Networks are built by a series of neurons connected between them with each layer at a matrix of weights, where this matrix of weights minimizes the objective function. The quantum version of the feedforward neural networks algorithm is based on the paper [Wan+17]. where inputs and outputs are represented by quantum states, and the objective function by a unitary transformation, the general architecture of a quantum neural network is as follows:

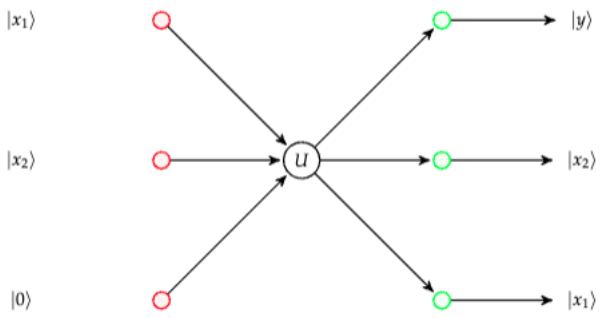


FIGURE 2.25: Quantum Neural Network

The objective function to be minimized by the quantum neural network is the following:

$$C = -\sum_{j=1}^{M} \langle y_{model}^{(j)} | y^{(i)} \rangle \tag{2.79}$$

2.10 Conclusion

This chapter gives a general mathematical review of quantum computing and allows to define what is a system of one qubit, two qubits, and n qubits. In each case, it allows to present how we can apply some transformations to these qubits. It presents the concept of quantum annealing, with the QUBO problem, gives the different classes of classical complexity, and positions the classes of quantum complexity in advance. The most well known and used quantum algorithms and quantum machine learning algorithms are presented in this chapter.

Chapter 3

A Quantum Vertex Separator Approach for Directed Graphs

The Vertex Separator Problem of a directed graph consists in finding all combinations of vertices which can disconnect the source and the terminal of the graph, these combinations are minimal if they contain only the minimal number of vertices. This chapter is based on our contribution in the paper [Zai+22], in which we introduce a new quantum algorithm based on a movement strategy to find these separators in a quantum superposition with a linear number of qubits and a linear number of movements. Our quantum algorithm is tested on small directed graphs using a real Quantum Computer made by IBM.

3.1 Modeling the reliability diagram of a PSA system using a directed graph

In the field of Probabilistic Safety Assessment (PSA), according to the paper [Hib13], we can use a directed graph to model the reliability diagram of a system, using for each component of the system a vertex of the graph, and the connection between two components an edge. For example, if we take a system with three missions, where the reliability diagram is represented as follows:

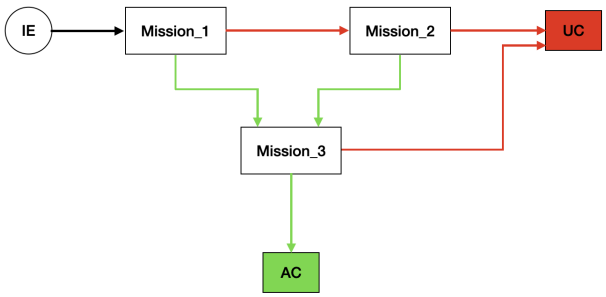


FIGURE 3.1: Reliability diagram of a 3 missions system

Each mission represents a part of the system, which is composed of several components. Each mission has a specific task, which can be a primary or a backup mission, i.e. replacing the failure of another mission. The three missions: Mission 1, Mission 2 and Mission 3 are represented by the three structures in Figures 3.2, 3.3 and 3.4 respectively.

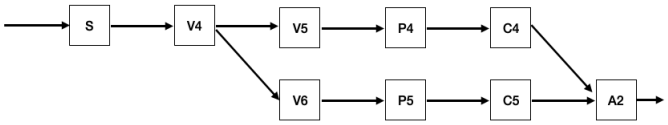


FIGURE 3.2: Mission 1

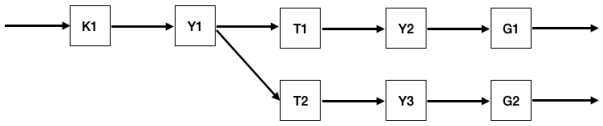


FIGURE 3.3: Mission 2

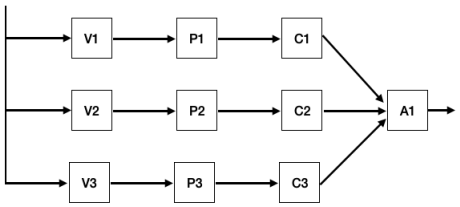


FIGURE 3.4: Mission 3

To build the directed graph of the system 3.1, we use for each component of the system a vertex and each connection an edge between the vertices. Then, the general graph of the system is represented in Figure 3.5. Each subgraph encapsulated by a rectangle represents a mission.

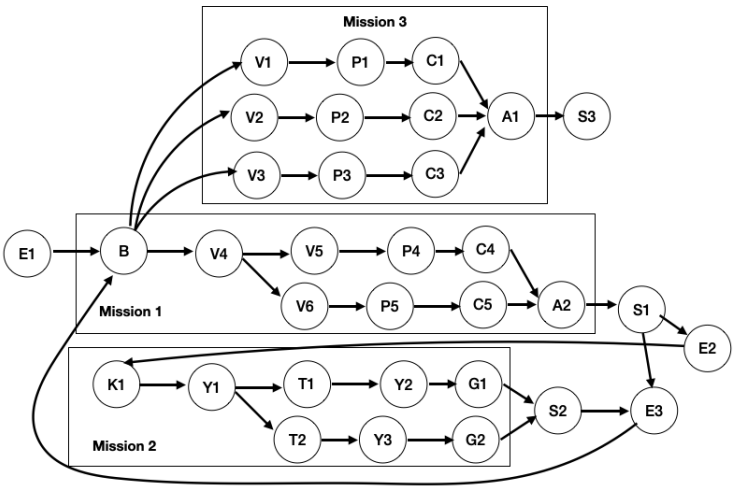


FIGURE 3.5: Directed graph of a 3 missions system

The graph 3.5 can be simplified by using some collection techniques [Hib13], we collect all the vertices in succession that have the same impact on the system, for example in the graph 3.5, the three components V_1, V_2, and V_3 have the same impact on the system, if one or two or all three components fail. So, these three components can be replaced by a single global vertex. After simplifying the graph we will find for this system the simplified graph 3.6.

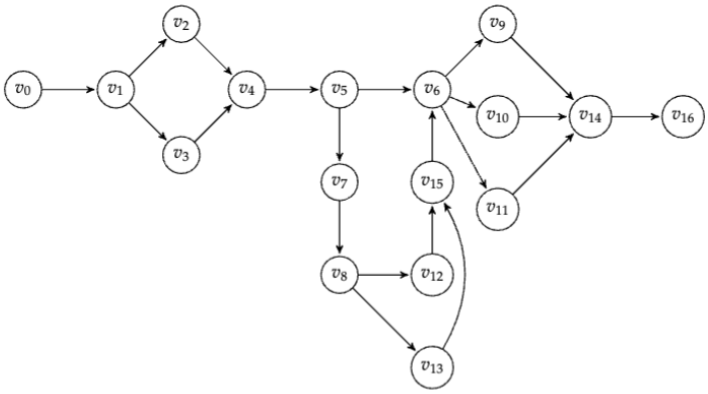


FIGURE 3.6: Simplified directed graph of a 3 mission system

Now, in order to identify the combinations of basic events that can generate unacceptable consequences in the system 3.1, it is enough to find all the vertex separators that can stop the flow in the graph 3.6 between the source v_0 and the terminal v_{16}. In order to find these vertex separators, we start by posing the problem using the directed graph.

In graph theory, the Vertex Separator Problem (VSP) consists of finding a subset of vertices (called a vertex separator) that allows the set of vertices in the graph to be divided into two related components. The VSP is NP-hard [BJ92]. There are a number of algorithms that can find these vertex separators. We mention [KL70], which presents a heuristic method for partitioning arbitrary graphs that is both efficient in finding optimal partitions and fast enough to be practical in solving large problems. The paper [FM82] presents a linear time heuristic method for improving network partitions. Both papers [KL70; FM82] are adapted by [AL94; HR98] to generalize the methods and improve the runtime. According to [HH15], the vertex separator problem can be formulated by a bilinear quadratic programming problem. And, recently, several works [Dav+19; HHS18; KD16] and [Kol19] have combined the traditional combinatorial method and the optimization-based method to improve the performance and quality of the separator. We mention the result of [KD20] who introduced a new hybrid algorithm for computing vertex separators in arbitrary graphs using computational optimization.

In the classical approach of the vertex separator problem for a planar graph with n vertices, Lipton and Tarjan [LT79] provided a polynomial time algorithm for finding a single vertex separator. This algorithm was improved in [LT80] for other families of graphs such as fixed genus graphs. These families of graphs include trees, 3D grids, and meshes that have small spacers. To obtain all minimal vertex separators of a graph, Kloks and Kratsch [KK98] provided an efficient algorithm listing all minimal vertex separators of an undirected graph. The algorithm requires a polynomial time

for every single separator found. In this chapter, we are interested in the vertex separator problem (VSP) in a directed graph that has a source s and a bound t, we search in this graph for all vertex separators that separate the source s and the bound t. In order to do this, let us first define a directed graph and a vertex separator:

Definition 3 A directed graph or network is a graph in which the edges are oriented. More precisely, a directed graph is an ordered pair (V, E) including: (i) V a set of vertices and (ii) $E \subset \{(x, y) | (x, y) \in V \times V, x \neq y\}$ a set of oriented edges or arcs that are ordered and distinct pairs of vertices.

Example 2 Take the directed graph of the Figure 3.7, this graph contains 9 vertices, $V = \{s, v_1, v_2, v_3, v_4, v_5, v_6, v_7, t\}$ and 11 edges $E = \{(s, v_1), (s, v_2), (v_1, v_7), (v_1, v_5), (v_2, v_3), (v_2, v_4), (v_3, v_5), (v_4, v_6), (v_4, t), (v_5, v_6), (v_6, v_7), (v_7, t)\}$.

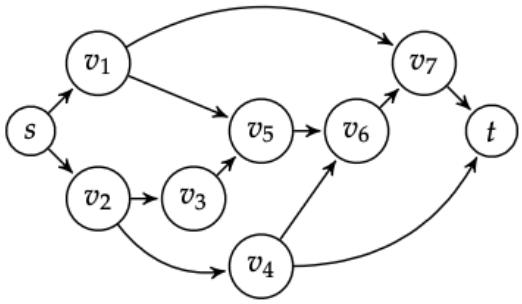


FIGURE 3.7: Example of a directed graph

Definition 4 A vertex separator $s - t$ noted (S, C, T) is a partition of V such that $s \in S$ and $t \in T$. Then the $s - t$-cut for us is a division of the vertices of the graph into three independent subsets S, C, and T, with the source s in the subset S, the terminal t in T and the subset C representing the vertex separator (the cut). The cut C is minimal, which means that the number of vertices existing in C is minimal, that is to say if we remove only one vertex from C the remainder is not sufficient for a cut.

Example 3 Consider the cut $\{v_1, v_2\}$ shown in the Figure 3.8. This cut allows to divide the set of vertices V on three subsets: $S = \{s\}$, $C = \{v_1, v_2\}$ and $T = \{v_3, v_4, v_5, v_6, v_7, t\}$. We say a cut because it breaks the connection between the source s and the terminal t of the graph. In addition to that, if we remove one of the two elements of the subset C (of the cut), we find a path between s and t, which means that removing a single element from C (any element) requires the cut to disappear, which implies that the cut C is minimal.

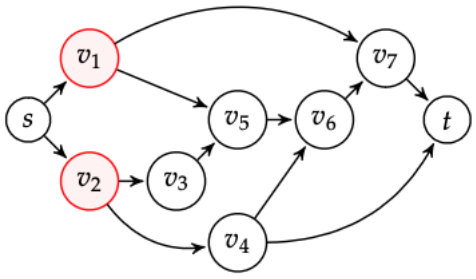


FIGURE 3.8: Minimal cut $\{v_1, v_2\}$

Example 4 Now for the cut $\{v_3, v_4, v_5, v_6, v_7\}$ illustrated in Figure 3.9. It allows to divide the set of vertices V on three subsets: $S = \{s, v_1, v_2\}$, $C = \{v_3, v_4, v_5, v_6, v_7\}$ and $T = \{t\}$. But if we remove the vertex v_6, the remaining part is always a cut between s and t, which means that the cut C is not minimal.

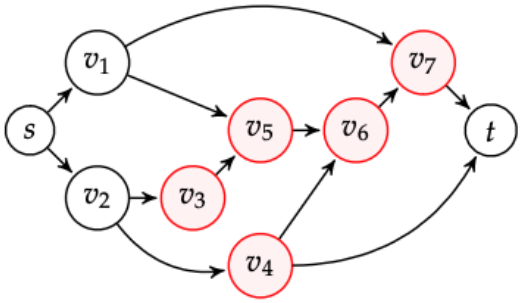


FIGURE 3.9: Cut non-minimal

For a single directed graph, we can find several minimal cuts between the source and the terminal. In the rest of this chapter, we propose our quantum algorithm to determine all the minimal cuts of a directed graph.

3.2 Quantum approach for the search of minimal cuts

In this section, we will describe our quantum algorithm for finding all the minimal cuts that can stop the flow between the source and the terminal of a directed graph. This algorithm is based on a movement strategy that uses movement oracles to construct a quantum superposition that contains all these minimal cuts. The first question that arrives here is how to represent all the sets of vertices with quantum qubits. For a graph with n vertices, we will find 2^n different subsets of these vertices. In the quantum framework, with n qubits, we can represent 2^n possible states. In both cases with n elements, we have 2^n possibilities, so we represent the subsets by the quantum states of these n qubits. To do this, we use for each vertex of the graph a qubit, and each state of these qubits represents a subset of the vertices.

Example 5 Taking the directed graph shown in the Figure 3.7, if we use a qubit for each vertex (see Figure 3.10), we can use the states of these qubits to represent all the possible combinations of the elements of V. For example, with the state $|000111010\rangle$ we can represent the cut $\{v_3, v_4, v_5, v_7\}$ where $|v_i\rangle = |1\rangle$ if $i = 3, 4, 5, 7$ and $|v_i\rangle = |0\rangle, \forall i \neq 3, 4, 5, 7$.

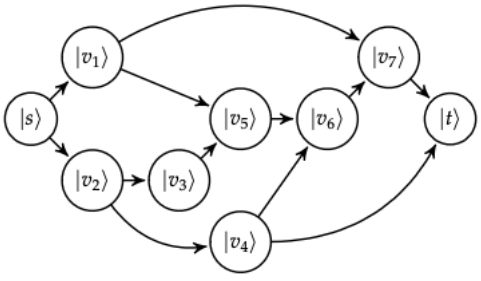


FIGURE 3.10: The representation of a graph by qubits

Before starting to explain the functioning of our algorithm, we start with the definition of a movement in a directed graph, and how these movements are applied by quantum circuits.

Definition 5 Let $G = (V, E)$, a directed graph, the movement of a vertex $v \in V$ corresponds to the move from v to its successors $Succ(v)$.

$$Mov(v) = Succ(v) \tag{3.1}$$

The movement of a vertex $v \in \theta$ where $\theta \subset V$ is the substitution of v by its successors $Succ(v)$ in the set θ.

$$Mov_\theta(v) = \{\theta \setminus v\} \cup Succ(v) \tag{3.2}$$

Example 6 Let's take the graph of Figure 3.7. The movement of the vertex s is represented in Figure 3.11.

Example 7 Consider the movement results of s in the previous example as a set $\theta = \{v_1, v_2\}$. Suppose that we need to apply the movement of v_2 in this set θ, therefore, it is enough to replace v_2 by its successors v_3 and v_4 in θ, then the result is: $Mov_\theta(v_2) = \{v_1, v_3, v_4\}$. These results are presented graphically in Figure 3.12.

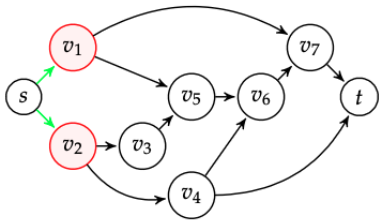


FIGURE 3.11: Example $Mov(s)$

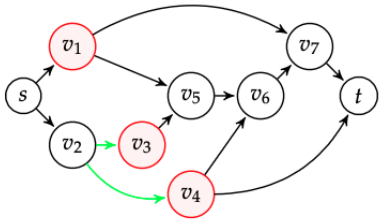


FIGURE 3.12: Example $Mov_{\{v_1,v_2\}}(v_2)$

In the quantum setting, to apply these moves, we use quantum oracles called movement oracles. Each vertex has a movement oracle, which allows to apply the movement if the vertex of the movement exists in the input subset, and provides in the output the input as well as the subsets after the movement.

For a vertex v and a given subset of vertices θ, such that $v \in \theta$. We assume that the subset θ is represented by the quantum state $|\psi_\theta\rangle$. To apply the movement of v, we give the quantum state $|\psi_\theta\rangle$ to the oracle as an input, and in the output of the oracle we find a superposition which contains two states: $|\psi_\theta\rangle$ and $|\psi_{\theta'}\rangle$, with $\theta' = (\theta \setminus v) \cup Succ(v)$.

More generally, suppose that the input set is the union of two subsets $\theta = \theta_1 \cup \theta_2$ represented by the quantum superposition $|\psi_\theta\rangle = \alpha_1 |\theta_1\rangle + \alpha_2 |\theta_2\rangle$, where the state $|\theta_1\rangle$ represents the subset θ_1 and the state $|\theta_2\rangle$ represents the subset θ_2. Consider a vertex $w \in \theta_1$ and $w \notin \theta_2$. The output of the movement oracle of $w \in \theta$ is $|\psi_{out}\rangle = \frac{\alpha_1}{\sqrt{2}} |\theta_1\rangle + \alpha_2 |\theta_2\rangle + \frac{\alpha_1}{\sqrt{2}} |\theta_3\rangle$, where the state $|\theta_3\rangle$ represents the subset $\theta_3 = Mov_{\theta_1}(w) = \{\theta_1 \setminus w\} \cup Succ(w)$.

To provide a general formula for a movement oracle, let $v \in V$ be a vertex, O_v be the movement oracle of v, and $|\psi_\theta\rangle = \sum_i \beta_i |\psi_{\theta_i}\rangle$ is a quantum superposition which represents N subsets of vertices $\{i = 1, \ldots, N\}$. The general formula for the movement oracle of a vertex v is:

$$|\psi'_\theta\rangle = O_v\,|\psi_\theta\rangle = \sum_e \alpha_e f_v(e)\,|\psi_{\theta_e}\rangle \tag{3.3}$$

with

$$f_v(e) = \begin{cases} 1 & \text{if } \theta_e \in \{\theta_i\}_{i=1,\dots,N} \\ 1 & \text{if } \theta_e \in \{Mov_{\theta_i}(v)\}_{i=1,\dots,N} \\ 0 & \text{else} \end{cases} \quad \text{and} \quad \begin{cases} \alpha_e = 0 & \text{if } f_v(e) = 0 \\ \sum_e \alpha_e = 1 \end{cases}$$

In order to represent an oracle with a simple quantum circuit, we need to add two additional control qubits $|c_0\rangle$ and $|c_1\rangle$. The qubit $|c_0\rangle$ is used to check if the vertex of the movement exists in the input set or not, it will be in the state $|1\rangle$, if it exists in the input set, and in $|0\rangle$ otherwise. For this, we use the C-X gate with the qubit corresponding to the vertex of the movement as control and the $|c_0\rangle$ qubit as a target. If the vertex is in the input set, we add another set to the collection of cuts. In other words, if the vertex qubit is in the state $|1\rangle$, we add a new state to the input superposition. To do this, we use the C-H gate with the qubit $|c_0\rangle$ as control and the qubit of the vertex of the movement as a target. Then, we use the C-X gate to apply the movement to the new set. To add all the successors of the movement vertex to the new set, we use the circuit of Figure 3.13 which allows us to flip the qubit of the successor into the state $|1\rangle$ if it is in the state $|0\rangle$ and to do nothing if it is in the state $|1\rangle$.

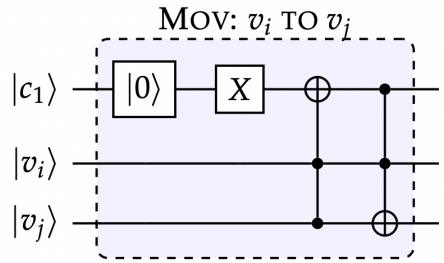


FIGURE 3.13: The movement circuit from v_i to v_j, this circuit uses three qubits: $|v_i\rangle$, $|v_j\rangle$ the movement vertex and the successor respectively and $|c_1\rangle$ for the control.

3.2.1 Algorithm description

Let $G = (V, E)$ be a directed graph, with V being the set of vertices such that $|V| = n$ and E the set of edges, the source of the graph is the vertex s and the terminal is the vertex t. The first step of our algorithm consists in preparing the necessary number of qubits to represent all the possible subsets of vertices of the graph. The graph has n vertices, so we use n quits. These n qubits are initialised in the state $|0\rangle$, so the quantum state it initialized to $|\psi\rangle = |0\dots0\rangle = |0\rangle^n$. With the first qubit corresponding to the source vertex s, the second qubit for the second vertex until the last qubit for the last vertex (the sink vertex t). There are only zeros in the state $|\psi\rangle$, which means all the vertices are absent in the set represented by $|\psi\rangle$.

$$|\psi\rangle = |tv_n\dots v_1 s\rangle = |00\dots00\rangle \iff \psi = \{\} \tag{3.4}$$

To start with a set ψ containing only the input vertex s, we apply the `Not gate` on the first qubit, which gives us $|\psi\rangle = |0\dots01\rangle$ as a result, with the qubit $|s\rangle$ is in the state

$|1\rangle$.

In the second step, for each vertex, we have an oracle of movement, so we call all these oracles of movement. The first oracle corresponds to the movement of the vertex s to its successors:

$$|\psi_1\rangle = O_1 |\psi\rangle = \alpha_1 |\psi\rangle + \alpha_2 |Mov_\psi(s)\rangle \tag{3.5a}$$
$$= \alpha_1 |\psi\rangle + \alpha_2 |Succ(s)\rangle \tag{3.5b}$$

After this iteration, the oracle O_1 adds to the superposition the state $|Succ(s)\rangle$ which represents the first cut of the graph. After that, we apply all the remaining oracles:

$$|\psi_{fin}\rangle = O_n O_{n-1} \dots O_2 |\psi\rangle \tag{3.6}$$

Each one of these oracles adds a number of states to the superposition, which means it adds a number of cuts to the set of cuts represented by the superposition.

$$|\psi_{fin}\rangle = \alpha_1 |cut_1\rangle + \dots + \alpha_k |cut_k\rangle \tag{3.7}$$

After the n oracles, in the output superposition $|\psi_{fin}\rangle = \sum_i \alpha_i |cut_i\rangle$, we find all the possible minimal cuts represented by the states $|cut_i\rangle$. Finally, we use a simple classical filter to eliminate non-minimal cuts. The algorithm 4 represents the steps to generate the quantum circuit to find all the possible minimal cuts.

Algorithm 4: All Minimum cuts sets

Input : Graph $G = (V, E)$, with $n = |V|$ is the number of vertices of the graph, source vertex s, sink vertex t

Output : Min-cuts Cs

Start:

Initialized n qubits,

$$|\psi_0\rangle = |0 \dots 0\rangle$$

Apply the not gate X in the first qubit which represents the source s

$$|\psi_1\rangle = |0 \dots 01\rangle$$

Make the movement of s we apply the oracle O_s

$$|\psi_2\rangle = O_s |\psi_1\rangle$$

for *each* $v \in V$ *and* v *!= s and t is not a successor of* v **do**

$$|\psi_{i+1}\rangle = O_v |\psi_i\rangle$$

end

Cs = measured $|\psi_{n-1}\rangle$ and eliminate non-minimal cuts.

return Cs

3.2.2 Complexity Analysis

Suppose that we have a graph $G = (V, E)$, with V the set of vertices and E the set of edges such that $|V| = n$ and $|E| = m$. To build the circuit, we need n qubits to represent all the possible states of the graph and 2 auxiliary qubits for the control.

For n oracles of movements we need n gates C-H, $2n-2$ gates C-X, $m+2$ gates X and $2m$ gates CC-X. Therefore, our algorithm has a linear complexity either in terms of memory (number of qubits used) or in terms of computation (n oracles of movements).

3.3 The progression of the algorithm through a case study

In this section, we present a detailed version of the case study of our algorithm. For that, let us take the graph $G = (V, E)$ represented in Figure 3.14, where V is the set of vertices of size 9 ($n = |V| = 9$), which is labeled from v_0 to v_8 as follows: $V = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$. And the set of edges between these vertices is noted E and is presented like the following: $E = \{(v_0, v_1), (v_0, v_2), (v_1, v_5), (v_1, v_7), (v_2, v_3), (v_2, v_4), (v_3, v_5), (v_4, v_6), (v_4, v_8), (v_5, v_6), (v_6, v_7), (v_7, v_8)\}$. We have also fixed the source s in the vertex v_0 and the terminal t in the vertex v_8.

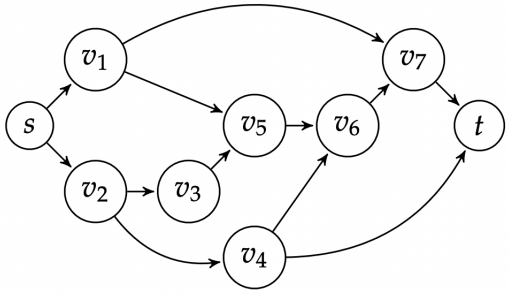


FIGURE 3.14: Directed graph of 9 vertices

Visually, we can identify the set of vertex separators defining the minimal cuts : $Cs = \{C_1 = \{v_1, v_2\}, C_2 = \{v_2, v_7\}, C_3 = \{v_4, v_7\}, C_4 = \{v_1, v_4, v_5\}, C_5 = \{v_1, v_3, v_4\}, C_6 = \{v_1, v_4, v_6\}\}$, with each subset $C_i, i = 1, \ldots, 6$ is a vertex separators of the graph with a minimal number of vertices. In the following, we want to find this set of minimal cuts Cs by our quantum algorithm. To do this, we use IBM's quantum simulator to show the intermediate results of our algorithm. In the graph G we have $n = 9$ vertices, to represent all the possible combinations of these vertices we use 9 qubits. The source s is represented by the first qubit $|v_0\rangle$ and each vertex $v_i, i = 1, \ldots, 7$ is associated to the qubit $|v_i\rangle, i = 1, \ldots, 7$ and the sink t is associated to the last qubit $|v_8\rangle$. These 9 qubits $|v_i\rangle, i = 0, \ldots, i = 9$ can represent 2^9 possibles states, therefore, these qubits can represent the superposition $|v_8 v_7 v_6 v_5 v_4 v_3 v_2 v_1 v_0\rangle = \sum_{i=0}^{2^9-1} \alpha_i |C_i\rangle$, which represent all possible subsets of vertices of the graph G. Each state $|C_i\rangle$ in the superposition $|v_8 v_7 v_6 v_5 v_4 v_3 v_2 v_1 v_0\rangle$ represent a subset of vertex, we say the vertex v_i belongs to the subset $|C_i\rangle$ (or the vertex separator $|C_i\rangle$) if the corresponding qubit $|v_i\rangle$ in the state $|C_i\rangle$ is in the state $|1\rangle$. For example, the vertex separator $C = \{v_1, v_2\}$ can be encoded by the quantum state $|000000110\rangle$.

Suppose now that all the successors of the source $s = v_0$ are down, then there is no other way to go to the next vertices, so the subset of the successors of the vertex $s = v_0$ is a vertex separator. In addition, if one of these successors is in good condition, we will find a way to go to the next vertices. Therefore, the subset of successors of $s = v_0$ is a vertex separator with a minimal number of vertices, in other words, a minimal cut. Then, to find this first minimal cut, we apply the first oracle of the movement O_s on the state $|\psi_0\rangle$. It is represented in Figure 3.15, this oracle starts by

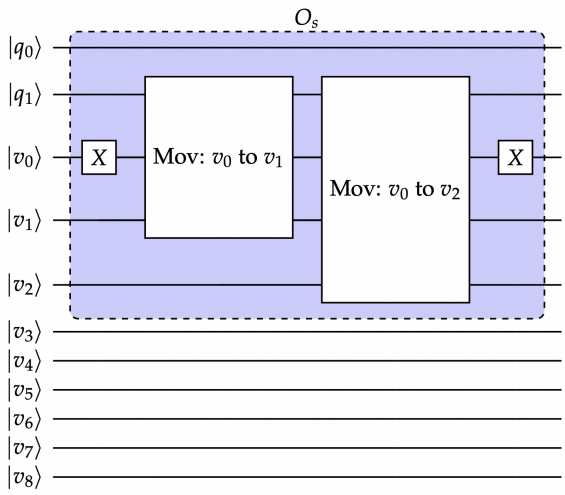


FIGURE 3.15: The Oracle of the movement of s towards the two successors v_1 and v_2.

applying the gate X on the qubit $|v_0\rangle$ that represents the source of the graph s, this gate allows to make it in the state $|1\rangle$ in order to represent the presence of the source s in the next operations. After that, it allows to apply two sub-circuits of movement towards the successors as shown in Figure 3.15, the first sub-circuit "Mov: v_0 to v_1", applied the movement towards the first successor v_1 of the source s, and the second one "Mov: v_0 to v_1" for the movement towards the second successors v_2. It means that the first sub-circuit allows to make the qubit $|v_1\rangle$ in the state $|1\rangle$ and the second to make the qubit $|v_2\rangle$ in the state $|1\rangle$. Finally, it applies the X gate on the qubit v_0 to remove the source s from the new state.

That is, we take $|\psi_0\rangle$ as the input state and apply the motion of the vertex $v_0 = s$ to its successors v_1 and v_2, as in the Figure 3.16.

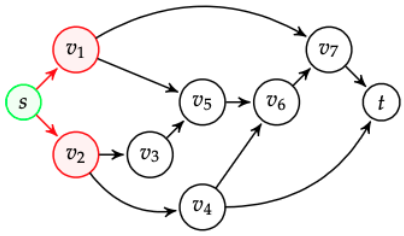


FIGURE 3.16: The movement of s towards the two successors v_1 and v_2

Applying the oracle O_s on the state $|\psi_0\rangle$ gives us the output state $|\psi_1\rangle = |000000110\rangle$ with the probability 1 (see Figure 3.17), which represents the first minimal cut $\{v_1, v_2\}$.

Now, suppose that one of the successors v_1 and v_2 is in a good condition, then there is a way to go to the following vertices. For example, if vertex v_1 is in a good condition we can find a path to the terminal through the successors of v_1. If these successors of v_1 are out of order, we cannot find a path to the terminal. Therefore, the subset $Succ(v_1) \cup C_1 \setminus \{v_1\}$ is a cut. So, if we apply the movement of v_1 in the state $|\psi_1\rangle$, we find a new minimal cut containing the successors of v_1 and the vertex v_2. The oracle that allows to apply the movement of v_1 is represented in Figure 3.18.

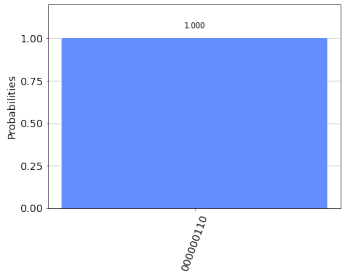


FIGURE 3.17: The result of the execution gives the state $|000000110\rangle$ which represents the first minimal cut $\{v_1, v_2\}$.

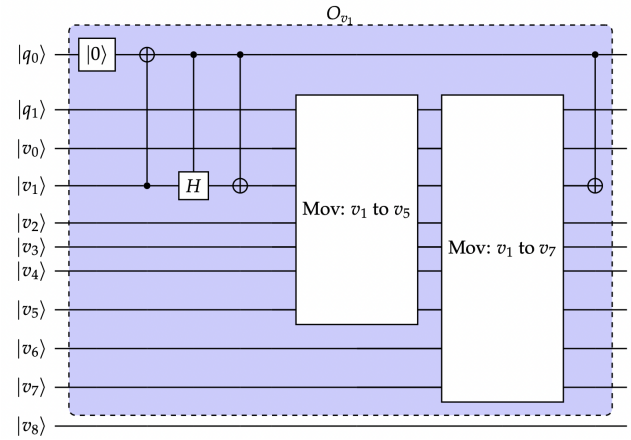


FIGURE 3.18: The Oracle of the movement of v_1 towards the two successors v_5 and v_7.

The oracle uses the Hadamard gate to keep the first cut and add the new state corresponding to the new cut. The Figure 3.19 shows the results of the execution of the oracle O_{v_1}, on the input $|\psi_1\rangle$. The Figure 3.20 shows the two cuts found until this step.

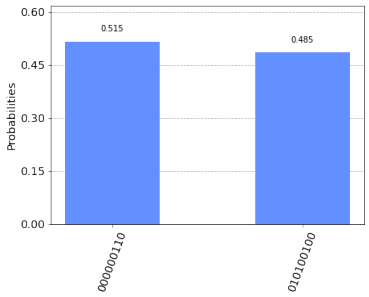


FIGURE 3.19: The result of the run gives two states: $|000000110\rangle$ represents the input and $|010100100\rangle$ represents the new cut after the movement.

At step k, we apply the oracle O_k on the output superposition of step $k-1$, therefore, we apply the movement of the vertex v_k corresponding to the oracle O_k, which adds new states in the superposition $|\psi_k\rangle$, if the qubit corresponding to the vertex v_k is in

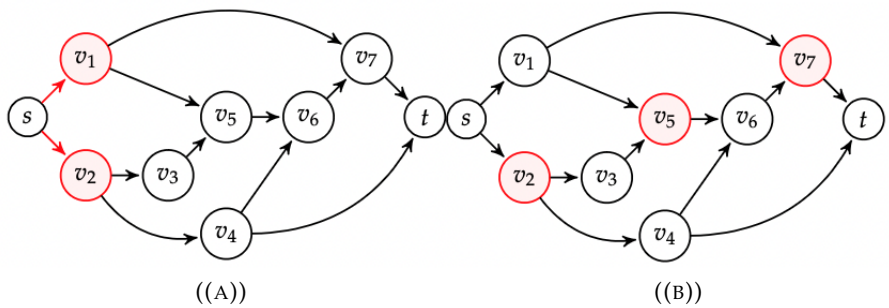


FIGURE 3.20: (A) cut $\{v_1, v_2\}$, (B) cut $\{v_2, v_5, v_7\}$

the state $|1\rangle$ for each state of the superposition $|\psi_{k-1}\rangle$.

$$\begin{aligned} |\psi_k\rangle &= O_k |\psi_{k-1}\rangle \\ &= O_k \sum_j \alpha_j |C_j\rangle = \sum_j \alpha_j O_k |C_j\rangle \\ &= \sum_j \beta_j |C_j\rangle + \sum_j \beta_j Mov_{v_k}(|C_j\rangle) \end{aligned}$$

with $\sum_j \alpha_j = 1$ and $\sum_j \beta_j = 1$.

The general circuit that represents all these oracles is shown in Figure 3.21:

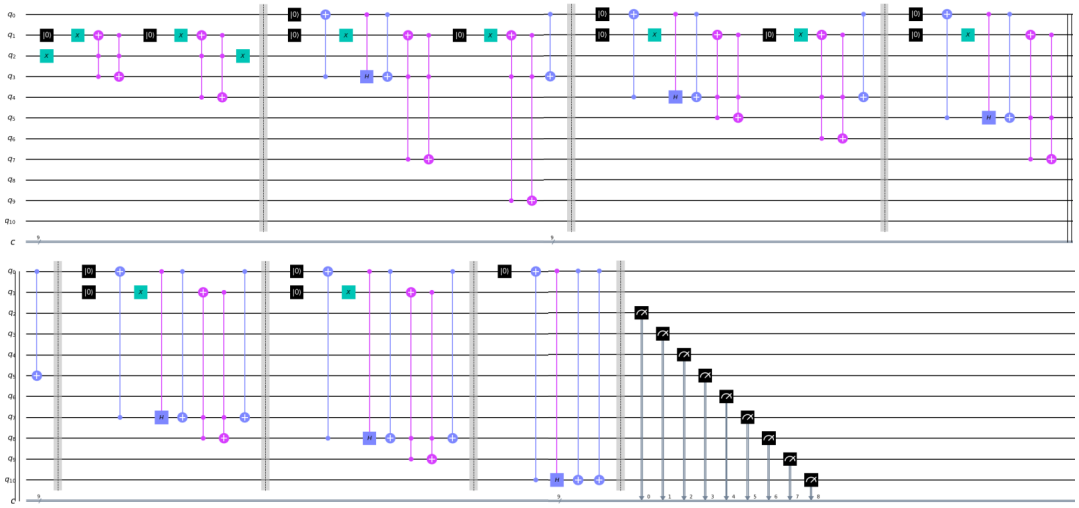


FIGURE 3.21: The circuit uses 11 qubits: 9 to represent all possible subsets of vertices, 2 for the control. And also it uses 7 movement oracles separated by vertical separators. Each oracle represents the movement of a vertex. At the end of the circuit, we measure the 9 qubits to find the superposition which represents all the minimal cuts.

After all the possible movements, we found the superposition $|\psi_{final}\rangle$.

$$|\psi_{final}\rangle = \sum_i \alpha_i |v_i\rangle = \sum_i \alpha_i |v_{i_8} v_{i_7} v_{i_6} v_{i_5} v_{i_4} v_{i_3} v_{i_2} v_{i_1} v_{i_0}\rangle \tag{3.9}$$

where $\sum_i \alpha_i = 1$ and each state $|i\rangle = |v_{i8} v_{i7} v_{i6} v_{i5} v_{i4} v_{i3} v_{i2} v_{i1} v_{i0} c_{i_1} c_{i_0}\rangle$ of the state $|\psi_{final}\rangle$ represents a cut, with $|v_{ij}\rangle = |1\rangle$ if the vertex j is in the cut i and $|v_{ij}\rangle = |0\rangle$ in the

otherwise.

In our graph example, after the execution of the circuit 3.21 in IBM's simulator and the quantum computer IBM Q 16 Melbourne, we present the results in the Figure 3.22.

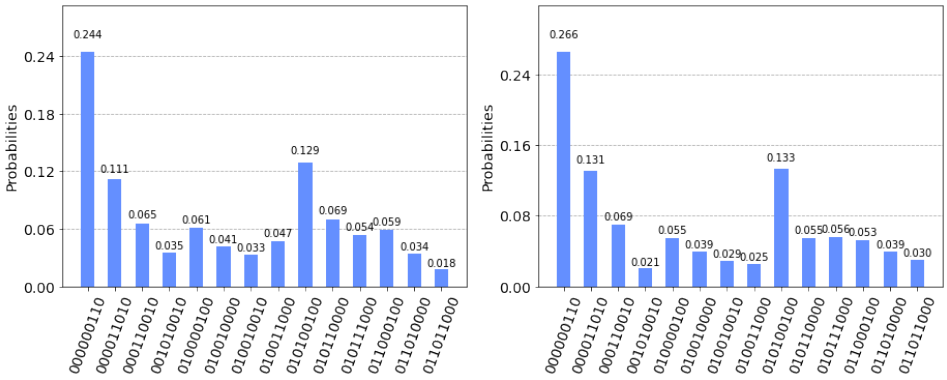


FIGURE 3.22: The histograms represent the superposition of the output $|\psi_{final}\rangle$. (a) The results of the execution in IBM's Qasm simulator. (b) The results of the execution in IBM Q 16 Melbourne quantum computer.

Finally, we remove the non-minimal cuts. For this, for each (i, j) we eliminate the cut C_j if $C_i \subset C_j$. To verify the results, we visualized each state of the superposition 3.22 in an independent graph 3.23, with red color if the vertex qubit in the state $|1\rangle$ (present in the minimal cut) and black if it's in the state $|0\rangle$.

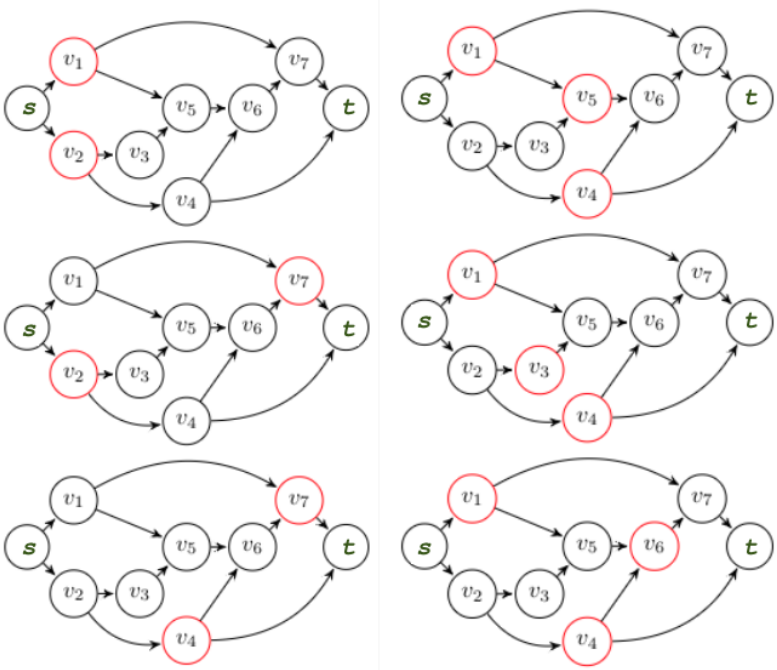


FIGURE 3.23: The set of all minimal cuts found. In each graph, the cut is represented by the vertices in red.

3.4 Conclusion

We have proposed a quantum algorithm to find all the minimal cuts of a directed graph. More precisely, we propose a quantum algorithm that uses movement oracles to generate as output a superposition of all states that represent the minimal cuts. In this chapter, cuts are represented by a set of vertices, which can separate the source and the terminal of the graph, and they are minimal if they contain just the minimal number of vertices to represent a cut. Also, the complexity of our algorithm is linear, because: it uses only $n + 2$ qubits, n to represent all the possible combinations of vertices and 2 for the counter, and it uses n oracles of movements, n being the number of vertices of the graph.

Chapter 4

Quantum Approaches for Sequence Processing

One of the main challenges in the field of Probabilistic Safety Assessment (PSA) resides in the search for all possible failure scenarios of the system components. This research helps to calculate the probability of the reliability of the system and also through this investigation, we find the weakness of our installation. In this chapter, we approach this challenge in a quantum manner, we approach it through the answer to three main questions:

1. How we can find all possible failure scenarios of a system from an initial state?
2. How can we calculate the similarity between these failure scenarios and how we can classify them?
3. If the initial state of the system changes, how we can find the most probable scenarios from the new state pending the search of all possible scenarios?

In order to answer the three questions in this chapter, we will represent the states of the system by a directed graph, each vertex of this graph represents a state of the system and each edge represents one of two actions: "repair" or "failure" of a component of the system. The critical state is represented in the graph by a marked vertex.

In order to find the scenarios that can be the cause to reach this critical state, we look for all the paths between the vertex that represents the current state in the graph and the marked vertex (the critical state). We explore graphs, by using quantum walks [ADZ93], which is a quantum strategy that reduces the complexity of exploring the graph and also doesn't let luck influence the choice of finding or not the desired vertex as in the case of random walks. Thanks to quantum walks, we can traverse the graphs in a parallel way and approach all possible paths at the same time. We propose our quantum walk-based approach to find all the paths between a source vertex and all the marked vertices in a graph. We also propose our hybrid approach to handle large graphs with the number of qubits available in the used quantum computer. In this way, we can address the challenge of finding the failure scenarios of a huge system. These two approaches that we proposed are tested and compared with the classical random walk approach on 6 graphs of different sizes.

For the second question, calculating the distance between two time series is a well-known challenge in machine learning domains such as in natural language processing and also for studying failure scenarios in nuclear power plants. The best technique to compute this distance is Dynamic Time Warping (DTW), this method has been used in the case where we process sequences element by element and also in the

case of processing sub-sequences. In the case of sub-sequences, the complexity of the problem is NP-complete which makes the processing of large sequences with this technique very difficult. Therefore, it is very important to provide an algorithm that can find results with less complexity or more speedup. So, we propose two quantum approaches for the DTW algorithm, the first algorithm as in the classical case we process the sequences element by element, and in the second, we process the case of sub-sequences in each sequence. We are the first to propose this algorithm in the field of quantum computing with the processing of sub-sequences. We evaluate both algorithms according to the results and we compare them with the existing approaches in the state of the art.

For question 3, in order to create a generative model we will use the Hidden Quantum Markov Models (HQMMs). Therefore, we will study and compare the results of HQMMs and classical Hidden Markov Models HMM on real datasets generated from real small systems in the field of PSA. As a quality metric, we will use Description accuracy DA and we will show that the quantum approach gives better results compared with the classical approach, and we will give a strategy to identify the probable and no-probable failure scenarios of a system.

4.1 Research of the failure scenarios of a system

In data science, finding the paths between two vertices in a graph is a very important problem. It is used in almost every field, such as transportation, chemistry, computer networks, etc. The problem of finding all paths between two vertices in a Directed A-cyclic Graph (DAG) is known as NP-complete [Val79]. This problem can be solved by many algorithms like the BFS algorithm [DAGa] in which we use a backup method at each step where we find the searched vertex, this method can be improved with the backtracking algorithm [Cat15], also, using dynamic programming [Man15],[DAGb].

Today with the explosion of information and also with the growth of datasets, the use of these algorithms becomes an obstacle to find results faster. As in the field of PSA, where we seek to find scenarios of failures of a system through a graph of states. The use of these algorithms is a very great weakness to our strategy in order to analyze the safety of our installation.

One of the main obstacles for a dynamic PSA is the difficulty to envisage an evolution of the state of a system in time and its transitions with stochastic failures. One of the main difficulties is the modeling of the different transitions because of the combinatorial explosion and the exploration of the different sequences in an exhaustive way. The multiple bifurcations of the sequences lead to very long exploration times for industrial exploitation. This is the reason why approximations are given up so that the exploration algorithm can finish. For example, if we want to model the transitions of a system of two redundant components A and B. We obtain the graph in Figure 4.1. With 3 components we obtain the graph of the Figure 4.2.

We can see the combinatorial explosion of the transitions that result from this. The use of classical algorithms does not allow to do the computation in a reasonable time and reduces the interest of these methods in the context of reliability or safety computation, or dynamic sequence exploration. To this end, we propose in the following a quantum approach to solve the problem of finding all paths between two vertices in

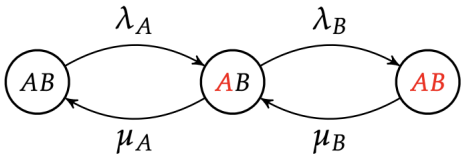


FIGURE 4.1: The graph of the transitions of the states of a graph of two redundant components

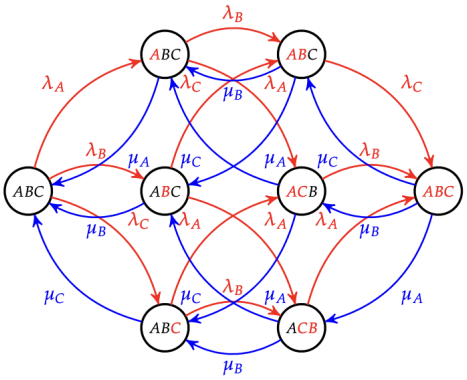


FIGURE 4.2: The graph of the transitions of the states of a graph of three redundant components

a DAG.

As we have explained in the section 2.8.5, with the concept of quantum walks we can propagate a graph in a parallel and fast way, and at the end we find the probability of reaching each searched vertex of the graph. In addition to that, with the concept of parallelism, we have the guarantee that we have processed all the paths between the source and the desired destinations. On the other hand, with this approach, we do not find the history of each path, that is to say, we do not find the order of the vertices that we have crossed for each path. For the problem that we are trying to solve, it is necessary to find all these paths with the order of the vertices in each path, as well as the probability of each one, where the probability is not only based on the configuration of the graph but rather it is based on the weights of the edges of it. So the questions that arise are:

1. How can we find the order of the vertices in each processed path?
2. How can we take the weights of the graph into account to find the exact probability of each path found?

To answer these two questions, in the following subsection, we suggest our approach based on the philosophy of quantum walks. We propose a new configuration of qubits to find the order of vertices in each path and we propose our quantum oracle which allows us to advance in each path according to the weights in the graph.

4.1.1 Quantum approach to find paths in a directed a-cyclic graph

Let $G = (\mathbf{V}, \mathbb{E}, \mathbf{C})$ be a graph of n vertices $\mathbf{V} = \{v_1, \ldots, v_n\}$ connected by edges defined in the set $\mathbb{E}$, where each $e_{i,j} \in \mathbb{E}$ represents the connection from the vertices v_i to v_j and has a weight $P_{i,j}$. C is the set of marked vertices in the graph.

A path λ in the graph is a succession of steps in the graph that reach a marked vertex $c \in \mathbf{C}$, where the probability of λ is calculated as follows:

$$P(\lambda) = \prod_{e_{i,j} \in \lambda} P_{i,j} \tag{4.1}$$

The objective of this subsection is to find all paths λ from a given vertex to some marked vertices using quantum walks. For that, in order to create the space in which this quantum walks pass, we represent the vertices of the graph by qubits, and we build with the help of these qubits a superposition containing all paths λ from the source of the graph v_0 to all marked vertices $c \in \mathbf{C}$.

In order to apply quantum walks and to store the steps of the walks, we propose our approach based on a path saving strategy using for each vertex of the graph a qubit and each state of these qubits is a path in the graph.

4.1.2 How can we encode paths with quantum states?

In quantum walks 2.8.5, we can traverse the graph through all paths simultaneously and with deterministic steps, we don't miss any possibility of a path between two vertices. This strategy allows us to walk through all possible paths between two vertices, but unfortunately, we can't keep these paths and we can't extract them. So, we suggest a strategy to encode these paths in order to extract them at the end of all the steps. In what remains in this chapter, the path is encoded in the quantum state as follows:
Suppose that we have a state $|\lambda\rangle = |v_0 v_1 \dots v_n\rangle$ which represents the path λ.

- We say that the vertex v_i is in the path if and only if the qubit v_i of the state $|\lambda\rangle$ is in the state $|1\rangle$.
- We say that the edge $e_{i,j} = (v_i, v_j)$ is in the path if and only if the two qubits v_i and v_j are in states $|1\rangle$ and $|1\rangle$ respectively and the qubits v_k for $i < k < j$ are all in state $|0\rangle$.

With this encoding of vertices using qubits, we can keep the history of each path. Take the example of Figure 4.3, in it, we represent 3 different paths, each path is represented by a colo,r and the quantum state that represents it is also represented with the same color.
The first path in red $e_0 \to e_3 \to e_{11} \to e_{12}$ is represented by the state $|1001000000011\rangle$, where the qubit that represents e_0 is in state $|1\rangle$, after that, we have the second element is e_3, then we find the two qubits that represent e_1 and e_2 in states $|0\rangle$, $|0\rangle$ respectively, and the qubit e_3 is in the state $|1\rangle$. Then, the qubits from e_4 to e_{10} are in state $|0\rangle^{\otimes 7}$ and the two qubits e_{11} and e_{12} are in state $|11\rangle$. For the two paths, blue and green, we find the qubits that represent the elements of paths in state $|1\rangle$ and the others in state $|0\rangle$.

The two principal differences between our approach and that of the quantum walks represented in the section 2.8.5 are:

- with our approach we can find the order of the vertices in the paths and with the approach presented in 2.8.5 we can't;

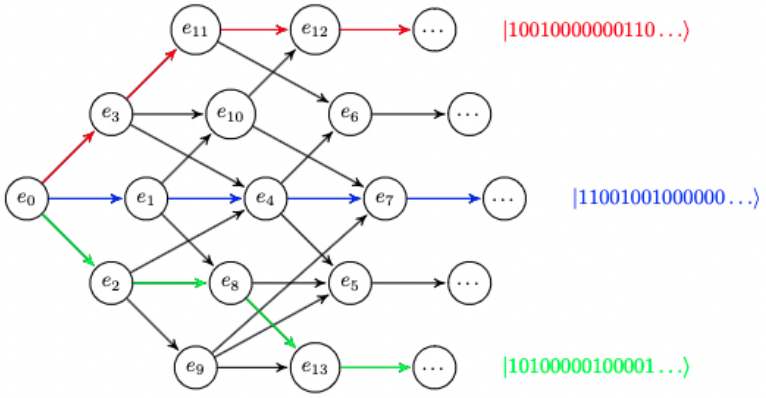


FIGURE 4.3: The representation of paths with quantum states

- the second difference is in the number of used qubits, with our approach for a graph with N vertices we use N qubits versus with the approach presented in 2.8.5 we use only $log_2 N$ qubits.

4.1.3 How can we manage loops in the graph?

The configuration that we propose to encode paths doesn't allow us to handle the loops in graphs, which can cause some problems if we have graphs with loops. The solution that we can use, is to remove these loops and change the structure of the graph to keep the possibility of having a single loop configured with more vertices. In the PSA field, assuming that a component can be repaired only once in a scenario. So, to remove these loops and improve the structure of the graph, we propose this small transformation: Consider a graph $G = (\mathbf{V}, \mathbb{E}, \mathbf{C})$, if $(v_i, v_j) \in \mathbb{E}$ and $(v_j, v_i) \in \mathbb{E}$ (there is a loop between v_i and v_j).

1. We remove the edge (v_j, v_i) from the set $\mathbb{E}$;
2. we add a new vertex v_i^* in the set of the vertices V;
3. we add the edge (v_j, v_i^*) in E and also we add the edges $(v_i^*, v_l), \forall v_l \in Succ(v_i) \setminus v_j$. Where $Succ(v_i)$ is the set of successors of v_i.

With the removal of the edge (v_j, v_i) from the set $\mathbb{E}$, we have ensured the removal of the loop, and adding the vertex v_i^* and the edges (v_j, v_i^*) and $(v_i^*, v_l), \forall v_l \in Succ(v_i) \setminus v_j$ makes it possible to keep the loop only one time on the paths of the graph, where the loop $v_i \longrightarrow v_j \longrightarrow v_i$ can be represented in the new graph structure by $v_i \longrightarrow v_j \longrightarrow v_i^*$. In the Figure 4.4, in order to remove the loop between v_i and v_j we add the vertex v_i^* and the edges (v_j, v_i^*), (v_i^*, v_{j+1}) and (v_i^*, v_{j-1}).

4.1.4 How we can use the weights of the graph?

The question that remains now is how we can apply quantum walks in a weighted graph and with our configuration? To answer this question, we begin by defining exactly the challenge that we are addressing.

Let us suppose that we have done t steps (we have already done t walks in the graph) and that we want to go to the step $t + 1$. Therefore, there are a number of

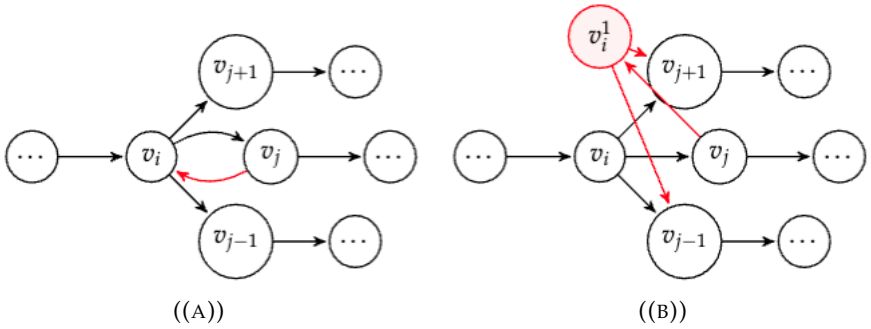


FIGURE 4.4: Example of the deletion of a loop

paths, we suppose k paths, $\lambda_i, i = 0, \ldots, k$. The goal is then to advance in the paths at the time t, it is necessary to advance towards the successors of each end of paths (the last summit of the path). Let's take the example of Figure 4.5 where we show an example of a path at time t.

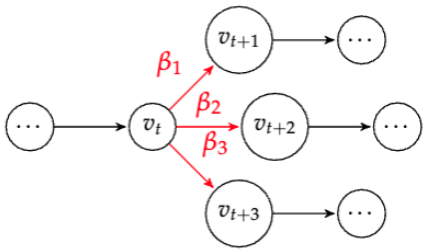


FIGURE 4.5: Example of processing a path at time t

Suppose that this path is represented by the state $|\lambda\rangle = |v_0 \ldots v_t\rangle \otimes |v_{t+1}v_{t+2}v_{t+3}\rangle \otimes |v_{t+k+1} \ldots v_n\rangle$. At the time t we have already successfully built the path to v_t and have nothing after v_t. Therefore the quantum state that represents the path at time t is $|\lambda\rangle = |v_0 \ldots v_t\rangle \otimes |000\rangle \otimes |0 \ldots 0\rangle$. The vertex v_t has three successors, so if we walk through the three successors we get 3 different paths, where each path contains, in addition, one of the 3 successors. So, quanticaly speaking, from the state $|\lambda\rangle$ we will extract 3 different states, where each state has a probability β_i. Formally, we are looking for an oracle O that allows us to perform the following transformations:

$$|\lambda'\rangle = O\,|\lambda\rangle \tag{4.2}$$

$$= \frac{\beta_1}{\beta_1+\beta_2+\beta_3}\,|\lambda_1\rangle + \frac{\beta_2}{\beta_1+\beta_2+\beta_3}\,|\lambda_2\rangle + \frac{\beta_3}{\beta_1+\beta_2+\beta_3}\,|\lambda_3\rangle \tag{4.3}$$

with the three qubits $|v_{t+1}\rangle$,$|v_{t+2}\rangle$ and $|v_{t+3}\rangle$ are in the state $|1\rangle$ respective in the states $|\lambda_1\rangle$,$|\lambda_2\rangle$ and $|\lambda_3\rangle$ as follows:

$$|\lambda_1\rangle = |v_0 \ldots v_t\rangle \otimes |100\rangle \otimes |0 \ldots 0\rangle \tag{4.4}$$

$$|\lambda_2\rangle = |v_0 \ldots v_t\rangle \otimes |010\rangle \otimes |0 \ldots 0\rangle \tag{4.5}$$

$$|\lambda_3\rangle = |v_0 \ldots v_t\rangle \otimes |001\rangle \otimes |0 \ldots 0\rangle \tag{4.6}$$

In quantum walks, we use the Hadamard gate to advance through the graph, the problem with using this gate is that we can't specify the probability for each step of progress and we can't deal with the case where vertices have 3 or more successors.

Then, to specify the probabilities and to handle the case of more than two successors for each vertex of the graph, we define the general quantum gate $U(\theta, \phi, \lambda)$. This gate is represented by the following matrix:

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2) \end{pmatrix} \tag{4.7}$$

with the three parameters θ, ϕ and λ, we can make any rotation of $|\psi\rangle$ with respect to x, y and z axis. If we set ϕ to 0 and λ to 0 we find the matrix:

$$U(\theta) = U(\theta, 0, 0) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \tag{4.8}$$

With this matrix, if we are in the state $|0\rangle$, we can exactly specify the probability of shifting to the state $|1\rangle$ with the θ rotation angle as follows:

$$U(\theta)|0\rangle = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos(\theta/2) \\ \sin(\theta/2) \end{pmatrix} \tag{4.9}$$

We use the $U(\theta)$ gate and we apply the following sub-circuit 4.6 to find the state $|\lambda'\rangle$. In this circuit, we take the qubit $|v_t\rangle$ as a control to apply the walk to all the paths that have reached the vertex v_t, the first gate $U(\beta_1)$ applies the walk to the vertex v_{t+1} according to the weight β_1 of the edge between v_t and v_{t+1}. After that, we apply the gate X on the qubits v_{t+1} and we use it also as a control to apply the walk to the second successor v_{t+2} and similarly for the third successor v_{t+3}. At the end, we apply the two X gates on the two qubits v_{t+1} and v_{t+2} to return to the state after the walk.

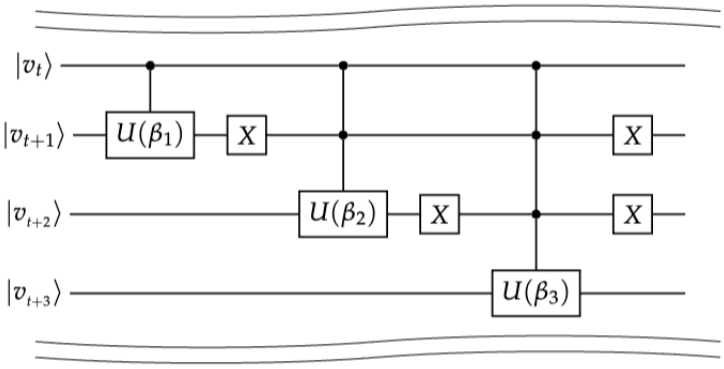


FIGURE 4.6: Sub-circuit to generate the state $|\lambda'\rangle$

4.1.5 Quantum Oracle for quantum walks

In the general case, suppose that we have k vertices $\{v_t, \ldots v_{t+k}\}$ and each $v_i, i \in \{t, \ldots, k\}$ has the successors $v_i^s, s \in N$ and each edge $e = (v_i, v_i^s)$ has a probability

$\beta_{(v_i,v_i^s)}$, the oracle is defined as follows 4.7:

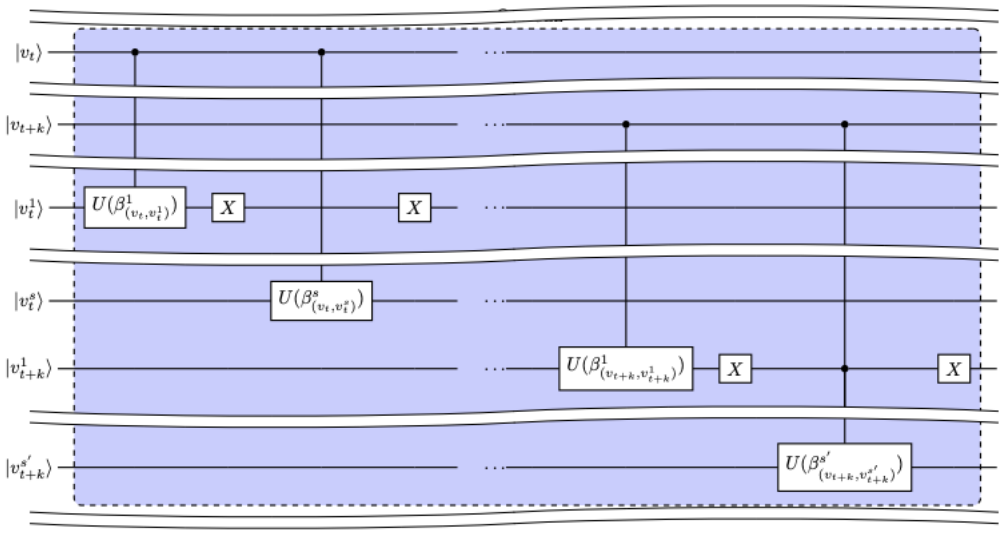


FIGURE 4.7: The architecture of our oracle

This quantum oracle takes k inputs, and it allows to apply the walks to all the successors of each input while keeping the memory of each path.

4.1.6 Quantum algorithm to obtain all paths in a directed a-cyclic graph

We use the general purpose oracle in Figure 4.7 to introduce the algorithm 5, it allows us to find all the paths between a source and all the marked vertices in a DAG.

Algorithm 5: Quantum algorithm to obtain all paths in a DAG (QAPAG)

Input : Graph $G = (\mathbf{V}, \mathbb{E}, \mathbf{C})$, source v_0.
Initialization
Initialize $n = |V|$ qubits in the state $|0\rangle^{\otimes n}$.
Apply the gate X into the qubit $|v_0\rangle$.
Initialize the set of steps to be handled M with the initial state $M = \{v_0\}$
while $M \neq \emptyset$ **do**
  Apply the oracle $O(M, \{Succ(v_t), \forall v_t \in M\})$.
  Empty M, $M = \{\}$.
  For each $v_i \in Succ(v_t)$ add v_i to M if $Succ(v_i) \neq \emptyset$ and $v_i \notin C$.
end
Measure the circuit and extract the set of paths P_s.
Return : P_s

The algorithm 5 takes as input the graph $G = (\mathbf{V}, \mathbb{E}, \mathbf{C})$, and the source v_0, with $\mathbf{V}$ is the set of vertices, $\mathbb{E}$ the set of edges, and $\mathbf{C}$ the set of marked vertices in the graph. We start by initializing n qubits to the state $|0\rangle^{\otimes n}$, where $n = |\mathbf{V}|$ is the number of vertices. We apply the gate X on the first qubit in order to indicate the source of the graph, and we initialize the list M by the source v_0.

After the initialization step, we apply the quantum walks with our configuration and

the oracle that we have presented previously. In each iteration, we go to the successors of each element of M in order to add a step in each path toward all successors. In this step, for each path λ_i and its last step v_t at time t, we will add k (k is the number of successors of v_t) similar paths up to step t, and in each path of its, we will add a successor.

With the quantum oracle, just calling the gate that allows the target qubits to rotate by a specified angle around the base axes allows us to create and add these steps. After that, we remove all the elements of M and we add the new vertices of the instant $t+1$ if they are not part of the set of the marked vertices $\mathbf{C}$. We repeat these three iterations as long as the set M is different from the empty set.

At the end, we measure the circuit to find the superposition that contains all the states that represent the paths. For each state of this superposition, we extract the corresponding path.

We take for example the graph of the Figure 4.8, with the circuit 4.9 we can apply all the possible steps in the graph to attract the marked vertices. In the graph, we show with the colors of each oracle the step that applies.

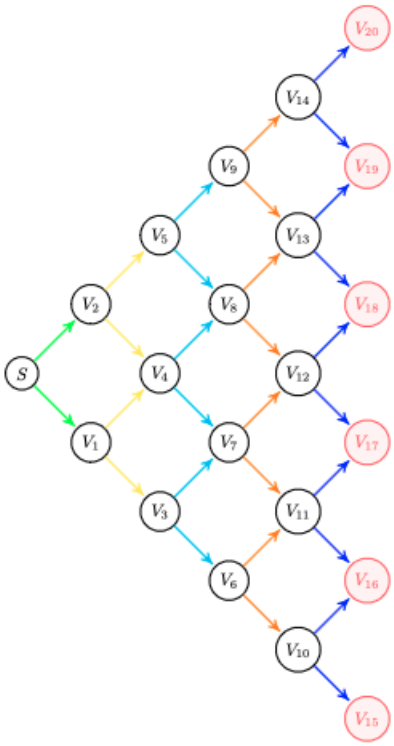


FIGURE 4.8: Example of a quantum walks

Each oracle M_i represents a specific case of the general circuit that we have shown in Figure 4.7. In the circuit of the Figure 4.9, we see that the oracle M_1 takes only the source S and allows to go to the two qubits v_1 and v_2 in order to apply the rotation that we have shown with the gate $U(\theta)$. The second oracle takes as inputs the two output qubits of the first oracle and the third takes the outputs of the second oracle and so on, until the end. By measuring the circuit we find the superposition that includes all the paths between the source S and all the red vertices of the graph 4.8.

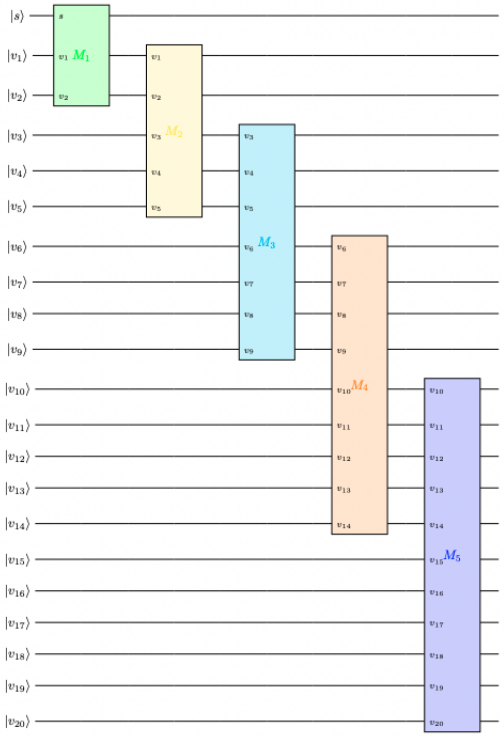


FIGURE 4.9: The general circuit to find the warping path between two sequences

4.1.7 Complexity analysis

For the memory complexity, for the graph $G = (\mathbf{V}, \mathbb{E}, \mathbf{C})$, we use $N = |\mathbf{V}|$ qubits (for each vertex we use a single qubit). For the computational complexity, we use at most M Multi control $U(\beta)$ ($Mc - U(\beta)$) gates, $M = |\mathbb{E}|$ is the number of edges. These gates are used to build oracles, they are a combination of gates with a single control qubit and M_s control qubits, where M_s is the maximum number of successors of the vertices of the graph. Also we use at most $2M$ X gates.

4.1.8 Hybrid approach to find paths in an a-cyclic directed graph

The biggest limitation of the above approach is that it requires a lot of qubits compared to the latter one, the approach that we have presented in the section 2.8.5. For us, we need N qubits instead of $Log_2 N$ of the previous approach. In addition to that, the number of qubits currently available in quantum computers or quantum simulators is very small, which makes the hybridization of our approach even more necessary. The hybridization process means that we should find an approach that allows us to use our algorithm in a way that benefits from the advantage of quantum computers according to the number of qubits available today for large problems. We divide the large circuit into several small circuits using a classical computer, we run these small circuits one by one on a quantum computer, and we aggregate the results in a classical way.

In our case, we need to divide the large circuit generated by the algorithm 5 into several smaller circuits and we process them separately, then we combine the results at the end. The questions that should be asked are:

- How we can divide the large circuit?
- How we can aggregate their results?

To answer these two questions, we propose our approach in this section. Let's suppose that we have a weighted graph $\mathbb{G} = (\mathbf{V}, \mathbb{E}, \mathbf{C})$, with the number of vertices as $N = |\mathbf{V}|$, and N_q is the number of qubits available in the quantum computer that we are using. Suppose that N is very large compared to the number of qubits N_q, $(N >> N_q)$. Then, finding the results with algorithm 5 only becomes impossible.

We start by defining a filter for the search, we define P_{min} as the minimal probability for the paths, i.e. we search for all the paths that have a probability greater than P_{min}. Of course, if we want to search all the paths without taking into account the probability P_{min}, we search with $P_{min} = 0$.

Now we start processing the graph $\mathbb{G}$, we begin with the source of the graph s and suppose that we have N_q qubits available in the quantum computer, we extract a graph of N_q vertices from the source s. This extraction is done as follows: we take the N_q vertices which are close to the source s. We start by adding the source and the successors of s to the set of vertices, then for each successor, we do the same thing as long as we have the number of selected vertices less than N_q. The algorithm 6 allows us to do this extraction.

Algorithm 6: Extraction of a sub-graph

```
Input      : Graph 𝔾 = (V, 𝔼, C), n = |V|, N_q, source s
Initialisation
Initialize a new graph 𝔾' = (V' = ∅, 𝔼' = ∅, C' = ∅)
Initialize a list Todo = [s]
while |V'| < N_q and |Todo| > 0 do
    Take an element v from the list Todo.
    if |Succ(v)| + |V'| < N_q then
        for all element v_i in Succ(v) do
            Add v_i to V'
            Add (v, v_i) to 𝔼'
            if v_i ∈ C then
                Add v_i to C'
            end
            Add v_i to the Todo list if it doesn't already exist
        end
    end
    Delete v from Todo list
end
Return     :𝔾' = (V', 𝔼', C')
```

Then we use the algorithm 5 of the previous approach to find all the paths of this sub-graph. From the results obtained after this first step, we eliminate the paths that have a probability lower than P_{min}, and we add to the result list the paths that have a probability higher than P_{min} and that arrived at a marked vertex $c_i \in \mathbf{C}$. For the rest of the paths, i.e., the paths that have a probability higher than P_{min} and they haven't yet arrived at a marked vertex (the current paths), we take the end of each path among these current paths and extract a sub-graph from each item of these vertices, and

use the algorithm 5 with the first oracle of it take all these output vertices as inputs. Of course, if the number of qubits allows us to do these tasks at the same time, if not, we can process all the outputs individually. And we do this again until all the paths arrive at the marked vertices or the end of the graph. This whole process is summarized in the algorithm 7.

Algorithm 7: Hybrid algorithm to obtain all paths in a DAG (HQAPAG)

Input : A DAG $\mathbb{G} = (\mathbf{V}, \mathbb{E}, \mathbf{C})$, source v_0, number of qubits available in the quantum computer N_q, and the minimal probability of the paths P_{min}.

Initialisation

Initialize the set of walks to be processed M with the source $M = \{v_0\}$

Initializes the set of paths that arrived at a given vertex marked at $P_g = \emptyset$ and the current paths at $P_t = \emptyset$.

while $M \neq \emptyset$ **do**

  v_t is the first item of M

  Extract a sub-graph $\mathbb{G}_i$ of $\mathbb{G}$ from v_t such that the number of vertices of $\mathbb{G}_i$ is less than N_q.

  Extract the paths Ps_i from $\mathbb{G}_i$ using the algorithm 5

  Calculate the exact probability of each path

  Remove the item v_t from the list M

  Update the two path sets P_g,P_t and the set M with the new list of found paths Ps_i according to P_{min}.

end

Return : P_g

In the algorithm 7, we proposed to use for each iteration a single source since the number of qubits today in quantum computers is very small. If the number of qubits is a little bit larger, we can simply process all the elements of the list M at the same time in each step. In order to make it clear, we show the circuit in the general case of a large graph in Figure 4.10.

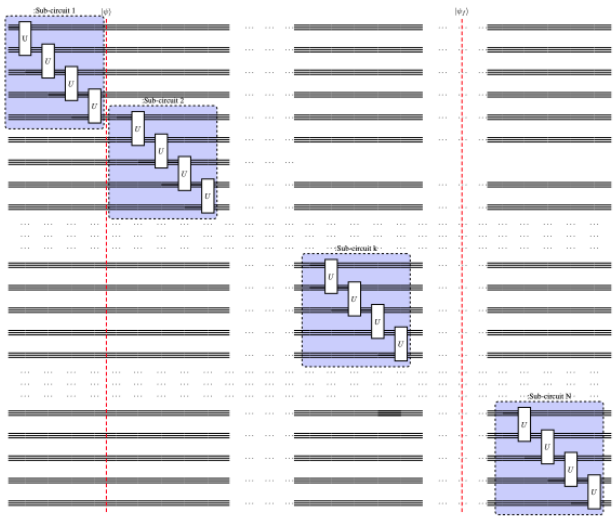


FIGURE 4.10: Subdivision of the big circuit for a big graph

Each box marked in blue in the circuit 4.10 is an iteration of our algorithm 7, where each box takes no more than N_q qubits as input, in the circuit $|\psi\rangle$ is the superposition found after the first step and $|\psi_f\rangle$ is the superposition of the execution at the last step.

With this hybrid approach, we can handle large graphs with our algorithm 5. That means, we can get all paths from the source (or any vertex) to the marked vertices in a DAG. The weights in this case play the role in the probability of each state in the superposition at the end, where we find the order of importance of each path. The exact probability of each path with our approach is calculated in a classical way along the running of our hybrid algorithm 7.

4.1.9 Results and tests

In this section, we analyze and compare the results obtained by each approach proposed in the previous section with the classical approaches.

4.1.10 Results of the approach HQAPAG

We start by comparing our proposed approach in 4.1.1 with the classical Random walk algorithm. To do this, we set the maximum running time to 20s, and the minimal probability of sequences to $P_{min} = 10^{-8}$. We have randomly generated 6 graphs of different sizes and we have chosen the marked vertices in these graphs in a random way. We have applied our algorithm and the classical algorithm to find the paths to these marked vertices and we have represented the results in the Table 4.1.

Graph	1	2	3	4	5	6
Number of vertices	50	60	70	80	100	120
Number of edges	235	285	335	385	485	585
NP founded by RW	2060	5182	6682	7726	8762	9405
NP founded by our approach	2463	17134	50989	317324	389869	1475909

TABLE 4.1: Number of found paths with our approach and the Random Walk approach (NP: Number of paths)

In Table 4.1, we can see that in all 6 cases, our approach finds more paths than the classical approach. Also, when we increase the size of the graph, we find a very large number of additional paths compared to the random walk approach.

To find these results we have used the IBM quantum simulator with 32 qubits, then, used our algorithm 7 with the setting $N_q = 30$. In order to compare the time spent for each approach, we present the comparative results in Figure 4.11.

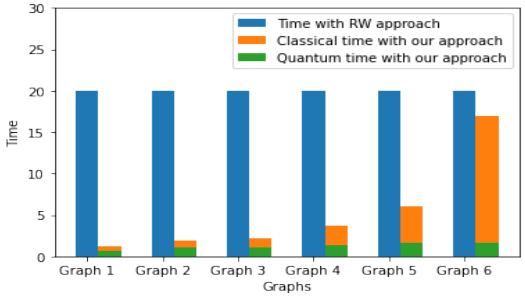
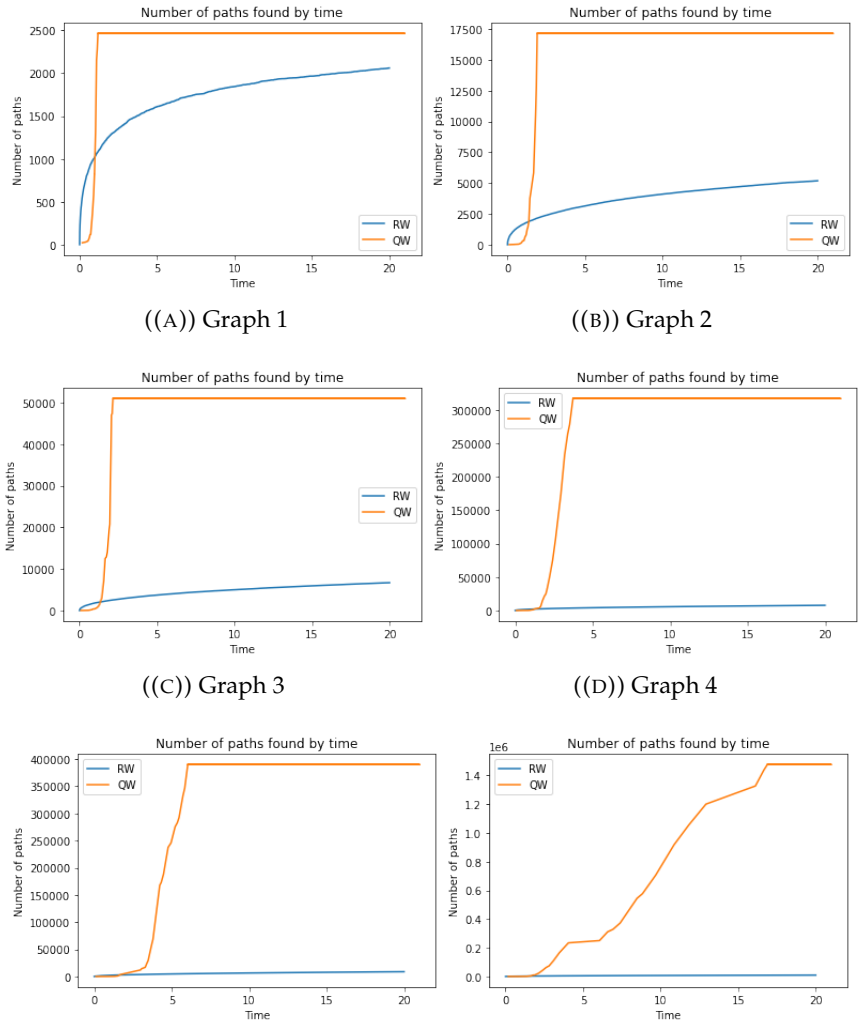


FIGURE 4.11: The running time spent for each approach

In the Figure 4.11, we can see that the classical time expects the maximum time given for each approach (20s) and that nevertheless, it doesn't succeed to find the number of paths found by our approach, as we have already shown in the Table 4.1, and for our approach, we can see that the time passed to run the quantum circuits is very small compared to the classical time, which shows that even if we use a simulator with a very small number of qubits (32 qubits) we can get better results than the classical one and our hybrid approach works perfectly well, so we can see in these results that the classical time for our approach (orange color) is increasing with the size of the graphs, which means, we find several partitions and to combine all those results to get the global results we spend too much time in the classical one. But, even with this time, we are still much less than the classical approach. This classical time spent in our approach will decrease with the increase of the number of qubits in the simulator that we use. In order to compare the functionality of our approach, we show the results of the number of paths found, according to time in the Figure 4.12.



((A)) Graph 1 ((B)) Graph 2

((C)) Graph 3 ((D)) Graph 4

((E)) Graph 5 ((F)) Graph 6

FIGURE 4.12: Number of paths founded according to time

In the Figure 4.12, for all 6 graphs, we can see that the search for paths starts with a small advance from the classical random walk, then our approach goes up very

quickly and remains stable, contrary to the classical approach which goes up very slowly all the time.

The classical approach has advanced in the first iteration because it searches the paths one by one and our approach propagates the graph in a parallel way, to look for all the paths at the same time, which requires some time to reach the marked vertices. But when our approach arrives, it arrives with a large number of paths, which implies an increase in the number of paths in this manner. In the case of our approach, the results after some time remain unchanged, which means that our approach converges toward all the possible paths to these marked vertices.

Obviously, by using a classical deterministic algorithm on these small graphs, we are better than in the quantum case, these results are just found using a simulator and the primary objective is to show the differences between classical random walks and quantum walks on our problems. The challenge of finding better results than the classical one is related to the availability of real quantum computers with a large number of qubits in order to test on large graphs.

4.2 Quantum approach to calculate the similarity between sequences

Measuring the distance between objects is one of the essential tasks in the field of machine learning. When the objects are time series, we use Dynamic Time Warping (DTW) to measure the distance between them. This method was first proposed in the 60's by R. Bellman and R. Kalaba in [BK59], and has been used extensively in the following years in the field of speech recognition [MRR80], [SC78]. Recently, it is used in [EFV07] to measure the similarity that tries to capture the "spirit" of the dynamic temporal deformation while being defined on continuous domains. [KZ07] uses DTW to classify the hands into one of the 21 possible classes of Croatian sign language. Also, it is used by [Cor01] to recognize human activity in the form of hand and arm movements from a small pre-defined vocabulary. It is used in several papers [NR07] [GJ06] [BB04] [KS01] [KSG02] [ER08] for the clustering of time series. Furthermore, in other fields such as computer vision and computer animation [Mül07a]. [Fel+20] proposes a Quadratic Unconstrained Binary Optimization (QUBO) formulation of DTW adapted to the execution on Quantum Annealing (QA) hardware. In addition to this, in many other fields [HLT11], [Bar+15], [SFY07], [Ros+17], [GHZ12], [Che+09], scientists are interested in finding the similarity between sub-sequences. The best measure of similarity between sequences is DTW [Din+08], which has been improved by Müller et al in [Mül07b] to find multiple similar sub-sequences between two sequences. To find a sequence within a large sequence we find the SPRING version [SFY07].

The overall problem considered here is to match optimally several sub-sequences of different lengths of the sequence X with several target sub-sequences of the sequence Y, taking into account that the sub-sequences of each large sequence are independent of each other and that the whole sequence is divided into sub-sequences. The sub-sequences can change their size from an iteration to another within the same sequence and the match can also be passed between two different sub-sequences of different sizes (see the Figure 4.13).

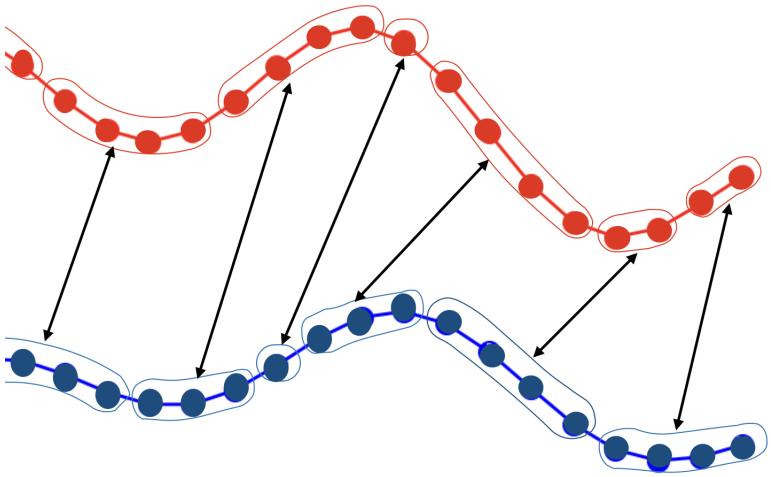


FIGURE 4.13: The matching between two sequences using sub-sequences

The total cost of alignment between the sub-sequences of X and the sub-sequences of the sequence Y is the sum of the distances between all pairs of matched elements of the sub-sequences. The distance between two elements is a domain-specific measure, such as the absolute difference between the scalars associated with these elements.

The optimal alignment is the one that finds a match between all the sub-sequences of X with the sub-sequences of Y with a minimal distance.

It has been proved by Wang and Jiang in [WJ94] that this problem is NP-complete. Moreover, it can be solved by applying DTW in an h-dimensional space (see Figure 4.14). This is an algorithm of $O(\prod_{l=1}^{h})$ operation within an exponential space with $n_1, \ldots, n_h$ are the dimensions of h sub-sequences. [PKG11] shows that this calculation is impossible in most cases when h is large.

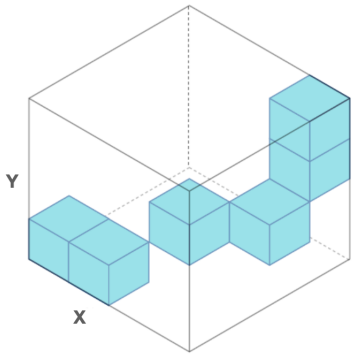


FIGURE 4.14: The problem of finding the matching between two sequences using subsequences in an h-dimensional space

On another side, quantum computing [Jae07] is a very interesting future solution for complex problems that are hard to solve with conventional calculators [Pre12]. In

order to reduce the complexity of the classical DTW, we propose our quantum DTW approach to measure the distance between two time series, we propose our algorithm to create the quantum circuit to measure this distance, and in addition to that, we demonstrate how we can deal with sub-sequences and answer our problem. We test our algorithm in the IBM quantum simulator, and we compare our results with the results found by the quantum approach in [Fel+20] and also the classical approach.

4.2.1 Classical Dynamic Time Warping

Dynamic Time Warping (DTW) is an algorithm to measure the similarity between two time sequences whose speed can vary. DTW can be applied to time sequences of video, audio, and graphic data. In fact, all data that can be transformed into a linear sequence can be analyzed with DTW. Generally, it is a method that computes an optimal match between two given sequences with some restrictions. Let's say that we have two sequences $X = \{x_1, x_2, \ldots, x_M\}$ and $Y = \{y_1, y_2, \ldots, y_N\}$, with $M, N \in \mathbb{N}$ and we want to check whether these two sequences match or not. To do this, we first construct the distance matrix $D \in \mathbb{R}^{M \times N}$ (see the Table 4.2), where each element $d_{m,n}$ represents the Euclidean distance between the two data x_m and y_n:

$$d(x_m, y_n) = ||x_m - y_n||_2 = |x_m - y_n| \tag{4.10}$$

y_m	$d_{0,m}$	$d_{1,m}$	$d_{2,m}$	$\ldots$	$d_{n,m}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
y_2	$d_{0,2}$	$d_{1,2}$	$d_{2,2}$	$\ldots$	$d_{n,2}$
y_1	$d_{0,1}$	$d_{1,1}$	$d_{2,1}$	$\ldots$	$d_{n,1}$
y_0	$d_{0,0}$	$d_{1,0}$	$d_{2,0}$	$\ldots$	$d_{n,0}$
	x_0	x_1	x_2	$\ldots$	x_n

TABLE 4.2: The distance matrix between the two sequences X and Y

This distance can be changed by any distance, and it is chosen according to the nature of the data.

The objective of the DTW algorithm is to find a Warping Path (WP) that connects the two elements $d_{1,1}$ and $d_{M,N}$ of the matrix D with a minimal distance. The WP indicates how the elements of a sequence are assigned to the elements of the other one. Mathematically, we try to find the WP: $p = (p_1, p_2, \ldots, p_L)$ with $p_l = (m_l, n_l) \in [1; M] \times [1; N]$ for $l \in [1; L]$ under the following constraints [Mül07b]:

1. The starting point of WP is $d_{1,1}$ and the end point is $d_{M,N}$.
2. The WP should be monotonous in time. A backward movement, i.e. to the left or downward, is not possible:

$$n_1 \leq n_2 \leq \cdots \leq n_L \text{ and } m_1 \leq m_2 \leq \cdots \leq m_L$$

3. Maintaining the continuity of the WP elements:

$$p_l + 1 - p_l \in \{(0,1), (1,0), (1,1)\}, \text{ for } l \in [1; L-1]$$

4.2.2 Quantum Dynamic Time Warping

Following the classical approach described above, we map all the possible distances between elements of the two sequences into a distance matrix D and search in this matrix for the WP between the two elements $d_{1,1}$ and $d_{M,N}$ that has the smallest distance. This matrix can be seen as a DAG, with the vertices of this graph representing the elements of the matrix and the edges representing the possible transition of the graph following the constraint of the classical DTW algorithm [Mül07b]. In section 7 of this chapter, we have proposed an algorithm that allows us to find all the paths between two vertices of a DAG. Therefore, we can modify this algorithm according to the objective sought here, i.e. modeling this algorithm in such a way as to seek the WP having the smallest distance between the two elements $d_{1,1}$ and $d_{M,N}$ of the graph.

Construction of the walking graph

We would like to use the quantum walks philosophy to propose a quantum version of the DTW algorithm, this quantum walk takes place in a DAG. In this paragraph, we explain how to build this DAG from the distance matrix D between two sequences X and Y. In this sense, we use $N \times M$ vertices, for each element $d_{i,j}$ of the matrix D we use a vertex $v_{i,j}$. Following the constraint 3 of classical DTW, we can step in the matrix from an element $x_{m,n}$ either to $x_{m+1,n}$, $x_{m,n+1}$ or $x_{m+1,n+1}$ and this steps must be continuous. Then, to ensure this constraint, for each vertex $v_{i,j}$ with $i = 0, \ldots, n-1$ and $j = 0, \ldots, m-1$ we add three edges:

$$(v_{i,j}, v_{i+1,j}), (v_{i,j}, v_{i,j+1}), (v_{i,j}, v_{i+1,j+1}) \tag{4.11}$$

with the three weights $d_{i+1,j}$, $d_{i,j+1}$ and $d_{i+1,j+1}$ respectively. Each $v_{i,m}$ has an edge to $v_{i+1,m}$ with the weight $d_{i+1,m}$ for each $i = 0, \ldots, n-1$, and $v_{n,j}$ has an edge to $v_{m,j+1}$ with the weight $d_{m,j+1}$ for each $j = 0, \ldots, m-1$. To take the distance $d_{0,0}$ into account, we add to the three edges $\{(v_{0,0}, v_{0,1}), (v_{0,0}, v_{1,0}), (v_{0,0}, v_{1,1})\}$ the weight $d_{0,0}$. In Figure 4.15 we show the equivalent graph of the general matrix presented in Table 4.2.

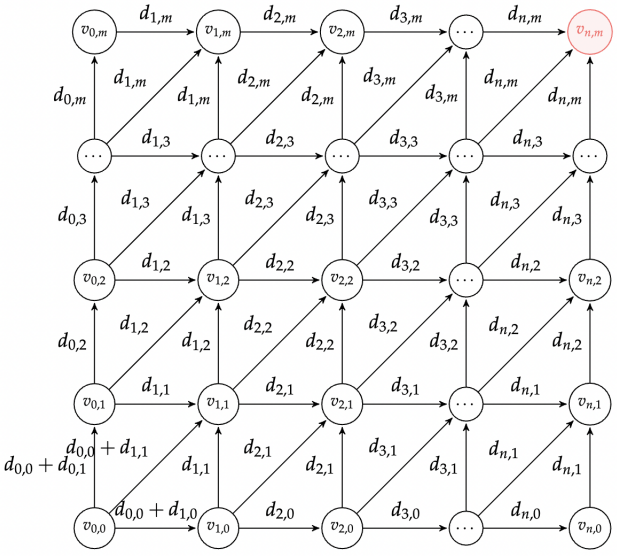


FIGURE 4.15: The graph of the distance matrix between the two sequences X and Y

The process to build this DAG is generalized in the algorithm 8, it takes as input the two sequences X and Y, and it allows to perform a DAG.

Algorithm 8: Building a DAG for QDTW

Input : Two sequences $X = \{x_1, x_2, \ldots, x_M\}$ and $Y = \{y_1, y_2, \ldots, y_M\}$.
Initialisation
Initialize the sets: $\mathbf{V} = \{v_{i,j}, \text{ for } i = 0, \ldots, M \text{ and } j = 0, \ldots, N\}$, $\mathbb{E} = \emptyset$ and $\mathbf{C} = \emptyset$.
for *all* $v_{i,j} \in \mathbf{V}$ **do**
  Add to $\mathbb{E}$ the edge $(v_{i,j}, v_{i+1,j})$ with the weight $d(x_{i+1}, y_j)$ if $i < n$.
  Add to $\mathbb{E}$ the edge $(v_{i,j}, v_{i,j+1})$ with the weight $d(x_i, y_{j+1})$ if $j < m$.
  Add to $\mathbb{E}$ the edge $(v_{i,j}, v_{i+1,j+1})$ with the weight $d(x_{i+1}, y_{j+1})$ if $i < n$ and $j < m$.
end
Add to $\mathbb{E}$ the edge $(v_{i,m}, v_{i+1,m})$ with the weight $d_{i+1,m}$ and $i = 0, \ldots, n-1$
Add to $\mathbb{E}$ the edge $(v_{n,j}, v_{m,j+1})$ with the weight $d_{m,j+1}$ and $j = 0, \ldots, m-1$
Add the weight $d_{0,0}$ for each edge $e \in \{(v_{0,0}, v_{0,1}), (v_{0,0}, v_{1,0}), (v_{0,0}, v_{1,1})\}$
$\mathbf{C} = \{v_{M,N}\}$
Return : $\mathbb{G} = (\mathbf{V}, \mathbb{E}, \mathbf{C})$

As a consequence, in order to find the optimal WP that matches the two sequences X and Y, we need to find the path that has the smallest accumulated distance between the source $v_{0,0}$ and the end $v_{n,m}$.

QDTW Algorithm

Suppose that we have two sequences X and Y, and we want to find the matching between these two sequences with a quantum algorithm as in the classical case with the DTW algorithm. We start by constructing the graph that represents the distance matrix between these two sequences using the algorithm 8, and we apply our quantum walks approach to this graph with a storage strategy while keeping only the path with the smallest distance.

In order to have the possibility of using our approach on large sequences, we present directly a hybrid approach that allows us to make calls to the algorithm 7 and find this path with the smallest distance, this approach is summarized in the algorithm 9.

The update of the warping path WP and the sets P_t and M is done as follows:

- If any path $\lambda_j \in \Lambda_i$ arrives at one of the marked vertices $c_i \in \mathbf{C}$, for each path $\lambda_k \in P_t$ if the last element of λ_k is v_t, concatenate the two paths $\lambda_{i+k} = \lambda_j + \lambda_k$ and compute the sum of their distances, if this cumulated distance is less than the distance of the current WP replace WP by λ_{i+k} otherwise λ_{i+k} is ignored.
- Updating P_t, for each $\lambda_j \in \Lambda_i$ and for each $\lambda_k \in P_t$ with the last item of λ_k is v_t, we add to P_t the concatenation $\lambda_{i+k} = \lambda_j + \lambda_k$ if the end v_t of λ_j not in $\mathbf{C}$, $v_t \notin \mathbf{C}$, we remove λ_k from P_t.
- Finally, all the ends of the paths of the set P_t are added to M if they are not already in M.

Algorithm 9: Quantum Dynamic Time Warping (QDTW)

Input : Two sequences: $X = \{x_1, x_2, \ldots, x_M\}$ and $Y = \{y_1, y_2, \ldots, y_N\}$, number of qubit available in the quantum computer N_q.

Initialisation

Construct the graph $\mathbb{G} = (\mathbf{V}, \mathbb{E}, \mathbf{C})$ corresponding to the two sequences using the algorithm 8

Initialize the set of walks to be processed M with the source $M = \{v_{0,0}\}$

Initializes the warping path at $WP = \emptyset$ and the set of current paths at $P_t = \emptyset$.

while $M \neq \emptyset$ **do**

v_t is the first item of M

Extract a sub-graph $\mathbb{G}_i$ of $\mathbb{G}$ from v_t such that the number of vertices of $\mathbb{G}_i$ is less than N_q.

Extract the set paths Λ_i from $\mathbb{G}_i$ using the algorithm 5

Calculate the exact distance of each path $\lambda_j \in \Lambda_i$ as follows:

$$d(\lambda_j) = \sum_{e \in \lambda_j} d_e, \text{ with } e \text{ denoting the edges of the path}$$

Remove the item v_t from the list M

Update WP, P_t and M with the new list of found paths Λ_i.

end

Return : WP

Complexity analysis

For calculating the quantum complexity of our approach, let us assume that we have a quantum computer with a large number of qubits that doesn't require the hybridization of our algorithm. Let us also suppose that we have two sequences X and Y of size N and M respectively. The graph that represents the distance matrix between these sequences is of $N \times M$ vertices because we use for each element of the matrix a vertex. For each vertex, we have three edges to the successors, except for the vertices of the extremity we have only one (look at the general architecture of the graph in Figure 4.15). Then in this graph, we have $3((N-1) \times (M-1)) + N + M$ edges. With our algorithm, we use for each vertex of the graph a qubit which signifies we use for the totality of our algorithm $N \times M$ qubits. For the number of the used gates, it is necessary to calculate them according to the edges that exist in the graph. We have $(N-1) \times (M-1)$ that they have three successors, then for constructing the oracles it is necessary to use $(N-1) \times (M-1)$ gates $CU(\beta)$, $(N-1) \times (M-1)$ gates $CCU(\beta)$ and $(N-1) \times (M-1)$ gates $CCCU(\beta)$ in addition to that $N + M$ gates $CU(\beta)$ for the vertices of the extremity.

Discussions

With the algorithm 9, we have proposed a quantum algorithm that gives the same results as the classical algorithm of DTW, this algorithm takes as input two sequences X and Y and it allows to return a WP with a minimal distance. It can be used in the case where we have a quantum computer enough to find these results with only one iteration (in the case where we have $N \times M < N_q$), also it can be used with a hybrid strategy to find these results using a small quantum computer (a number of qubits N_q is very small). This algorithm allows to make calls to the algorithm 5, which finds all the paths between two vertices, for QDTW, it allows to make calls to the algorithm5,

using the graph that represents the distance matrix between the two sequences X and Y. The latter is constructed with the algorithm 8 and the two vertices $v_{0,0}$ and $v_{m,n}$. Thus, as we are guaranteed to have processed all the paths, we are guaranteed to have found the WP that has the smallest distance.

The complexity of the classical DTW is $O(MN)$, and for us, in the quantum case we use $N \times M$ qubit and $3N \times M$ gates if we generalize, which means that with this approach we don't really see a gain in the complexity, except that with the use of the quantum computers we have the possibility to execute the circuit faster than a classical algorithm in a classical computer. With this approach we have treated the case of one to one, i.e. we have not taken the case of using sub-sequences in each sequence as we have presented in the problem. So, what remains now is to find out how we can make our algorithm valid to deal with the problematic with sub-sequences, and also whether we have a gain of complexity or not. Next to this, we address this problem with more details.

4.2.3 Quantum Dynamic Time Warping with sub-sequences

In the above discussion, we have proposed a quantum version of the DTW algorithm, where we deal only with the case of matching an element of the first sequence with an element of the second sequence at each time. In what follows, we focus on the case of matching an element x_i of X to an element y_i of Y, and the case of a sub-sequence $\{x_k, x_{k+m}\}$ of X to a sub-sequence $\{y_k, y_{k+n}\}$ of Y, where subsequences can be of different sizes ($n \neq m$ or $n = m$).

Sub-sequences matching

In order to define the issue here, let's take two general sequences, X and Y, and let's define the limit of the size of the sub-sequences by β. The objective is to find the matching that has the smallest distance between X and Y, where at each iteration we progress in X with a sub-sequence of size k with $1 \leq k \leq \beta$, and in Y with a sub-sequence of size k' with $1 \leq k' \leq \beta$. k and k' here can be different, ($k = k'$ or $k \neq k'$), in the same iteration and also can be changed from an iteration to another. Figure 4.13 shows a case of two sequences matched by different sub-sequences at each time. In this figure, we can see that the sub-sequence (with different colors in the same sequence) can change the size from one sub-sequence to another and that two sub-sequences (in the same color) can be matched even if they don't have the same size.

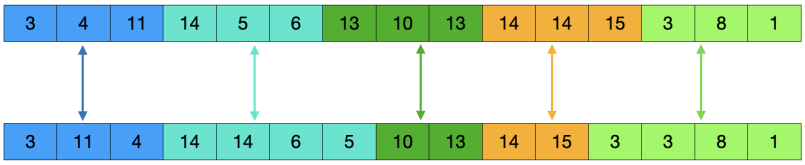


FIGURE 4.16: Example of matching between two sequences taking into account the sub-sequences

In the classical case, we can do this with the DTW algorithm in β-dimensional space,

where it searches in this space for the WP that has the smallest distance between the first element in the bottom corner and the last element in the top corner (see fig. 4.17).

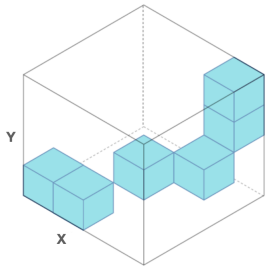


FIGURE 4.17: The problem of finding the matching between two sequences using sub-sequences

As we have already mentioned, this problem is NP-complete [PKG11] and it's impossible to solve it when β is large [WJ94].

Construction of the walking graph

In the case that one element can be matched by another at each time, we have for each vertex of the graph three successors (three edges).

1. The first edge $(v_{i,j}, v_{i+1,j})$ allows us to advance with only one element in the sequence X with the weight $d_{i+1,j}$ and do not advance in the sequence Y,
2. The second edge $(v_{i,j}, v_{i,j+1})$ allows us to advance with only one element in the sequence Y with the weight $d_{i,j+1}$ and do not advance in the sequence X,
3. The third edge $(v_{i,j}, v_{i+1,j+1})$ to advance in the two sequences with a single element with the weight $d_{i+1,j+1}$.

With this strategy, we treat the case of one to one, we just advance with an element to the successors. In order to make it possible to treat sub-sequences, it's necessary to give our algorithm the possibility to advance directly to other vertices (not only the three nearest). For example, if we want to advance with three elements in each sequence, in the first iteration, we can go from the first vertex $v_{0,0}$ to the vertex $v_{3,3}$ directly with the weight equal the distance between the two first sub-sequences of size 3. In Figure 4.18 we add the edge that gives this possibility with the green color.

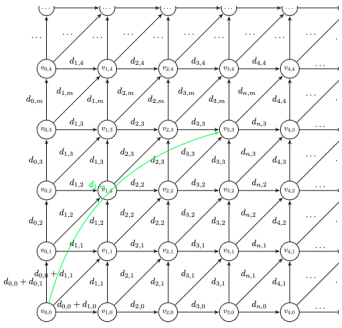


FIGURE 4.18: Add the possibility to process the forward with 3 elements in each sequence with an edge in the graph

In order to add the possibility to go forward with any sub-sequence size less than β, it's enough to add to the graph all possible edges to represent them. To do that, for each vertex $v_{i,j}$ we add the following edges: $(v_{i,j}, v_{i+K,j+k'})$ with the weight $d(x_{i,\dots,i+k}, y_{j,\dots,j+k'})$ if $i+k < n$ and $j+k' < m$ for each $k = 1,\dots,\beta$ and $k' = 1,\dots,\beta$. β is the maximum size of the sub-sequences.

Therefore, the algorithm to build the graph for the case of QDTW with sub-sequences becomes the algorithm 10. This latter takes as input two sequences X and Y and the maximum size of the sub-sequences β, it allows to apply the construction of the graph exactly like the algorithm 8 and adds all the possible edges that represent all the possible matching between two sequences. At the end, it gives a DAG that gives us the possibility to find the best matching between two sequences by processing sub-sequences. The value of β is chosen according to each application, it represents the maximum size that can be taken in a sub-sequence. In our problem in the PSA field, we can affirm that we try to find the similarity between the sequences and we take into consideration that we can have a maximum of five successive failures in a very short time, so we can choose $\beta = 5$.

Algorithm 10: Building graph for QDTW with sub-sequences

Input : Two sequences $X = \{x_1, x_2, \dots, x_M\}$ and $Y = \{y_1, y_2, \dots, y_M\}$ the maximum size of the sub-sequences β.

Initialisation

Initialize the sets: $\mathbf{V} = \{v_{i,j}, \text{ for all } i = 0,\dots,M \text{ and } j = 0,\dots,N\}, \mathbb{E} = \emptyset$ and $\mathbf{C} = \emptyset$.

for *all* $v_{i,j} \in \mathbf{V}$ **do**
- Add to $\mathbb{E}$ the edge $(v_{i,j}, v_{i+1,j})$ with the weight $d(x_{i+1}, y_j)$ if $i < n$.
- Add to $\mathbb{E}$ the edge $(v_{i,j}, v_{i,j+1})$ with the weight $d(x_i, y_{j+1})$ if $j < m$.
- Add to $\mathbb{E}$ the edge $(v_{i,j}, v_{i+1,j+1})$ with the weight $d(x_{i+1}, y_{j+1})$ if $i < n$ and $j < m$.
- **for** $k' = 0,\dots,\beta$ **do**
  - **for** $k = 0,\dots,\beta$ **do**
    - Add to $\mathbb{E}$ the edge $(v_{i,j}, v_{i+K,j+k'})$ with the weight $d(x_{i,\dots,i+k}, y_{j,\dots,j+k'})$ if $i+k < n$ and $j+k' < m$ and $(v_{i,j}, v_{i+K,j+k'}) \notin \mathbb{E}$ and $k \neq k' \neq 0$.
  - **end**
- **end**

end

Add to $\mathbb{E}$ the edge $(v_{i,m}, v_{i+1,m})$ with the weight $d_{i+1,m}$ and $i = 0,\dots,n-1$

Add to $\mathbb{E}$ the edge $(v_{n,j}, v_{m,j+1})$ with the weight $d_{m,j+1}$ and $j = 0,\dots,m-1$

Add the weight $d_{0,0}$ for each edge $e \in \{(v_{0,0}, v_{0,1}), (v_{0,0}, v_{1,0}), (v_{0,0}, v_{1,1})\}$

$\mathbf{C} = \{v_{M,N}\}$

Return : $\mathbb{G} = (\mathbf{V}, \mathbb{E}, \mathbf{C})$

In Figure 4.19, we show the edges that we have added to the vertex $v_{0,0}$, in the case of using $\beta = 2$ with the green color. We have added 5 edges:

1. $(v_{0,0}, v_{2,2})$ with the weight $d_0 = dist(X_{\{:,2\}}, Y_{\{:,2\}})$ to represent the case of taking one sub-sequence of size 2 in each sequence among X and Y,

2. $(v_{0,0}, v_{1,2})$ with the weight $d_1 = dist(X_{\{:,1\}}, Y_{\{:,2\}})$ to represent the case of a sub-sequence of size 1 in X and of size 2 in Y,

3. $(v_{0,0}, v_{2,1})$ with the weight $d_2 = dist(X_{\{:,2\}}, Y_{\{:,1\}})$ to represent the case of a sub-sequence of size 1 in Y and of size 2 in X,

4. $(v_{0,0}, v_{0,2})$ with the weight $d_3 = dist(X_{\{0\}}, Y_{\{:,2\}})$ to represent the case of a sub-sequence of size 2 in Y and and doesn't move forward in X,

5. $(v_{0,0}, v_{2,0})$ with the weight $d_4 = dist(X_{\{:,2\}}, Y_{\{0\}})$ to represent the case of a sub-sequence of size 2 in X and and doesn't move forward in Y,

The distance $dist(x, y)$, it can be chosen at any distance depending on the application domain.

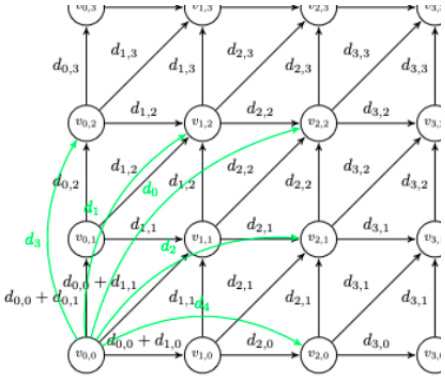


FIGURE 4.19: Example of added edges to process sub-sequences

QDTW-β Algorithm

Finally, to find the matching between the two sequences X and Y taking into account the case of sub-sequences, we use the algorithm 10 to construct the graph and we propose the algorithm 11.

All the characteristics of the algorithm 9, remain present in this algorithm, except that here we have an additional parameter, the beta value that represents the maximum size of sub-sequences. This algorithm uses the algorithm 10 that allows to add the edges that give the possibility of processing sub-sequences. Also, if we have a quantum computer able to process the two sequences X and Y, it finds the results in a single iteration, if not, it works with a hybrid strategy to find it.

Complexity

As in the case of QDTW, let us suppose that we have a quantum computer able to process any size of sequences. And suppose that we have two sequences X and Y of size N and M respectively. Following the graph used in the first algorithm 9, we did not add any vertex to the graph for the algorithm 11. According to the complexity of the algorithm 9, in the algorithm 11 we use $N \times M$ qubits. For the number of edges, we added for each vertex a set of edges depending on the value β. For each vertex of the graph we add $\beta\beta! + 1$ edges. Therefore, to build the walking oracles in this graph, we use $3N \times M + N \times M(\beta\beta! + 1)$ quantum gates.

Algorithm 11: QDTW with sub-sequences (QDTW-β)

Input : Two sequences: $X = \{x_1, x_2, \ldots, x_M\}$ and $Y = \{y_1, y_2, \ldots, y_N\}$, number of qubits available in the quantum computer N_q, β value.

Initialisation

Construct the graph $\mathbb{G} = (\mathbf{V}, \mathbb{E}, \mathbf{C})$ corresponding to the two sequences X and Y using the algorithm 10

Initialize the set of walks to be processed M with the source $M = \{v_{0,0}\}$.

Initializes the warping path at $WP = \emptyset$ and the set of current paths at $P_t = \emptyset$.

while $M \neq \emptyset$ **do**

v_t is the first item of M.

Extract a sub-graph $\mathbb{G}_i$ of $\mathbb{G}$ from v_t such that the number of vertices of $\mathbb{G}_i$ is less than N_q.

Extract the set paths Λ_i from $\mathbb{G}_i$ using the algorithm 5.

Calculate the exact distance of each path $\lambda_j \in \Lambda_i$ as follows:

$$d(\lambda_j) = \sum_{e \in \lambda_j} d_e, \text{ with e denoting the edges of the path}$$

Remove the item v_t from the list M.

Update WP, the set P_t and the set M with the new list of found paths Λ_i.

end

Return : WP

Discussion

In this section, we proposed the first quantum algorithm that can handle the problem of calculating the similarity between two sequences taking into account the case of the matching between sub-sequences. The great difficulty in dealing with this problem in classical computing is the complexity, it is impossible to process it in the case where the value of β is very large [PKG11]. In our contribution, we propose a quantum version to improve this computation time and to find the results quickly. The proposed algorithm QDTW-β, doesn't use more than $N \times M$ qubits as the case of QDTW, the only addition here is the number of used gates. This is a very positive point for our algorithm because it makes easier the use of our algorithm in the near future with a slightly larger number of qubits and does not wait for a quantum computer with a very big number of qubits. Also, it can be used in a hybrid way according to the available number of qubits. Thanks to the quantum walks (a deterministic approach), we have the guarantee of processing all possible matching between X and Y, and we have selected the matching with the smallest distance.

4.2.4 Results of the two algorithms QDTW and QDTW-w

In this section, we compare the results obtained for each proposed algorithm. We start by defining the distance used to construct the matrix of distances D of the classical DTW algorithm, and the DAG in the quantum case. After that, we compare the obtained results with the three algorithms: QDTW, the quantum approach proposed in the paper [Fel+20], and the classical DTW. Secondly, we compare the results found by the QDTW-β approach with two other sub-sequence approaches.

Distance

In all the tests of this section, and the following section, we will use accidental sequences that represent failure scenarios of a system in the PSA field, to do this, we define how to calculate the distance in this application case. Let's suppose that we have two sequences $X = \{x_1, \ldots, x_N\}$ and $Y = \{y_1, \ldots, y_M\}$, the distance between X and Y is defined as follows:

$$dist(X^*, Y^*) = \frac{|X^* \Delta Y^*|}{|X^*| + |Y^*|} \tag{4.12}$$

with Δ represents the symmetric difference, X^* all the elements that build X, Y^* all the elements that build Y, $|X^*|$ the number of independent elements that build X, and $|Y^*|$ the number of independent elements that build Y.

Comparative results of the QDTW approach

Starting by testing and comparing our approach presented in 4.2.2 with the approach of [Fel+20]. In contrast to the latter, which uses adiabatic quantum computers, we use universal quantum computers. Therefore, in order to test our approach, we need to find a universal quantum computer able to apply the general gate U defined in **??**. For this purpose, we use the IBM quantum simulator with 32 qubits, which is currently able to perform this task. So, we randomly took 100 sequences for each size $L = 5, \ldots, 30$ and we performed a series of tests to calculate the distances between these sequences and represented the results in the Table 4.3.

Sequence length (N×M)	5×5	10×10	15×15	20×20	from 20×20
Classical approach	100%	100%	100%	100%	100%
[Fel+20] approach	92%	90%	60%	70%	20%
Our approach	100%	100%	100%	100%	100%

TABLE 4.3: The results of comparing our approach with [Fel+20]'s approach and the classical approach.

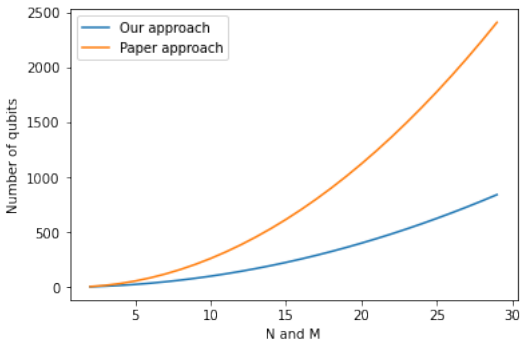


FIGURE 4.20: Number of qubits used by each approach

In Table 4.3 we can see clearly that our approach is more powerful compared to the approach of [Fel+20]. In addition to that, our approach uses only $N \times M$ qubits instead of $(N-1) * (M-1) * 3 + (N-1) + (M-1)$ qubits from the approach of [Fel+20]. In Figure 4.20, we show the number of qubits used by this approach and our approach, we can clearly see that the number of qubits used by [Fel+20]

increases more quickly with the increasing of the size of the sequence compared to our approach.

Comparative results of the QDTW-β algorithm

In most of the papers that deal with the case of using sub-sequences to find the optimal matching between sequences, a learning algorithm (1-NN algorithm) is used to choose the best size of sub-sequences used for the whole dataset. This provides a result of a fixed size of sub-sequences. In our contribution, instead of using this learning strategy to find the best size of this sub-sequence, we will process all the possible cases and compare the results with our approach.

Consider two different sequences $A = \{3, 4, 11, 14, 5, 6, 7, 10, 13, 14, 15, 3, 4, 11, 14, 5, 6, 7, 10, 13, 14, 15\}$ and $B = \{3, 4, 11, 11, 3, 4, 11, 11, 7, 14, 5, 13, 14, 15, 10, 3, 4, 11, 14, 5, 6, 7\}$, and compare the results found by the 4 different approaches:

1. Classical approach without sub-sequences (Classical DTW),
2. Classical approach with sub-sequences of fixed size k and moving forward in the sequence with one element at each time (DTW_1-k),
3. Classical approach with sub-sequences of fixed size k and moving forward in the sequence with the size k of the sub-sequence at each time (DTW_k-k).
4. Our quantum approach QDTW-β with $\beta = 5$

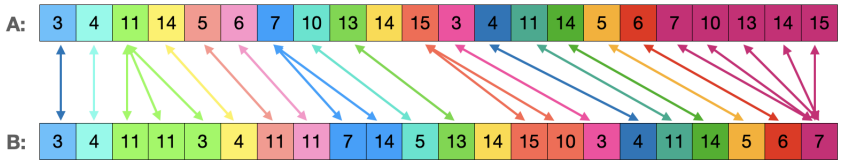
The results of these tests on the two sequences A and B are shown in Table 4.4.

Approche	Result
Classical DTW	11.02
DTW_2-2	13.0
DTW_3-3	10.33
DTW_4-4	7.625
DTW_5-5	6.8
DTW_1-2	35.5
DTW_1-3	36.66
DTW_1-4	36.0
DTW_1-5	35.94
$QDTW$-5	1.84

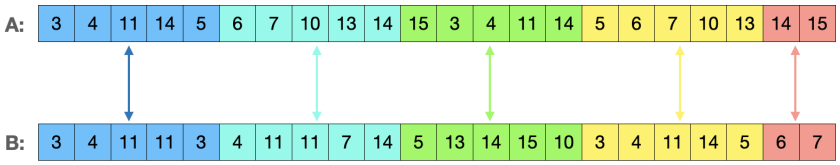
TABLE 4.4: Results of the test of the QDTW-β approach

In this table, we can see that our approach (QDTW-β) finds the best matching between the two sequences A and B. For the other two approaches, we find that the best is the classical DTW for the first and the second is DTW_5-5. This matching for the three cases is shown in Figure 4.21. Each matching is represented by an identical color and an arrow.

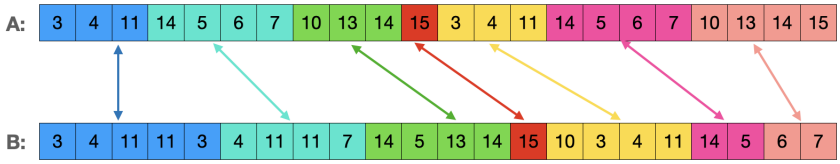
Regarding the result of our QDTW-5 approach in the Figure 4.21, we can see that the size of sub-sequences changes from one iteration to another, and that the matching



((A)) Classical DTW



((B)) DTW_5-5



((C)) QDTW-5

FIGURE 4.21: The matching between A and B: (a) using the classical DTW. (b) using classical DTW_5-5. (c) using QDTW-5.

can be done between two sub-sequences of different sizes, which makes our approach more realistic and more suitable for data in any domain. This richness of changing sizes, both within the sub-sequences in each sequence and between the two sequences, allows our approach to find the best matching between sequences. In the classical case, we can't process the sub-sequences and for the DTW_k-k case, unfortunately, it can work very well in the ideal case where we know the size of the sub-sequence at 100%.

Classification of sequences using QDTW-β and 1-NN

In order to check the impact of our QDTW-β approach on the classification results of sequences, we use the K-NN algorithm with $k = 1$. Therefore, we present in the following the testing datasets and the results found for the three approaches: Classical-DTW, DTW_k-k and $QDTW$-β with $k = 2, 3, 4, 5$, and we consider that the sub-sequences cannot exceed the size of 5. So, for our approach $QDTW$-β $\beta = 5$.

Datasets

We have 6 systems, each system has a dataset of sequences and each class of each dataset represents the scenarios that can drive the system to an unacceptable consequence. The details related to these datasets are represented in the Table 7.3.

Dataset	SC_1	SC_2	SC_3	SC_4	SC_5	SC_6
Length of each scenario	8	7	9	8	9	9
Length of training dataset	44	44	44	64	16	16
Length of test dataset	24	24	16	16	12	9
Number of classes	3	3	3	3	3	3

TABLE 4.5: 6 datasets of 6 systems

Result of classification

We use the K-NN algorithm with k=1 and Accuracy as a metric, and we test the 3 approaches on the 6 datasets mentioned in the previous subsection. The results that we found are shown in Table 7.4.

Method	SC_1	SC_2	SC_3	SC_4	SC_5	SC_6
Classical DTW	0.25	0.25	0.25	0.81	0.25	0.58
Quantum DTW	0.25	0.25	0.25	0.81	0.25	0.58
Faster DTW-2	0.67	0.44	0.88	0.50	0.44	0.50
Faster DTW-3	0.46	0.25	0.38	0.31	0.19	0.42
Faster DTW-4	0.29	0.25	0.19	0.25	0.19	0.33
Faster DTW-5	0.25	0.25	0.19	0.25	0.25	0.33
QDTW-w	0.75	0.67	0.88	0.81	0.75	0.67

TABLE 4.6: Classification results of sequences with different approaches

In the Table 7.4, we can see that our approach has the highest metric in all datasets, which demonstrates that using our approach to calculate the distance between scenarios in our particular problem is the best choice.

4.3 A quantum learning approach based on Hidden Markov Models for failure scenarios generation

One of the most studied models recently in quantum computing [Jae07] is the Quantum Hidden Markov Models (QHMMs), the paper [Adh+20] demonstrates that QHMMs are a special subclass of the general class of Observable Operator Models (OOMs) and also provides a learning algorithm for HQMMs using the constrained Gradient Descent method. It also demonstrates that this approach is faster and more suitable for larger models compared to previous learning algorithms. In addition, there are other learning algorithms of QHMMs like [SGB18], this work demonstrates some theoretical advantages of QHMMs compared to HMMs and also that each HMM of finite dimension can be modeled by a QHMM of finite dimension.

4.3.1 Classical Hidden Markov Models

We begin by briefly recalling the definition of the classical Hidden Markov Model (HMM). A hidden Markov model is a tool for representing probability distributions over sequences of observations. It is assumed that each observation X_t was generated by some process whose state $S_t \in [1..K]$ is unknown (hence the name hidden).

Definition 6 (HMM) An HMM is made of five key elements:

1. **An alphabet** $\Sigma = \{o_1, \cdots, o_M\}$.
2. **A set of index of states** $Q = \{1, \cdots, K\}$
3. **A transition probability matrix** $A = (a_{kk'})$, $A \in \mathbb{R}^{K \times K}$, $\forall k \sum_{k'} a_{kk'} = 1$, $a_{kk'}$ is the probability of transition from state k to state k'.
4. **Emission probabilities** within each state: $e_k(x) = \mathbb{P}(x|Q = k)$, $\sum_{x \in \Sigma} e_i(k) = 1$.
5. **Starting probabilities**: $\pi_1, \cdots, \pi_K$, $\sum_k \pi_k = 1$.

Therefore a HMM model is denoted $M = \{A, B, \pi\}$, where: A is the transition probability matrix, B contains the emissions probability laws $e_k(x)$, π the starting probabilities.

The Forward-Backward algorithm

Given an HMM, we can generate a sequence of length n as follows:

1. Start at state Q_1 according to π_1
2. Emit observation x_1 according to $e_{Q_1}(x_1)$
3. Go to state Q_2 according to a_{Q_1,Q_2}
4. ... until emitting x_n

More precisely, given the model M, we want an algorithm that can compute the following probabilities:

- $\mathbb{P}(X)$ the probability of X
- $\mathbb{P}(x_i \cdots x_j)$ the probability of a substring of X
- $\mathbb{P}(Q_i = k|X)$ the posterior probability that the i^{th} state is k, given X

In order to calculate $\mathbb{P}(X)$, the probability of the whole sequence given the HMM, we sum all possible ways of generating X:

$$\mathbb{P}(X) = \sum_{\boldsymbol{Q}} \mathbb{P}(X, \boldsymbol{Q}) = \sum_{\boldsymbol{Q}} \mathbb{P}(X|\boldsymbol{Q})\mathbb{P}(\boldsymbol{Q}) \tag{4.13}$$

To avoid summing over an exponential number of paths $\boldsymbol{Q}$, we define the **forward probability**:

$$f_k(i) = \mathbb{P}(x_1 \cdots x_t, Q_t = k) \tag{4.14}$$

This represents the probability of observing the sequence $x_1 \cdots x_t$ and having the t^{th} state being k. Straightforward computations give:

$$f_k(t) = \mathbb{P}(x_1 \cdots x_t, Q_t = k) = e_k(x_t) \sum_l f_l(t-1) a_{lk} \tag{4.15}$$

The Equations (4.15) represent the key of the so-called "Forward Algorithm" since we can compute all the $f_k(t)$ using dynamic programming as follows:

1. firstly we initialize $f_0(0) = 1$ and $\forall k > 0 \quad f_k(0) = 0$,
2. secondly we iterate $f_k(i) = e_k(x_t) \sum_l f_l(t-1) a_{lk}$,

3. we terminate by putting $\mathbb{P}(X) = \sum_k f_k(N)$.

In order to compute $\mathbb{P}(Q_t = k|X)$ the probability distribution on the t^{th} position given X, we proceed as follows: since we have $\mathbb{P}(Q_t = k|X) = \frac{\mathbb{P}(Q_t=k,X)}{\mathbb{P}(X)}$, we start by computing $\mathbb{P}(Q_t = k, X) = \mathbb{P}(x_1 \cdots x_t, Q_t = k, x_{t+1} \cdots x_N)$, we obtain:

$$\mathbb{P}(Q_t = k, X) = f_k(t)b_k(t),$$

where $b_k(t)$ are the backward probability $b_k(t) = \mathbb{P}(x_{t+1} \cdots x_N|Q_t = k)$. It follows that:

$$b_k(t) = \sum_l e_l(x_{t+1})a_{kl}b_l(t+1). \tag{4.16}$$

The Equations (4.16) represent the key of the so-called "Backward Algorithm" since we can compute all the $b_k(t)$ using dynamic programming as follows:

1. firstly we initialize $\forall k \quad b_k(N) = 0$,
2. secondly we iterate $b_k(i) = \sum_l e_l(x_{t+1})a_{kl}b_l(t+1)$,
3. we terminate by putting $P(X) = \sum_l \pi_l e_l(x_1)b_l(1)$.

The Baum-Welch algorithm

The Forward-Backward algorithm can be adapted into an EM like procedure to learn the model. This is known as the Baum-Welch algorithm.

Repeat until convergence:

- Compute the probability of each state at each position using forward and backward probabilities. This gives the expected distribution of the observations for each state using the Bayes Theorem.
- Compute the probability of each pair of states at each pair of consecutive positions t and $t+1$ using forward(t) and backward(t+1). This gives the expected transition counts.

4.3.2 Quantum Hidden Markov Models

The quantum version of HMM is the Quantum Hidden Markov Model (QHMM). To represents a QHMM we use quantum circuits.

Definition 7 (Quantum Hidden Markov Model) A K-dimensional Quantum Hidden Markov Model $(K - HQMM)$ is made by :

1. **An alphabet**, a set of discrete observations $\Sigma = \{o_1, \cdots, o_M\}$.
2. **A set of index of states** $Q = \{1, \cdots, K\}$
3. **A set of Kraus operators** $\{\psi_{x,\mu_x}\}_{x\in\Sigma,\mu_x\in\mathbb{N}} \in \mathbb{C}^{K\times K}$ where $\sum_{x,\pi} \psi^{\dagger}_{x,\pi}\psi_{x,\pi} = \mathbb{I}$ and † denotes the complex conjugate transpose.
4. **An initial state** $\pi_0 \in \mathbb{C}^{K\times K}$ where π_0 is a Hermitian positive semidefinite matrix of arbitrary rank and $tr(\pi_0) = 1$.

Therefore a $K-HQMM$ model is denoted $qM = (\mathbb{C}^{K\times K}, \{\psi_{x,\mu_x}\}_{x\in\Theta}, \pi_0)$ where $\pi_0 \in \mathbb{C}^{K\times K}$ is the initial state, and $\{\psi_{x,\mu_x}\}_{x\in\Theta,\mu_x\in\mathbb{N}} \in \mathbb{C}^{K\times K}$ are the Kraus operators. In classic context the 3) in Definition 6, we use linear map A and B to define different actions: emission and transition. In the quantum context, we use Kraus operators which characterize completely any positive map. Moreover, by Kraus Theorem any action Φ to a quantum state ρ writes $\Phi(\rho) = \sum_i \mathbb{B}_i \rho \mathbb{B}_i^\dagger$, with $\sum_i \mathbb{B}_i^\dagger \mathbb{B}_i = \mathbb{I}$, the complex matrix $\mathbb{B}$ are the Kraus operators. In the quantum context, both emission and transition are written as action by using Kraus operators. Therefore the update status π_t rule is computed as follows:

$$\pi_t = \frac{\sum_{\mu_x} \psi_{x,\mu_x} \pi_{t-1} \psi_{x,\mu_x}^\dagger}{tr(\sum_{\mu_x} \psi_{x,\mu_x} \pi_{t-1} \psi_{x,\mu_x}^\dagger)} \tag{4.17}$$

and probability of a given sequence is given by:

$$\mathbb{P}(X) = tr(\sum_{\mu_{x_t}} \psi_{x_t,\mu_{x_t}}, ..(\sum_{\mu_{x_1}} \psi_{x_1,\mu_{x_1}} \pi_0 \psi_{x_1,\mu_{x_1}}^\dagger), ..\psi_{x_t,\mu_{x_t}}^\dagger)) \tag{4.18}$$

Definition 8 The Learning Problem

- **Given:** The sequence of observations: $X = \{x_1, \ldots, x_L\}$
- **Question:** How do we learn Kraus operators $\{\psi_{x,\mu}\}$ to model X using an QHMM?

Both approaches [SGB18; Adh+20] use the negative $\mathcal{L}$ log-likelihood of the data as a loss function. This function can be written as a function of the set of Kraus operators $\{\psi_{x,\mu}\}$ as follows:

$$\mathbb{L} = -\ln(\sum_{\mu_{x_t}} \psi_{x_t,\mu_{x_t}}, ..(\sum_{\mu_{x_1}} \psi_{x_1,\mu_{x_1}} \pi_0 \psi_{x_1,\mu_{x_1}}^\dagger), ..\psi_{x_t,\mu_{x_t}}^\dagger)) \tag{4.19}$$

with the following constraint $\sum_{x,\mu} \psi_{x,\mu}^\dagger \psi_{x,\mu} = \mathbb{I}$.

An equivalent formulation to the problem of learning a set of N trace-preserving $k \times k$ Kraus operators can be the problem of learning a matrix $\Psi \in \mathbb{C}^{kN\times k}$ where $N = M\mu$ ($M = |\Sigma|$ and μ is the number of Kraus operators for each observation) and $\Psi^\dagger\Psi = \mathbb{I}$, where the blocks of Ψ represent the Kraus operators $\{\psi_{x,\mu}\}$ that parametrize the QHMM. The initialization of the matrix Ψ_0 is done randomly, provided that it is unitary. The goal is to update this matrix until we find a matrix Ψ^* that minimizes the function $\mathcal{L}$ in Equation (4.19).

Srinivasan's approach [SGB18] seeks iteratively to find a series of Givens rotations that locally increase the log-likelihood. But, a Givens rotation only modifies two rows at a time, which leads to the fact that this approach is too slow for learning large matrices. [Adh+20] proposes to solve this problem using gradient descent. It proposes to learn Ψ directly with the gradient descent method where G denotes the gradient of the loss function $\mathcal{L}$ in Equation (4.19) where the update function is:

$$\Psi_{i+1} = \Psi_i - \tau \mathbb{U}_i (\mathbb{I} + \frac{\tau}{2} \mathbf{V}_i^\dagger \mathbf{U}_i)^{-1} \mathbf{V}_i^\dagger \Psi_i \tag{4.20}$$

where $\mathbf{U}_i = [G|\Psi_i]$, $\mathbf{V}_i = [\Psi_i| - \mathbb{G}]$, and $\mathbb{G}$ is the gradient at Ψ_i. Algorithm 12 summarizes the approach performed by [Adh+20] to learn the parameters of QHMMs

using gradient descent.

Algorithm 12: Learning QHMMs [Adh+20]

Input : Training dataset $X \in \mathbb{N}^{m \times l}$, learning rate τ, learning rate decay α, number of batchs β, number of epochs v.
Output : $\{\psi_i\}_{i=1}^{M\mu}$
Initialisation
Complex orthonormal matrix on Stiefel manifold $\Psi \in \mathbf{C}^{M\mu n \times n}$ and partition into Kraus operators $\{\psi_i\}_{i=1}^{M\mu}$ whith $\psi_i \in \mathbf{C}^{n \times n}$
for *epoch = 1:v* **do**
  Partition training data X into β batches $\mathbb{X}_\beta$
  for *b=1:β* **do**
    Compute gradient $\mathbb{G}_i \leftarrow \frac{\partial l}{\partial \Psi_i}$ for batch $\{\mathbb{X}_\beta\}$ and loss function l.
    Compute $\frac{\partial l}{\partial \Psi} = \mathbb{G} \leftarrow [\mathbb{G}_1, \ldots, \mathbb{G}_M \mu]^T$
    Construct $\mathbf{U} \leftarrow [\mathbb{G} \,|\psi]$,$\mathbf{V} \leftarrow [\psi| - \mathbb{G}]$
    Update $\Psi \leftarrow \Psi - \tau \mathbf{U}(\mathbb{I} + \frac{\tau}{2} \mathbf{V}^\dagger \mathbf{U})^{-1} \mathbf{V}^\dagger \Psi$
  end
  Update learning rate $\tau = \alpha\tau$ Re-partition Ψ into $\{\Psi_i\}$
end
Return : $\{\Psi_i\}$

4.3.3 Highlights of our contribution

Suppose that we have a system $S = (\Xi, \Phi, \Lambda)$ of $|\Xi| = n$ basic events, and we search for the failure scenarios of this system Λ that have the probability greater than a fixed probability $\mathbb{P}_{min}$ $(\mathbb{P}(\lambda_i) > \mathbb{P}_{min})$ to reach a serious failure state $\phi_i \in \Phi$.

In order to answer this question, we represent the states of the system by a state graph, and we search in this graph for this subset of path Λ between the current state ϕ_t of the system and all the states of the serious failures $\phi_i \in \Phi$, using our algorithm 11. If we have a system with n basic events, automatically we will have 2^n states of the system, which gives a graph of states with 2^n vertices. The problem here is that the complexity of finding k paths between two vertices of this graph is NP-complete [AZCN21]. So, for each element of Φ, it takes a lot of time to find only k paths instead of all paths. Also, if the initial state ϕ_t is changed, all calculations must be performed again. This approach becomes impossible if we have a system with a very large number of basic events, which leads us to look for other methods to find these failure scenarios. For this reason, we have proposed the algorithm 11.

Instead of using this approach, we propose to learn QHMMs to represent the set of failure scenarios Λ of the system, which allows us to handle these scenarios in a better way, and also to create some predictive models that can be exploited dynamically to generate different possible scenarios from a given state of the system. This approach gives us the possibility to process the failure scenarios to all severe failures instead of processing them one by one, and it also gives us the possibility to process only the possible scenarios with a high probability. In addition, this approach gives us the possibility to see if a failure scenario is probable or not for a serious accident.

4.3.4 Experimental validation

In this section, we will show the results of the test on real small systems. Firstly, we start by comparing the results of QHMMs and HMMs using as metric DA described in the following.

Metric

Description accuracy is a scaled log-likelihood independent of sequence of length L [ZJ10],[SGB18]. Consider a non linear function $f : (-\infty, 1] \to (-1, 1]$, the metric is defined as the following:

$$DA = f(1 + \frac{\log_s \mathbb{P}(Y/D)}{L}) \tag{4.21}$$

where

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \frac{1-\exp(-x/4)}{1+\exp(-x/4)}, & \text{if } x < 0. \end{cases}$$

Where L is the length of the sequence, s is the number of output symbols in the sequence, Y is the data, and $\mathbb{D}$ is the model. When $DA = 1$, the model predicted the sequence with perfect accuracy, and when $DA > 0$, the model performed better than random. On the real-world dataset, we report the average accuracy for a classification problem.

Complexity

The complexity of computing the loss $\mathcal{L}$ using the Equation (4.19) is $O(\mu MLK^3)$, with M the number of sequences in the batch, L the length of sequences and for the complexity of the update function (4.20) is $O(\mu MK^3)$. On the other hand, in the classic case, we need to do K operations for each cell which gives the complexity analysis $O(MLK^2)$ for the Forward and Backward algorithms, and we need to store a matrix of size $K \times M$ for each cell which gives $O(MLK)$ for space complexity.

Test results

In this section, we will show the results of the tests on four datasets $D1, D2, D3$ and $D4$ of two small PSA systems S_1 and S_2. Where D1 and D2 is the dataset of probable and no-probable scenarios of S_1 respectively, D3 and D4 is the dataset of probable and no-probable scenarios of S_2 respectively.

Firstly, we compare the classical HMMs and the quantum QHMMs methods by using the DA metric. For this purpose we compute the average of DA metric in the results obtained by training and tests for the four datasets $D1, D2, D3, D4$, and both approaches HMMs and QHMMs, the results are shown in Figure 4.22.

From the results in 4.22, we can see that the average DA metric for QHMMs is always higher than HMMs, which means that QHMMs are more efficient than HMMs.
In order to identify the probable and no-probable scenarios of a system, we use the following strategy: For each system, we learn two QHMMs, the first to identify the probable scenarios, and the second to identify the no-probable scenarios. When we have a new scenario, we use these two models to decide if it is probable or not, we choose the model that gives the greatest metric DA. In Figure 4.23, we show the results of the two models of the first system S_1 where the blue color represents the

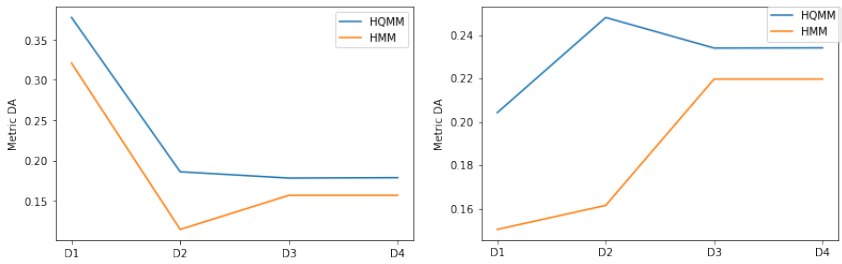


FIGURE 4.22: A comparison between HMMs and QHMMs according to the average DA metric. Figure at left for the training dataset and at right for the test dataset.

no-probable failure scenarios dataset, red color for the training dataset of the probable scenarios, young for the test dataset of the probable scenarios and green color for the test dataset of the no-probable scenarios. The same thing for the second system S_2, we show the results in Figure 4.24.

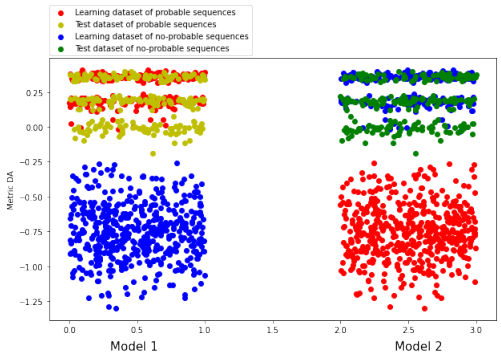


FIGURE 4.23: Results of the two models of the system S_1.

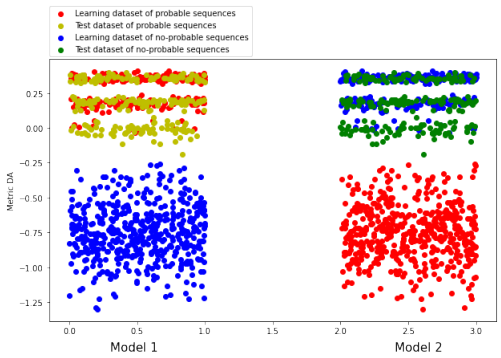


FIGURE 4.24: Results of the two models of the system S_2.

In these results, we can see clearly that Model 1 and Model 3 are able to detect the probable scenarios, and Model 2 and Model 4 are able to detect the no-probable scenarios for both systems. After that, we use those two models of each system in order to get the probable or not probable scenario, to do that, we calculate the metric of that sequence for each model and we decide whether the sequence is probable or not according to the biggest metric found. In addition, we use these two models of the system to directly generate the probable and no-probable scenarios from a given state of the system, instead of searching for new ones. To do this, we simply give the

current state of the system as input to the two models and generate the requested scenarios.

4.4 Conclusion

In this chapter, we have discussed one of the most important challenges in several fields, which is the problem of sequence processing. In the PSA area, the sequences represent the failure scenarios of systems. In order to process them in a reasonable time for the industry, we have prepared for each objective a quantum algorithm. The objective is to seek results more quickly in the near future with quantum computers. For this purpose, we have answered three general questions: How to find the failure scenarios of a system? how to calculate the similarity between them? how to create a generative model from a dataset of sequences?

For the first question, we use the graph of states of the system to represent all possible scenarios. In it, we look for paths between the current state of the system and the critical failure states. We proposed a quantum algorithm that finds all the paths between two vertices in a DAG with N qubits and M gates (N is the number of vertices of the graph and M is the number of edges). This algorithm also allows us to search for all the paths to several destinations from a source vertex at the same time. Unfortunately, due to the size of the quantum computers available today, it is not yet possible to use this algorithm alone to find these paths with a single quantum circuit. This is why we have proposed a second hybrid algorithm that allows to call the first algorithm to deal with large graphs. These two algorithms are tested in an IBM quantum simulator with 32 qubits, to show how well our algorithms work and also to show the effect of the convergence of our algorithm to a set of fixed paths, as opposed to the classical random walk method.

For the second question, the problem of calculating the similarity between two sequences. We proposed two quantum algorithms, the first one, Quantum Dynamic Time Warping (QDTW), is a quantum version of the DTW algorithm to compute the similarity between two sequences, by treating the scenarios element by element. This algorithm uses $N \times M$ qubits and $3N \times M$ gates to build the circuit that finds the warping path between two sequences of sizes N and M. The second algorithm QDTW-β allows us to find this similarity between two sequences using in addition to the case of one to one the case of treating the sub-sequences. These sub-sequences can take different sizes in each sequence, and the matching can be passed by two sub-sequences of different sizes. The biggest advantage of this algorithm is that we use only $N \times M$ qubits which allows us to have a speedup in finding the results compared to the classical approach (NP-complete approach). The number of gates increases depending on the β value. Both algorithms are tested and compared with the classical approaches. The first, QDTW gives the same results as the classical approach, except that here it is a quantum algorithm. The second QDTW-β, gives better results than the classical, either for finding the best matching or for classification.

For the third question, we have proposed a strategy to learn failure scenarios for PSA systems. We have proposed to use QHMMs models to generate failure scenarios from a given state of the system and also to identify the probable and no-probable failure scenarios of a system. This strategy gives us several advantages compared to the current approaches used in the field of PSA, among them, by doing the learning only

once and we can find the failure scenarios from any state of the system. In addition, it allows us to generate new scenarios that we do not have in the dataset. To test this approach, we have used four datasets for two small real systems. The results show that the QHMMs are more efficient than the HMMs. Furthermore, it shows that the models are able to detect probable and no-probable failure scenarios.

Chapter 5

Distance Estimation for Quantum Prototypes Based Clustering

Computing distances between vectors is a very important task in most areas of data science. In machine learning algorithms (clustering and classification), the computation of similarity is a key step and it requires a lot of complexity. On the other hand, the complexity of most machine learning algorithms is NP-complete, and finding solutions for these problems with less complexity is still an open question, particularly with the explosion of datasets in the world. Quantum computing is a field that has found its place these last years to solve these kinds of problems with less complexity, and it continues to have the attention of researchers in all fields, especially with the possibility of having quantum computers with very large numbers of qubits in the future. In several recent papers, we find more configurations to calculate the similarity between two vectors with quantum circuits. In this chapter we use our contribution in the paper [Ben+19] as a basis and compare these proposals in terms of stability to get a general idea about each approach. To clearly see the difference between each method, we improve the quantum k-means (QK-means) algorithm with a quantum update of the centroids. Also, we present how we can compute the distance between several vectors in a single quantum circuit. To measure the performance of the improved QK-means we propose a quantum version of the Davies-Bouldin index.

5.1 How to estimate the distances between the different data and centroids?

In the classical case, the distance calculation is very simple, in each specific domain, we can calculate a distance, for example for real vectors we use the Euclidean distance. The data are stored as vectors in the live memory of the computer (RAM), where we have easy access to it. In the quantum case, the data are encoded with quantum states, so how we can calculate the distance between two vectors encoded with quantum states? To answer this question, fortunately, there are a lot of papers that deal with this case using fidelity. In the following, we will present the quantum fidelity, and how we can compute it with quantum gates. And we will present several proposals to encode these classical vectors in quantum states.

Fidelity as a similarity measure

Since the 70's, there are some scientific papers [Alb83] [Uhl76] that focus on quantum fidelity. In the context of quantum communication, [Joz94] and [Sch95] are interested in quantum fidelity. For the quantum phase transition, we find [Gu10].

Suppose that we have two quantum states, $|\psi\rangle$ and $|\phi\rangle$, the fidelity between these two quantum states is calculated as follows:

$$Fid(|\psi\rangle, |\phi\rangle) = |\langle\psi|\phi\rangle|^2 \tag{5.1}$$

Lloyd et al in the paper [LMR13] demonstrate how can compute the Euclidean distance using the fidelity between two states. In order to begin, we give the properties of fidelity:

1. **Symmetry:** $Fid(|\psi\rangle, |\phi\rangle) = Fid(|\phi\rangle, |\psi\rangle)$.
2. **Bounded values**: The fidelity varies between 0 and 1, $0 \leq Fid(|\psi\rangle, |\phi\rangle) \leq 1$.

   If the states are orthogonal $Fid(|\psi\rangle, |\phi\rangle) = 0$ (that's to say, perfectly distinguishable), and $Fid(|\psi\rangle, |\phi\rangle) = 1$ if the states are identical.
3. **Invariance under unitary transformation:** $Fid(|U\psi\rangle, |U\phi\rangle) = Fid(|\phi\rangle, |\psi\rangle)$.

   This means that if we apply the same unitary transformation U to two quantum states this does not change their fidelity.
4. **Convexity:** For any three quantum states $|\psi\rangle$, $|\phi\rangle$ and $|\gamma\rangle$ and given p_1 and p_2 such that $p_1 + p_2 = 1$, then $Fid(|\psi\rangle, p_1|\phi\rangle + p_2|\gamma\rangle) = p_1 Fid(|\psi\rangle, |\phi\rangle) + p_2 Fid(|\psi\rangle, |\gamma\rangle)$.
5. **Multiplicativity:** $Fid(|\psi_1\rangle \otimes |\psi_2\rangle, |\psi_3\rangle \otimes |\psi_4\rangle) = Fid(|\psi_1\rangle, |\psi_3\rangle) Fid(|\psi_2\rangle, |\psi_4\rangle)$.

We can obtain the fidelity $|\langle\psi|\phi\rangle|$ [ABG06] of two quantum states $|\psi\rangle$ and $|\phi\rangle$ as a measure of similarity through the presented quantum circuit swap test.

SwapTest

The *SwapTest* circuit 5.1 has three inputs: a control qubit and two registers, the first register $|\alpha\rangle$ to represent the first vector α and the second $|\beta\rangle$ for the vector β. This

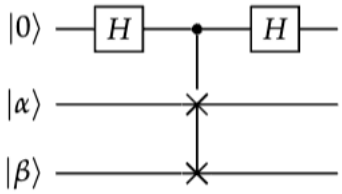


FIGURE 5.1: *SwapTest* Circuit

small circuit 5.1, allows to compute the overlap $\langle\alpha|\beta\rangle$ by measuring the control qubits. It has been used to compute the similarity between two vectors for the first time in the paper [ABG06]. To clearly understand how this circuit allows us to find the similarity between to vectors we describe in the following mathematical analysis.

Let us suppose that we have two vectors α and β represented by the two quantum states $|\alpha\rangle$ and $|\beta\rangle$ with n qubits for each one. The mathematical progress of the circuit is as follows:

1. As input of the circuit we have two quantum states $|\alpha\rangle$, $|\beta\rangle$, in addition, a control qubit initialized to the state $|0\rangle$ as follows:

$$|\psi_0\rangle = |0, \alpha, \beta\rangle \tag{5.2}$$

2. After the initialization, we apply the Hadamard gate to the first control qubit which gives:

$$|\psi_1\rangle = (H \otimes \mathbb{I}^{\otimes n} \otimes \mathbb{I}^{\otimes n}) |\psi_0\rangle = \frac{1}{\sqrt{2}}(|0,\alpha,\beta\rangle + |1,\alpha,\beta\rangle) \tag{5.3}$$

3. We apply the swap gate on the two registers $|\alpha\rangle$ and $|\beta\rangle$ under the control with $|1\rangle$ of the first qubit:

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0,\alpha,\beta\rangle + |1,\beta,\alpha\rangle) \tag{5.4}$$

4. We apply a second Hadamard to the control qubit:

$$|\psi_3\rangle = \frac{1}{2}|0\rangle (|\alpha,\beta\rangle + |\beta,\alpha\rangle) + \frac{1}{2}|1\rangle (|\alpha,\beta\rangle - |\beta,\alpha\rangle) \tag{5.5}$$

5. We measure the control qubit, and the probability of measuring the state $|0\rangle$ is computed as follows:

$$P(|0\rangle) = |\frac{1}{2}\langle 0|0\rangle (|\alpha,\beta\rangle + |\beta,\alpha\rangle) + \frac{1}{2}\langle 0|1\rangle (|\alpha,\beta\rangle - |\beta,\alpha\rangle)|^2 \tag{5.6}$$

$$= \frac{1}{4}|(|\alpha,\beta\rangle + |\beta,\alpha\rangle)|^2 \tag{5.7}$$

$$= \frac{1}{4}(\langle\beta|\beta\rangle\langle\alpha|\alpha\rangle + \langle\beta|\alpha\rangle\langle\alpha|\beta\rangle + \langle\alpha|\beta\rangle\langle\beta|\alpha\rangle + \langle\alpha|\alpha\rangle\langle\beta|\beta\rangle) \tag{5.8}$$

$$= \frac{1}{2} + \frac{1}{2}|\langle\alpha|\beta\rangle|^2 \tag{5.9}$$

According to the probability $P(|0\rangle)$, we can decide that $|\alpha\rangle$ and $|\beta\rangle$ are orthogonal if $P(|0\rangle) = 0.5$ and identical if $P(|0\rangle) = 1$.

Lloyd et al in the paper [LMR13] demonstrate how can calculate the Euclidean distance using SwapTest, the method is named DistCalc and described as follows:

1. We encode each vector x in a quantum state as [NDW16] [Mot+04]:

$$|x|^{-1}x \to |x\rangle = \sum_{i=1}^{M} |x|^{-1}x_i |i\rangle \tag{5.10}$$

with this encoding we use only $log_2 M$ qubits with M is the dimension of x, and the state $|x\rangle$ is normalized because $\langle x|x\rangle = |x|^{-2}x^2 = 1$.

2. In order to use *SwapTest* we initialize the two states as follows:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0,\alpha\rangle + |1,\beta\rangle) \tag{5.11}$$

$$|\phi\rangle = \frac{1}{\sqrt{Z}}(|\alpha|\,|0\rangle + |\beta|\,|1\rangle) \tag{5.12}$$

with $Z = |\alpha|^2 + |\beta|^2$

3. We use *SwapTest* and we calculate the Euclidean distance as follows:

$$|a-b|^2 = 2Z|\langle\phi|\psi\rangle|^2 \tag{5.13}$$

Here Lloyd et al are prepare the inputs of the *SwapTest* with their configuration, this initialization can be changed with other methods, in the following we introduce, other approaches to initialize these inputs.

States construction to estimate the distance-type measurements

Building quantum states from classical data is the first step in order to profiting from the power of quantum computers. This construction of states must be done in a clean way that keeps the information clean and easy to use. For the case of calculating the similarity between two vectors, there are several methods to build the two inputs of the *SwapTest* circuit. Here we present three different methods for the preparation and construction of $|\psi\rangle$ and $|\phi\rangle$ states.

Wiebe et al. approach

1. Data Preparation

   Given $N = 2^n$ dimensional complex vectors $\vec{x}$ and $\vec{w}$ with components $x_j = |x_j|e^{-i\alpha_j}$ and $w_j = |w_j|e^{-i\beta_j}$ respectively. Assume that $\{|x_j|, \alpha_j\}$ and $\{|w_j|, \beta_j\}$ are stored as floating point numbers in quantum random access memory.

2. Construction of the states

   An alternative representation of the states was suggested by Wiebe, Kapoor and Svore [WKS18]. It aims to write the parameters into amplitudes of the quantum states.

$$|\psi\rangle = \frac{1}{\sqrt{d}}\sum_j |j\rangle \left(\sqrt{1-\frac{|x_j|^2}{r_{max}^2}}e^{-i\alpha_j}|0\rangle + \frac{x_j}{r_{max}}|1\rangle\right)|1\rangle$$

$$|\phi\rangle = \frac{1}{\sqrt{d}}\sum_j |j\rangle|1\rangle \left(\sqrt{1-\frac{|w_j|^2}{r_{max}^2}}e^{-i\beta_j}|0\rangle + \frac{w_j}{r_{max}}|1\rangle\right)$$

   Where $j = \{1, ..., n\}$, and r_{max} is an upper bound on the maximum value of any feature in the dataset. The input vectors are d-sparse, i.e., contain no more than d non-zero entries.

   Using the swap test, the inner product is evaluated by:

$$d_{q_1}(|x\rangle, |w\rangle) = d^2 r_{max}^4 (2P(|0\rangle) - 1) \tag{5.14}$$

Lloyd et al. approach

1. Data Preparation

   For the future of quantum machine learning, it may be vital to find quantum ways of representing and extracting information. To use the forces of quantum mechanics without being limited to the classical ideas of data encoding; Lloyd, Mohseni, and Rebentrost [LMR13] proposed a way to encode the classical vectors into a quantum state.

Consider $N = 2^n$ dimensional complex vectors $\vec{x}$ and $\vec{w}$, we have:

$$|x\rangle = \frac{\vec{x}}{|\vec{x}|} \ , \ |w\rangle = \frac{\vec{w}}{|\vec{w}|}$$

2. Construction of the states

   Seth Lloyd and his co-workers proposed a way to construct the state $|\psi\rangle$ and $|\phi\rangle$. The idea is to adjoin an ancillary qubit to states creating an entangled state $|\psi\rangle$. The greater the difference between the states $|x\rangle$ and $|w\rangle$, the more the resulting state is entangled [Cai+15].

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle\,|x\rangle + |1\rangle\,|w\rangle)$$

$$|\phi\rangle = \frac{1}{\sqrt{Z}}(|\vec{x}|\,|0\rangle - |\vec{w}|\,|1\rangle)$$

   Where $Z = |\vec{x}|^2 + |\vec{w}|^2$

   After applying the *SwapTest* circuit, the distance is evaluated by:

$$d_{q_2}(|x\rangle, |w\rangle) = 2Z(2P(|0\rangle) - 1) \tag{5.15}$$

We can verify that the desired distance $|\vec{x} - \vec{w}| = \sqrt{2Z|\,\langle\phi|\psi\rangle\,|^2}$.

$$\begin{aligned}
|\,\langle\psi|\phi\rangle\,|^2 &= \frac{1}{2Z}\,|(|\vec{x}|\,\langle 0| - |w|\,\langle 1|)(|0\rangle\,|x\rangle + |1\rangle\,|w\rangle)|^2 \\
&= \frac{1}{2Z}\,||\vec{x}|\,\langle 0|0\rangle\,|x\rangle + |\vec{x}|\,\langle 0|1\rangle\,|w\rangle - |w|\,\langle 1|0\rangle)\,|x\rangle - |w|\,\langle 1|1\rangle)\,|w\rangle|^2 \\
&= \frac{1}{2Z}\,|(|\vec{x}|\,|x\rangle - |\vec{w}|\,|w\rangle)|^2 \\
&= \frac{1}{2Z}|\vec{x} - \vec{w}|^2
\end{aligned}$$

The desired distance is $d_{q_2} = 2Z(2P(0) - 1)$, we can see that if vector $\vec{x}$ is identical to the centroid of the cluster w the state $|\psi\rangle$ is orthogonal to $|\phi\rangle$ and the success probability of projective measurement is $P(0) = 1/2$, therefore the distance $d_{q_2} = 0$.

Anagolum approach

1. Data Preparation

   For simplification, we assume that we are in 2-dimensional space. Let's consider that we have two vectors $\vec{x}(x_0, x_1)$ and $\vec{w}(w_0, w_1)$.

   We can map data values to θ and α values using these equations.

   For x we get:

$$\alpha_0 = (x_0 + 1)\frac{\pi}{2} \ , \ \theta_0 = (x_1 + 1)\frac{\pi}{2} \tag{5.16}$$

   Similarly for w we get:

$$\alpha_1 = (w_0 + 1)\frac{\pi}{2} \ , \ \theta_1 = (w_1 + 1)\frac{\pi}{2} \tag{5.17}$$

2. Construction of the states

To construct the two states $|\psi\rangle$ and $|\phi\rangle$ we use U gate as follows:

$$|\psi\rangle = U(\theta_0, \alpha_0, 0)|0\rangle \tag{5.18}$$

$$|\phi\rangle = U(\theta_1, \alpha_1, 0)|0\rangle \tag{5.19}$$

Indeed, U gate implements the rotations we need to perform to encode our data points.

$$U(\theta, \alpha, \lambda) = \begin{pmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\alpha}sin\frac{\theta}{2} & e^{i\lambda+i\alpha}\cos\frac{\theta}{2} \end{pmatrix}$$

This instruction would cause the qubit to move θ radians away from the positive z-axis, and α radians away from the positive x-axis.

Using the swap test, the distance is evaluated by:

$$d_{q_3}(|x\rangle, |w\rangle) = P(|1\rangle) \tag{5.20}$$

5.2 How to search for the closest centroid to a given data?

As we have already explained in chapter 2, Grover's algorithm is one of the most successful in the field of quantum computing, is an algorithm that allows to search for an element in a dataset with a complexity $O(\sqrt{N})$ instead of $O(N)$ in the classical case. Therefore, to minimize the complexity of searching the nearest centroid in the quantum k-means algorithm we use this algorithm for the search of the smallest distance. In this section, we first present Grover's algorithm in detail, and then we explain the approach proposed by [DH96] to find the nearest centroid using Grover's algorithm.

5.2.1 Grover's Algorithm

In quantum computing, Grover's algorithm allows to search for one or more elements in an unsorted dataset with N elements in $O(\sqrt{N})$ time. Grover's algorithm begins with a quantum register of n qubit initialized to $|0\rangle$ when n is the number necessary to represent the search space, we have $2^n = N$ which means $|0\rangle^{\otimes n} = |0\rangle$.

Equal superposition: The first step is to apply the Hadamard transform $H^{\otimes n}$ to put the system into an equal superposition of states:

$$|\psi\rangle = H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle$$

Quantum oracle O**:** An oracle is a black-box function and a quantum oracle is a quantum black box, which means that it can observe and modify the system without collapsing it to a classical state. It will recognize if the system is in the correct state: if the system is in the correct state, then the oracle will rotate the phase by π radians, otherwise, it will do nothing. To create the circuit that represents this oracle it is

necessary to implement this transformation of $|x\rangle$:

$$|x\rangle \rightarrow (-1)^{f(x)}|x\rangle$$

where $f(x) = 1$ if x is the correct state, and $f(x) = 0$ otherwise.

Diffusion transform: It performs inversion about the average, transforming the amplitude of each state so that it is as far above the average as it was below the average before the transformation, and vice versa. This part consists of another application of the Hadamard transform $H^{\otimes n}$, followed by a conditional phase shift that shifts every state except $|0\rangle$ by -1, followed by another Hadamard transform. The *diffusion transform* can be represented by this equation, using the notation $|\psi\rangle$:

$$H^{\otimes n}[2|0\rangle\langle 0| - I]H^{\otimes n} = 2H^{\otimes n}|0\rangle\langle 0|H^{\otimes n} - I = 2|\psi\rangle\langle\psi| - I$$

Giving the entire Grover iteration:

$$[2|\psi\rangle\langle\psi| - I]O$$

The total run-time of a single Grover iteration is $\Theta(2n)$ from the two Hadamard transforms, plus the cost of applying $O(n)$ gates to perform the conditional phase shift, is $O(n)$. It follows that the runtime of Grover's entire algorithm, performing $O(\sqrt{N}) = O(2^{\frac{n}{2}})$ iterations each with a runtime of $O(n)$, is $O(2^{\frac{n}{2}})$.

Algorithm 13: Grover's algorithm

Input: A quantum oracle O which performs the operation $O|x\rangle = (-1)^{f(x)}|x\rangle$, where $f(x) = 0$ for all $0 \leq x < 2^n$ except x_0, for which $f(x_0) = 1$, n qubits initialized to the state $|0\rangle$

Runtime: $O(\sqrt{N})$ operations, with $O(1)$ probability of success.

Output: x_0

Procedure:

Initial state

$$|0\rangle^{\otimes n}$$

apply the Hadamard transform to all qubits

$$H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle = |\psi\rangle$$

apply the Grover iteration $R \approx \frac{\pi}{4}\sqrt{2^n}$ *times*

$$[(2|\psi\rangle\langle\psi| - I)O]^R|\psi\rangle \approx |x_0\rangle$$

measure the register x_0

5.2.2 Search for the minimal with Grover's algorithm

In the literature, Grover's algorithm is used to solve optimization problems, it is used to find the global minimum of an optimization problem. The problem is represented by a function f(X) and with Grover's algorithm we look for the global minimum that solves this problem. For us, we are interested in the version that deals with the case

of finding the minimum proposed by the paper [DH96]. In this paper, the authors use Grover's search algorithm with quantum oracles indicating which elements are smaller than an arbitrary threshold.

Thanks to quantum parallelism, the global minimum is always found, which has great implications for machine learning procedures, as they often get blocked in local optima.

Let's assume that we have an optimization problem represented by the following objective function:

$$f(x) : \{0,1\}^n \rightarrow \mathbb{R} \tag{5.21}$$

Let us suppose that we also have a quantum oracle O which acts on n qubits. The steps of the algorithm to find the minimum of $f(x)$ are described as follows:

1. The first step is to choose an input x_1 in a random way where $y_1 = f(x_1)$
2. In the quantum circuit, we will process the inputs (the x_i) and the corresponding inputs by the function $f(x)$ (the y_i). Then the algorithm needs two quantum registers: the first one to encode the x_i and the second one to encode the y_i. The first register contains n qubits and the second one m qubits.
3. In order to create a quantum superposition that represents all possible inputs in the first register, we apply Hadamard gates on all qubits of the first register.
4. We apply Grover's algorithm on the first register and we find $y = f(x)$.
5. In a classical computer, we use y to model y_i, if $y < y_i$ then $x_i = x$.

Following [AK99] [DH96], we repeat items 2-5 $\sqrt{2^n}$ times to find the global minimum..

5.3 Comparison of different quantum distances

In this section, we compare the different quantum distances explained before in terms of relevance and stability by answering these two questions through some empirical evaluations:

- Which quantum distance has a high probability of finding the right nearest center?
- Which quantum distance has a high probability of finding the nearest centers in the right order with a good stability?

5.3.1 Which quantum distance has a high probability of finding the right nearest center?

Because of the noise of quantum computers, the distance between two states is hard to compute as we will get a probabilistic result; the distance is unstable. However, it's easier to assign data points to different groups because we don't need the exact distances to each one but only the closest centroid. Thus, we can just put the new data point in the cluster associated with the smallest value that our parameter takes. To illustrate more our idea, we gave an example of two distributions. Figure 5.2 and

Figure 5.3 represent two distributions where the black dot X is the test data and the three crosses are the centers ($C1$, $C2$, $C3$).

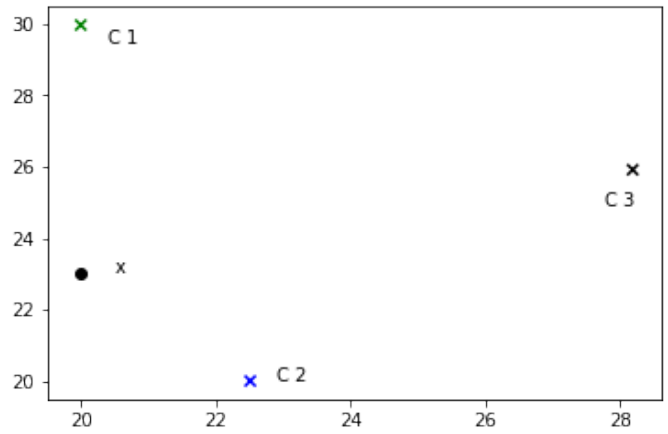


FIGURE 5.2: Distribution 1

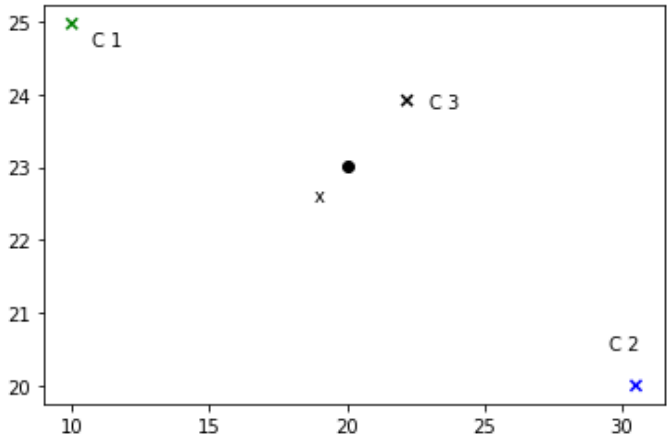


FIGURE 5.3: Distribution 2

From table 5.1, we can notice that the distance changes from an iteration to another but the assignment to the closest centroid is correct. Comparing the three approaches in terms of the confidence interval, we can notice that the Wiebe et al. approach gives a higher confidence interval in a time lower than other approaches.

Distance	Dist	Green	Blue	Black	Probability of success
Wiebe et al. approach (d_{q_1})	1	665 times	9322 times	13 times	[92.71%, 93.70%]
	2	0 times	0 times	10000 times	[99.96%, 100%]
Lloyd et al. approach (d_{q_2})	1	2190 times	7620 times	190 times	[75.35%, 77.02%]
	2	0 times	515 times	9485 times	[94.40%, 95.26%]
Anagolum approach (d_{q_3})	1	2722 times	6530 times	748 times	[64.36%, 66.22%]
	2	2486 times	2175 times	5339 times	[52.41%, 54.36%]

TABLE 5.1: Distance-types Comparison

5.3.2 Which quantum distance has a high probability of finding the nearest centers in the right order with a good stability?

After analyzing the performance of the different quantum distances in terms of the stability of the values allowing the choice of the right center, we will study the behavior of these quantum distances, but this time in terms of the stability of the order of the nearest centers. In other words, how far away is it possible to find the nearest centers in the right order whatever the iteration? To do this, we carried out 10,000 searches for the nearest centers for the two distributions studied. We analyzed the stability of the order of the nearest centers found by each quantum distance. The results show that the distance d_{q_1} is the best one which offers a very good stability in the order of the nearest centers in the case of the two distributions studied. As shown in table 5.1, the distance d_{q_1} shows a very good stability in the order of the nearest centers compared to the other two quantum distances. For distribution 1, the correct order of the nearest centers is [$C2$ $C1$ $C3$]. The distance d_{q_1} finds this order with a probability of 85.32% (8532 times out of 10,000 searches), while the distance d_{q_2} and d_{q_3} find the order with a probability of 53.26% (5326/10,000) and 26.10% (261/10,000) respectively. In the case of distribution 2, the situation is more complicated because the test point is almost halfway between the two centers. This situation is confirmed by the results obtained in Figure 5.5. Indeed, the distance d_{q_1} always finds the right order of the nearest centers [$C3$ $C1$ $C2$]. Nevertheless, this distance continues to provide the right solution but the order changes significantly [$C3$ $C2$ $C1$]. Compared to the other two quantum distances, the distance d_{q_1} seems much more stable in the order of the nearest centers. As can be seen in both Figure 5.4 and Figure 5.5, the other two quantum distances d_{q_2} and d_{q_3} change order quite often compared to the distance d_{q_1}. Order stability is a very relevant information on the behavior of quantum distances.

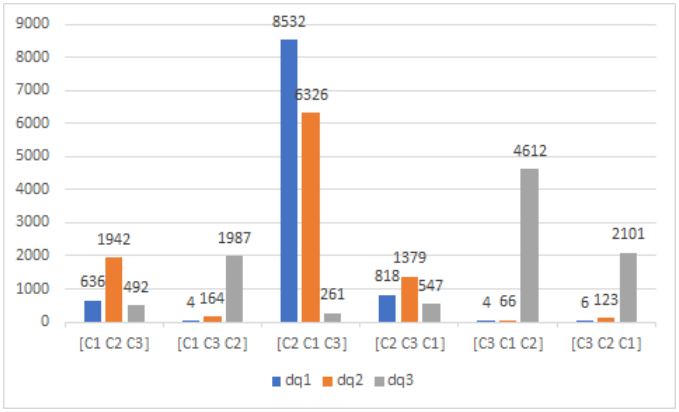


FIGURE 5.4: Order of belonging distribution 1

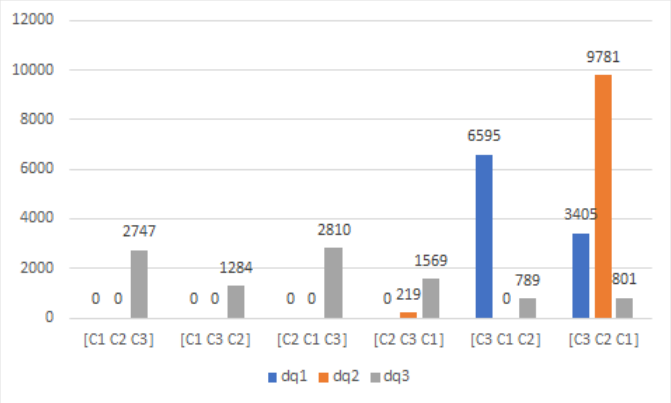


FIGURE 5.5: Order of belonging distribution 2

5.4 Validation criteria

As we have already described, the notion of distance in quantum mechanics is not stable. Therefore, the estimation of the distance in the classical version it's not the same as in the quantum version. For example, if we take the distance between two quantum states, after each measurement we get different values but the belonging of the state to a cluster it would be all the time the same. This is a totally different way of thinking about the way we estimate distances between points which can make the quantum clustering algorithms faster than its classical counterpart.

Our purpose is to efficiently define the validation criteria in the context of quantum learning in order to evaluate the goodness of clustering in quantum algorithms.

One of the most important considerations regarding the machine learning model is assessing its performance which means the quality of the model. In the case of supervised learning algorithms, evaluating the quality of the model is easy because we already have labels for every example. However, in the case of unsupervised learning algorithms, we are not that much blessed because we deal with unlabeled data. But still, we have some metrics that give the practitioner insight into the happening of change in clusters depending on the algorithm.

Indeed, there are several indices that are used to measure cluster validity. In our case, we chose the Davies Bouldin index which is an internal index that used clustering and the underlying data set to assess the quality of the clustering. It places similar objects in the same cluster and dissimilar objects in different clusters. In what follows, we give the Davies-Bouldin (DB) index as a criteria of validation, and we transform the classical Davies-Bouldin index into the quantum version; quantum Davies-Bouldin (QDB) index. This transformation is basically done by transforming the classical distance to the quantum version.

In fact, a good clustering algorithm aims to create clusters whose:

- The intra-cluster similarity is high (The data that is present inside the cluster is similar to one another)
- The inter-cluster similarity is less (Each cluster holds information that isn't similar to the other)

5.4.1 Classical Davies-Bouldin index

The Davies Bouldin index introduced by David L. Davies and Donald W. Bouldin in 1979 is a metric for evaluating clustering algorithms [DB79] that can be calculated with the following formula:

$$DB = \frac{1}{K}\sum_{k=1}^{K}\max_{k\neq k'}\frac{d_n(w_k)+d_n(w_{k'})}{d(w_k,w_{k'})}, \tag{5.22}$$

where K is the number of clusters, d_n is the average distance of all elements from the cluster C_k to their cluster centre w_k, $d(w_k, w_{k'})$ is the distance between clusters centres w_k and $w_{k'}$. Just like the Silhouette score, Calinski-Harabasz index, and Dunn index, the Davies-Bouldin index provides an internal evaluation schema. I.e. the score is based on the cluster itself and not on external knowledge such as labels. This index well evaluates the quality of unsupervised clustering because it's based on the ratio of the sum of within-clusters scatter to between-clusters separation. More the value of DB is lower, means that we have a better cluster. The main objective is to evaluate how well the clustering has been done.

5.4.2 Quantum Davies-Bouldin index

As we have already mentioned before, the notion of distance in quantum approaches is different from the classical case. Quantum distance does not need to be proportional to the real distance, but only has a positive correlation with it. We need only the nearest centroid, not the exact value of the real distance. To evaluate the quality of quantum clustering with a Davies-Bouldin index-type based on intra- and inter-cluster distances, we propose to adapt it to the quantum case. To do this, we will define the Quantum Davies-Bouldin (QDB) quality index as follows:

$$QDB = \frac{1}{K}\sum_{k=1}^{K}\max_{k\neq k'}\frac{\delta_n(w_k)+\delta_n(w_{k'})}{\delta(w_k,w_{k'})}, \tag{5.23}$$

where

$$\delta_n(w_k) = \frac{1}{|C_k|}\sum_{i=1}^{|C_k|} d_{q_2}(|x_i\rangle_{x_i\in C_k}, |w_k\rangle) \quad \text{and} \quad \delta(w_k,w_{k'}) = d_{q_2}(|w_k\rangle, |w_{k'}\rangle)$$

This index has been done by transforming the classical distances to the quantum one as we have explained previously. Where K is the number of clusters, δ_n is the quantum version of the distance d_n and $\delta(w_k, w_{k'})$ is the quantum distance between clusters centres w_k and $w_{k'}$.

In fact, the estimation of Davies-Bouldin in the context of quantum learning is totally different from the classical learning.

While in the classical version, the Davies Bouldin index gives a fixed value, in quantum learning the quantum DB will give different values as the distance change from an iteration to another. Therefore, instead of having only one value of DB, we will get an interval of the QDB, this interval is due to the probabilistic nature of the

qubits and it represents the different variations of this index while it is in the quantum state.

5.5 Classical K-means

The k-means clustering [Mac+67] is a type of unsupervised clustering, one of the most widely used clustering methods early developed by Lloyd [LLo82]. Let $X = \{x_1, x_2, \ldots, x_N\}$ be the data set, each row vector $x_n \in \mathbb{R}^M$, $1 \leq n \leq N$ is composed of M attributes (features). The k-means clustering allows to divide the set of data X into K clusters $C = \{C_1, \ldots, C_K\}$ by minimizing the following sum of squared errors:

$$R = \sum_{k=1}^{K} \sum_{n \in C_k} \|x_n - w_k\|^2 = \sum_{k=1}^{K} \sum_{n=1}^{N} g_{nk} \|x_n - w_k\|^2, \tag{5.24}$$

where $w_k \in \mathbb{R}^M$ is the centroid of the data within the cluster C_k and $G \in \mathbb{R}_+^{N \times K}$ is the binary classification matrix defined by $g_{nk} = 1$, if the data $x_n \in C_k$, and 0 otherwise. Firstly, the k-means algorithm initialize randomly K centroids $w_1, w_2, ..., w_K \in \mathbb{R}^M$. Then the algorithm usually iteratively unfolds in two phases:

1. at first we go over every point and assign each to the cluster of the nearest centroid.

2. at the second phase, the centroid of each cluster is updated.

The algorithm converges when there is no further change in the assignment of instances to clusters. The idea behind k-means clustering is very natural. It just puts every new data point you ask it to classify into the group that it is closest to.

Algorithm 14: k-means algorithm

Input: Set of vectors $x_n \in \mathbb{R}^M, n = \{1, 2, ..., N\}$, initial centroids $w_1, w_2, ..., w_K \in \mathbb{R}^M$.

Output: The set of K clusters C_k, $|C_k|$ is the number of vectors within the cluster k.

repeat

Assignment step (clustering): Assign each data to the cluster C_{k^*}, k^* is computed by:

$$k^* = \underset{k \in \{1,2,...,K\}}{\operatorname{argmin}} \|x_n - w_k\|^2$$

Update step: For all $k = \{1, 2, ..., K\}$, update the centroid w_k of each cluster C_k by:

$$w_k = \frac{1}{|C_k|} \sum_{n=1}^{N} g_{nk} x_n$$

until a stopping condition is satisfied.

5.6 Quantum k-means clustering

The first version of the quantum k-means algorithm was proposed by Lloyd et al in their paper [LMR13], where they use their method to construct quantum states to calculate the distance between data and centers. In the quantum case, with the

quantum fidelity computation and the quantum distance finding, we have achieved an exponential speedup compared to the classical case depending on M. In the classical case, we need a complexity $O(M)$ to compute the distance between vectors instead of $O(log_2 M)$ quantically. So, how can we get this speedup also depending on the size of the dataset N? Here, we show how we can find this speedup depending on the size of the dataset N, and propose an upgrade of the quantum version of k-means proposed by [LMR13] with a quantum centroid upgrading.

5.6.1 Data preparation and states construction

Normally, there are several methods to prepare the data and construct the states. According to the previous section, the method of Wiebe, Kapoor, and Svore [WKS18] gives good results in terms of clustering and also stability.

As we have already shown in the previous section, in order to find more stable results, it is better to use the method proposed by [WKS18] in order to build the quantum states for the classical data. Then we can choose this method to build the quantum states of data in order to do clustering using the quantum k-means algorithm. The idea behind the method proposed by [WKS18] is to use a quantum circuit for each observation and center, here we will propose an improvement of this to calculate the distance between all the observations and all the centers in a single circuit.

Suppose that we have N observations and K centers. To be able to calculate all the distances between all the N observations and all the K centers with a single circuit, we need to code everything in a single circuit. In total we have $N * K$ distance to find, so we add the number of qubits that allows us to index these possibilities. Then, we use three registers of qubits, the first one $|\psi_1\rangle$ to encode the indexes, $|\psi_2\rangle$ to encode observations, $|\psi_3\rangle$ to encode the centers. We reserve, $log_2(N * K)$ qubits for $|\psi_1\rangle$, $log_2(M)$ qubits for $|\psi_2\rangle$ and $log_2(M)$ for $|\psi_3\rangle$. These registers are encoded as follows:

$$|\psi\rangle = |\psi_1\rangle |\psi_2\rangle |\psi_3\rangle |c\rangle = \sum_{i=0}^{N} \sum_{j=0}^{k} |i * K + j\rangle |x_i\rangle |c_j\rangle |0\rangle \tag{5.25}$$

Then we apply the *SwapTest* circuit on the last two registers with the control of the last qubit. After that, we measure the first register and the last qubit. The general circuit is represented as follows:

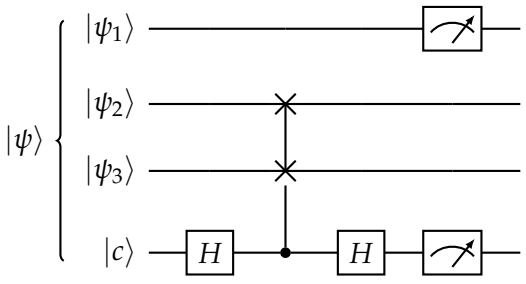


FIGURE 5.6: Calculate the distance between several observations

In order to get the distances, for each state of the superposition, we take the probability to be in the state $|0\rangle$ for the last qubit. The states of the qubits in register $|\psi_1\rangle$

are the indices and the last qubit $|c\rangle$ is the result of the inner product between an observation and a center related to each index. The output superposition is displayed as follows:

$$|\psi_1\rangle|c\rangle = \sum_{j=0}^{N*K} \frac{1}{\sqrt{2^{N*k}}} |j\rangle |c\rangle \tag{5.26}$$

$$= \sum_{j=0}^{N*K} \frac{1}{\sqrt{2^{N*k}}} |j\rangle (\alpha_{\lfloor \frac{j}{N} \rfloor, j-\lfloor \frac{j}{N} \rfloor} |0\rangle + \beta_{\lfloor \frac{j}{N} \rfloor, j-\lfloor \frac{j}{N} \rfloor} |1\rangle) \tag{5.27}$$

where $\lfloor x \rfloor$ is the integer part of x, $\alpha_{\lfloor \frac{j}{N} \rfloor, j-\lfloor \frac{j}{N} \rfloor}$ is the probability of being in state $|0\rangle$ for the qubit $|c\rangle$ related to the observation $\lfloor \frac{j}{N} \rfloor$ and center $j - \lfloor \frac{j}{N} \rfloor$. We use the $\alpha_{\lfloor \frac{j}{N} \rfloor, j-\lfloor \frac{j}{N} \rfloor}$ for all $j = 0, \ldots, N * k$ and the equation (5.14) to find all the distances between observations and centers.

5.6.2 Cluster assignment

After computing the distance between each training state and each cluster centroid. We assign each state $|x_n\rangle$ to the closest centroid $|w_k\rangle$ by using Grover's algorithm [Gro98] and the algorithm proposed by [DH96] for the minimization problems. More precisely, we should find the solution to the following minimization problem:

$$\underset{w}{\operatorname{argmin}} D(|x\rangle, |w\rangle) = \underset{C}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{|x_n\rangle \in C_k} d_q^2(|x_n\rangle, |w_k\rangle) \tag{5.28}$$

While the best classical algorithms for a search over unordered data requires $\mathcal{O}(N)$ time, Grover's algorithm performs the search on a quantum computer in only $\mathcal{O}(\sqrt{N})$ operations, which means a quadratic speed-up over its classical version. This speed is done thanks to the superposition of states, in other words, the search is done globally, which means a significant improvement in optimization routines.

5.6.3 Update the centroid

Assume that we have a set of quantum states $|X\rangle = \{|x_n\rangle \in \mathbb{C}^M, n = 1, ..., N\}$, and a set of K clusters C_k, $|C_k|$ is the number of vectors within the cluster k. k-means clustering aims to partition the N observations into K clusters C_k with $|W\rangle = |w_1\rangle, |w_2\rangle, ..., |w_K\rangle$ centroids, so as to minimize the within-cluster variance. Formally, the objective is to find a solution for the minimization problem (5.28).

We compute the distance between each training state and each cluster centroid using the *SwapTest* circuit. Then, we assign each state to the closest centroid using Grover's algorithm as explained before.

The second step of QK-means is updating the centroid of each cluster. To do so, the updated centroid of each cluster is given by:

$$|w_k^{(t+1)}\rangle = |(Y_k^{(t)})^T X\rangle \tag{5.29}$$

where

$$|Y_k^{(t)}\rangle = \frac{1}{\sqrt{|C_k|}} \sum_{n=1}^{N} y_{nk} |n\rangle \quad \text{and} \quad y_{nk} = \begin{cases} 1 & \text{if } x_n \in C_k \\ 0 & \text{else} \end{cases}$$

We give the main steps of the proposed algorithm in the following. The distance $d_{q_i}(|x_n\rangle, |w_k\rangle)$ is at the user's choice, in our case we opt for the distance d_{q_1} as it gives the best result. For the stopping criterion, we use the relative distortion between two iterations with respect to a threshold ϵ.

The algorithm of QK-means becomes as the following:

Algorithm 15: Quantum k-means algorithm

Input: $|X\rangle = \{|x_n\rangle \in \mathbb{C}^M, n = 1, ..., N\}$, K number of clusters C_k, initial centroids of the clusters at $t = 0$: $|w_1^{(0)}\rangle, |w_2^{(0)}\rangle, ..., |w_K^{(0)}\rangle$.

Output: K clusters C_k.

repeat

Assignment step (clustering): Each data is assigned to the cluster with the nearest center using Grover's search:

$$C_k^{(t)} \longleftarrow \{|x_n\rangle : d_{q_i}^2(|x_n\rangle, |w_k^{(t)}\rangle) \leq d_{q_i}^2(|x_n\rangle, |w_j^{(t)}\rangle), \forall j, 1 \leq j \leq K\}$$

where each $|x_n\rangle \in |X\rangle$ is assigned to exactly one $C_k^{(t)}$, even if it could be assigned to more of them.

Update step: The center of each cluster C_k is recalculated as being the average of all data belonging to this cluster (following the previous assignment step):

$$|w_k^{(t+1)}\rangle \longleftarrow |(Y_k^{(t)})^T X\rangle$$

until Convergence is reached

Convergence can be considered as achieved if the relative value of the distortion level $D(|x\rangle, |w^{(t)}\rangle)$ falls below a small prefixed threshold ϵ:

$$\frac{D(|x\rangle, |w^{(t-1)}\rangle) - D(|x\rangle, |w^{(t)}\rangle)}{D(|x\rangle, |w^{(t)}\rangle)} < \epsilon$$

5.7 Experimental results

5.7.1 Datasets

The classical and quantum version of k-means was tested on three real world datasets available for public use in the UCI Machine learning repository [DG17].

- *Iris* - Iris data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.
- *Wine* - Wine is a dataset that is related to a chemical analysis of wines grown in the same region in Italy but derived from different cultivars.
- *Wisconsin Diagnostic Breast Cancer (WDBC)* - This data has 569 instances with 32 variables (ID, diagnosis, 30 real-valued input variables). Each data observation is labeled as benign (357) or malignant (212).

5.7.2 Clustering through quantum K-means

We used three different datasets to show the experimental results of QK-means. Figures 5.7, 5.8, and 5.9 represent the projection of the datasets iris, wine, and breast

cancer respectively using the principal component analysis. We can notice that the algorithm of *QK*-means has identified the different clusters (groups) which are significantly different (distant) from each other. Therefore, the quantum *k*-means gives a good classification just like its classical version, but the advantage of the quantum version is that it can deal with high dimensional spaces in a time much more quickly than the classical version, which is crucial in nowadays.

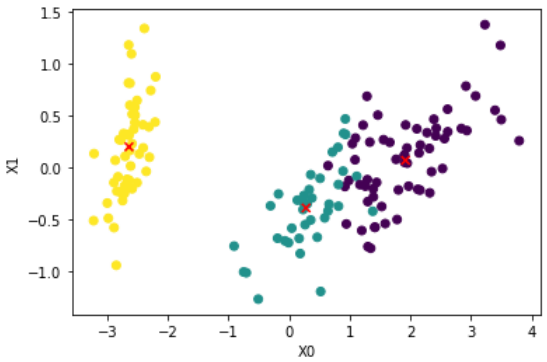


FIGURE 5.7: *QK*-means clustering on Iris data

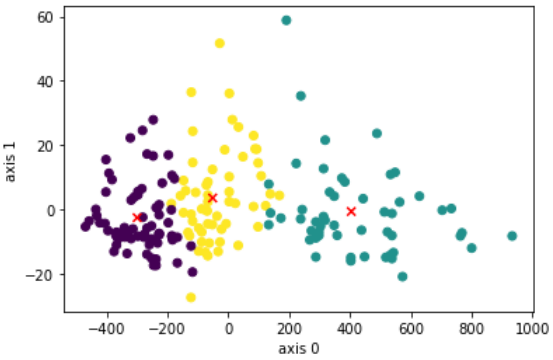


FIGURE 5.8: *QK*-means clustering on Wine data

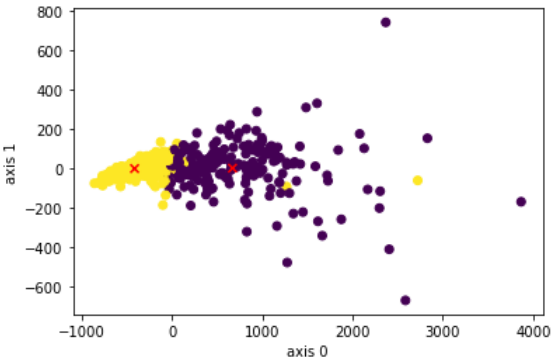


FIGURE 5.9: *QK*-means clustering on Breast Cancer data

For each data set we compare the Davies-Bouldin (DB) index for both the classical and the quantum version of *K*-means. These results are represented in table 5.2. DB

and QDB indexes are not calculated with the same distances. Direct comparison is therefore difficult, but we can see that QDB shows a decreasing behavior during different iterations of the learning process, indicating an improvement in the quality of quantum clustering. We can therefore consider that QDB is a good quality indicator for quantum clustering.

Dataset	*K*-means (DB)	*QK*-means (QDB)
Iris	0.66	[0.37, 0.56]
wine	0.53	[0.40, 0.59]
Breast Cancer	0.50	[0.38, 0.57]

TABLE 5.2: *k*-means & *QK*-means using DB index

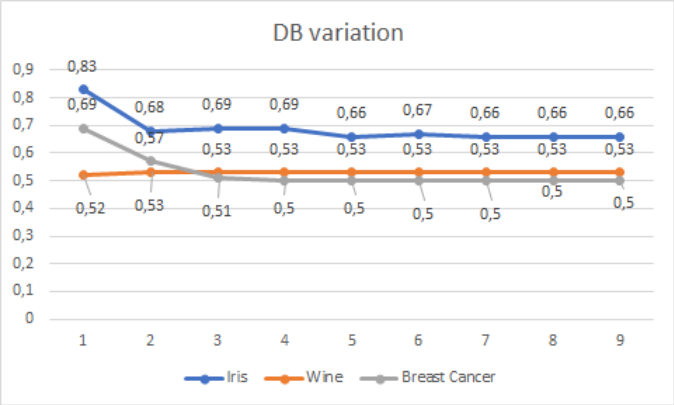


FIGURE 5.10: Davies-Bouldin Variation

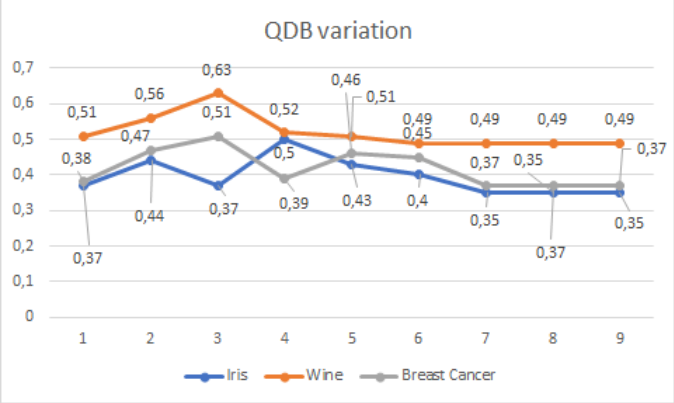


FIGURE 5.11: QDavies-Bouldin Variation

5.8 Computational time complexity

Let us consider the *k*-means problem of assigning N vectors to K clusters such that the average distance from each cluster centroid to all points of the cluster is minimized.

After randomly choosing the initial centroids, the standard method to solve this minimization problem supposes two different steps (i) assign each vector to the closest centroid and (ii) update all the centroids. This strategy is repeated until the

assignment becomes stationary. The euclidian distance to the centroids in some M-dimensional is obtained in $\mathcal{O}(M)$. Therefore each iteration of the classical algorithm takes time $\mathcal{O}(K \times M \times N)$. The factor N arises since each vector is tested eventually for some reassignment. This complexity analysis is valid for classical K-means.

Let us now analyze the quantum version of the k-means algorithm. In our strategy, the assignment step in the quantum version is based on the application of several quantum gates. Instead to compute the euclidean distance based on the coordinates list of two different points in $\mathbb{R}^M$, in the quantum version we directly compare the two quantum states. Also finding the problem of the closest centroid is optimized since is based on Grover's algorithm [Gro98].

The cluster assignment in the quantum context is no more a list of the different cluster assignments as in the classical k-means problem. The assignment is a quantum state which contains the different cluster labels correlated with the corresponding cluster assignments by using a quantum superposition.

As we have shown in section 5.6.1, to compute the distance between two vectors of size M, a complexity of $log_2(M)$ is required, and to compute the distance between N observations and K centers of size M a complexity of $Log(N \times K \times N)$ is required. Therefore the unsupervised quantum k-means has a complexity of order $\mathcal{O}(\log(K \times M \times N))$.

5.9 Conclusion

In this chapter, we have implemented a new logarithmic time complexity quantum algorithm for k-means clustering. We have analyzed three different methods to estimate the distance for quantum prototypes based clustering algorithms. Through this analysis, we noticed that the notion of distance in quantum computing is different from the classical one. Because what counts in the quantum computation is the correlation not the real values of the distance. This analysis is so crucial as it can solve any prototype based clustering algorithm. To measure the quality of clustering, we have adapted a classical criterion to the quantum case. This quantum version of k-means has given a good classification just like its classic version, the only difference is its complexity; while the classic version of k-means takes polynomial time, the quantum version only takes logarithmic time, especially in large data sets.

Chapter 6

Clustering using Quantum annealing

Data clustering is an unsupervised task whose objective is to determine a finite set of categories (clusters) to define a partition of a dataset based on the similarities between its objects. There are several algorithms to perform this task, among them we find the well know *K*-means, NMF, Semi-NMF, and Convex-NMF. All these algorithms are based on some optimization problems, where some fixed functional defined by using a fixed metric is minimized. The goal of this strategy is to minimize the distances between data within the same cluster and maximize the distances between clusters.

In machine learning we deal with very high dimensional data. Therefore, solving optimization problems in high dimensions has a huge interest. Optimization in very high dimensions is a true challenge since the near-optimal optimization procedures are very slow. In addition to that, one of the most important challenges of machine learning applications is the complexity of their algorithms. The majority of them have NP-hard or NP-complete complexity, including the algorithms: *K*-means [Alo+09], Neural Network [BR93] and decision tree algorithm [LR76]. Furthermore, with the explosion of data in the world over the past few years, the problem of complexity has become very important, which has motivated researchers to focus more and more on finding the best approach to process these huge datasets as quickly as possible.

One of the proposed solutions for these problems is Quantum Computing. This last one has become a very interesting research area to solve problems of high complexity, following the two main revolutionary algorithms introduced by Grover and Shor. One of the main goals of this new area of research is to solve problems that can't be solved in the classical framework, reducing the computational complexity of many difficult problems and sometimes finding good shortcuts and new approaches to solve them.

This solution has been used by several researchers to find more powerful machine learning algorithms. Most of these quantum algorithms are tested in the D-wave 2000Q quantum computer. Some of these works are: Support Vector Machines (SVM) on the D-Wave quantum annealer [Wil+20] by D. Willsch et all, which introduce a method to train SVM algorithm in D-wave 2000Q quantum annealer. Training and Classification using a Restricted Boltzmann Machine on the D-Wave 2000Q [Dix+20] by V. Dixit et all. Adiabatic Quantum Linear Regression [DP20] by P. Date and T. E Potok. Non-negative/binary matrix factorization with a D-wave quantum annealer [O18] by D. OMalley et all and Low-Rank Non-Negative Matrix Factorization with D-Wave 2000Q [OA18] by O. Daniele and A. Alfonso. The partitioning of the graph into k parts is treated in the D-Wave 2X computer and the quantum annealing by

Ushijima-Mwesigwa et all in [UMNM17]. The paper [NVDS18] uses the D-wave 2000Q quantum processing unit (QPU) and explains how the Quantum-assisted cluster analysis algorithm can be represented by a quadratic unconstrained binary optimization (QUBO) problem. K. Wereszczysk et all propose in [Wer+18] a new method of data clustering by using quantum annealing. In addition, there are several papers that propose QUBOs problems to solve the problems of clustering algorithms such as [Bau+18] for binary clustering, [Bau+19] for K-medoids, [Kum+18] to approximate K-means and [DAPN21] proposes QUBOs problems for the three algorithms: linear regression, support vector machine (SVM) and balanced K-means clustering. The clustering problem is addressed on another quantum computer called IBMQX2 by K. Sumsam Ullah [KAVL19], in this later, the authors allow to implement a similar version of the classical K-means algorithm on the IBMQX2 quantum computer.

In this chapter, we address both algorithms, Balanced K-means and Convex-NMF, based on our contributions in the two papers [Zai+21a] and [Zai+21b]. For the first one, the Balanced K-means clustering algorithm, has already been treated in [Art+20]. This later presents a QUBO formulation equivalent to the Balanced K-means optimization problem. From our side, we will modify this formulation to get better clustering results. We will use D-wave 2000Q quantum computer to test our approach on the three datasets: Iris, Wine, and Breast cancer. Also, we will do a small study about the number of qubits that we can use in D-wave 2000Q in order to justify the usage of the small datasets. To compare the obtained results with our method and the one proposed in the paper [Art+20] and classical Balanced K-means algorithms, we will use the Davies-Bouldin quality index. For the second one, we propose a quantum version of the Convex-NMF algorithm. This means we propose a quantum version of the algorithm to find the global minimum of the functional $\|X - XWG\|_F^2$ for a real valued matrix X. We adopt an alternate optimization strategy for this functional leading to two independent optimization problems for fixed G and W respectively. As already mentioned, our quantum approach is based on the D-Wave quantum annealer which deals with binary optimization problems. We explain here how to construct the two QUBOs problems to find the two matrices W and G. Both of these QUBOs will be executed in D-wave 2000Q to find the two matrices W and G that minimize the Frobenius norm $\|X - XWG\|_F^2$. This is a constrained optimization problem. The constraints are, on the one hand, on the non-negativity of all elements of G and W respectively, and on the other hand, the sum of rows/columns of G/W is always equal to 1.

6.1 Quantum annealing

In order to find the global minimum of an objective function we use the quantum annealing, which was first proposed by B. Apolloni et all in 1988 in the following papers [ADFCB88][ACDF89] and has been subsequently reformulated in [KN98][Fin+94]. In 2012, D-wave announced the first computer for quantum annealing with 128 qubits. This adiabatic quantum computer prepares a Hamiltonian, i.e. prepares a quantum system with several interconnected qubits. These qubits are superposed at the beginning of the processing. The computer will then evolve this Hamiltonian in an adiabatic way to find the solution of the problem. Today, it is possible to go up to 2000 qubits with D-wave 2000Q quantum computer. This computer is deposited by D-wave in open source [Fin17], it contains the necessary tools for quantum annealing and solves the QUBO problem in a hybrid way on quantum processors and classical

hardware architectures using the Qbsolv software. D-Wave quantum annealer manipulates the QUBO problems in a native way [McG14]. It starts with a set of superposed qubits, with each qubit having the same probability of state 0 and state 1. After a few microseconds, we found the classical states in the qubits that represent the minimum energy of the problem, or a state very close to it.

In order to use this quantum computer we just need to transfer the problem to a QUBO problem, and we do the embedding to give the problem as input to D-wave 2000Q, this later search for the global minimum of the QUBO.

The generic QUBO problem has the following form:

$$\sum_{b} \psi(b) q_b + \sum_{b<b'} \psi'(b,b') q_b q_{b'} \tag{6.1}$$

where $\psi(b) \in \mathbb{R}$ are the linear coefficients, $\psi'(b,b') \in \mathbb{R}$ are the quadratic coefficients of the problem and $q_b, q_{b'} \in \mathbb{B}$ for all $i, j \in [0,n]$ where $\mathbb{B} = \{0,1\}$ and $0 \leqslant j \leqslant i \leqslant n$. n is the number of binary variables of the problem.

The problem can be formulated using matrix notation as follows:

$$\min_{q \in \mathbb{B}^n} q^T \Psi q \tag{6.2}$$

where $\Psi \in \mathbb{R}^{n \times n}$ is a symmetric $n \times n$ matrix containing the coefficients $\psi(b)$ and $\psi'(b,b')$ and q it's a binary vector.

6.2 Balanced K-means using Quantum annealing

Balanced K-means algorithm is a special case of K-means algorithm, with K-means we try to divide the dataset into k clusters, and the clusters can be of different size, with Balanced K-means, all the k clusters are of the same size N/k. In this section, we propose a new quantum version of the Balanced K-means algorithm in the D-wave quantum computer, we modify the QUBO formulation of Balanced K-means that has been proposed in the paper [Art+20] and we demonstrate that our setup provides the best results.

6.2.1 QUBO formulation of Balanced K-means

The QUBO of Balanced K-means has already been constructed in the paper [Art+20]. This latter demonstrates how we can write the Balanced K-means problem in the following form :

$$\min_{q \in \mathbb{B}^n} q^T \Psi q \tag{6.3}$$

The authors of this paper have started with the objective function of the classical K-means:

$$\min_{C} \sum_{k=1}^{K} \sum_{x \in C_k} \|x - w_k\|^2 \tag{6.4}$$

where $X = \{x_1, x_2, \dots x_N\}$ is the dataset, $C = \{c_1, c_2, \dots, c_K\}$ is the set of K clusters and each w_k is the centroid of the cluster c_k where $k = 1, \dots, K$.

We can write the problem (6.4) as follows if we use the law of total variance:

$$\min_{C} \sum_{k=1}^{K} \frac{1}{2|c_k|} \sum_{x,x' \in c_k} \|x - x'\|^2 \tag{6.5}$$

In the case of Balanced K-means, each cluster contains the same number of data, so $|c_k|$ is the same for all clusters, so the equation (6.5) can be reformulated as follows :

$$\min_{C} \sum_{k=1}^{K} \sum_{x,x' \in c_k} \|x - x'\|^2 \tag{6.6}$$

To construct the QUBO problem of problem (6.6), the paper [Art+20] defines a distance matrix $D \in \mathbb{R}^{NxN}$ where each d_{ij} denotes the distance between x_i and x_j, also, it defines a binary matrix Φ where $\phi_{ij} = 1$ if only if the element x_i is in the cluster c_j.

Using these two matrices D and Φ, we write the following formula :

$$\sum_{x,x' \in C_k} \|x - x'\|^2 = \phi_j^T D \phi_j \tag{6.7}$$

where ϕ_j is the j^{th} column of the matrix Φ.

Through this equation, the paper starts the construction of the QUBO problem of the Balanced K-means. It starts by converting the Φ matrix into a vector as follows:

$$\phi = [\phi_{11} \dots \phi_{N1} \phi_{12} \dots \phi_{N2} \dots \phi_{1k} \dots \phi_{NK}]^T$$

So, the equivalent of (6.6) if we use the vector ϕ is:

$$\min_{\phi} \phi^T (I_k \otimes D) \phi \tag{6.8}$$

I_k is the identity matrix of dimension K.

In general, when building QUBOs, the constraints are included in the form of penalty terms. The paper [Art+20] includes the following penalty term (6.9) to add the constraint that guarantees that each cluster contains N/K elements at the end of the clustering.

$$\alpha(\phi_j^T \phi_j - \frac{N}{K})^2 \tag{6.9}$$

In order to make sure that each data element belongs to a single cluster, the authors of [Art+20] add the following constraint:

$$\beta(\phi_j^T \phi_j - 1)^2 \tag{6.10}$$

The two constants α and β are used to make the two penalties sufficiently large in order to ensure that the two constraints are always respected.

The paper [Art+20] uses these penalties and proposes the QUBO corresponding to the Balanced K-means (6.6) as follows:

$$\min_{\phi} \phi^T (I_K \otimes (D + \alpha F) + H^T (I_N \otimes \beta G) Q) \phi \tag{6.11}$$

where

$$F = 1_N - \frac{2N}{K} I_N \text{ and } G = 1_K - 2I_K \tag{6.12}$$

1_N is a matrix of ones of the dimension $N \times N$.

The matrix $H \in \mathbb{B}^{NK \times NK}$ defined as follows:

$$h_{ij} = \begin{cases} 1 & \text{if} \quad j = Nmod(i-1,k) + \frac{i-1}{k} + 1 \\ 0 & \text{else} \end{cases} \tag{6.13}$$

The problem (6.11) and the general representation of the QUBO (6.3) are equivalent if we take:

$$q = \phi \text{ and } \Psi = (I_K \otimes (D + \alpha F) + H^T (I_N \otimes \beta G) Q) \tag{6.14}$$

6.2.2 How we can set up the QUBO to find better results?

The two constants α and β are very important to get good results. The paper [Art+20] chooses these two constants as follows:

$$\alpha = \frac{max(D)}{2(N/K) - 1} \text{ and } \beta = max(D) \tag{6.15}$$

with $max(D)$ denote the maximum distance in the matrix D.

The choice of β here is made in a very general manner on the whole dataset. In some cases, it can generate problems to assign elements to clusters. For example, consider the case of two variables ϕ_i and ϕ_j where we try to minimize $\alpha_1\phi_i + \alpha_2\phi_j + \beta(\phi_i\phi_j - 1)^2$, to ensure that this equation evaluates the minimum that is possible and that ϕ_i and ϕ_j aren't both at 0 at the same time, we must choose the right value of β.

If we follow the proposal of β in (6.15), the problem of non-assignment of elements in the clusters can be encountered. In order to solve this problem, we propose to use a vector of weights instead of a general constant on the dataset. Each data element x in the dataset is associated with a weight that is equal to the sum of distances of this element with all elements divided by the sum of distances between all elements in the dataset. So, the constant β for us is chosen as follows:

$$\beta = [\beta_1, \ldots, \beta_N] \tag{6.16}$$

where

$$\beta_i = \frac{\sum_{j=1}^{N} d_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij}} \tag{6.17}$$

On the other hand, for non-assigned elements, the paper [Art+20] proposes to automatically add these elements to the first cluster. In our case, to get the best results, we calculate the centroids of the founded clusters and then we assign each non-assigned element to the closest cluster.

6.2.3 Results and analysis of Quantum Balanced K-means

The optimization problem (6.11) is equivalent to the following minimization problem:

$$\min_{\phi} \sum_{l=1}^{k} \sum_{j=1}^{N} \sum_{i=1}^{N} \sum_{m=1}^{d} \phi_{il}(x_{im} - x_{jm})^2 \phi_{jl} \tag{6.18a}$$

$$+ \alpha \sum_{l=1}^{k} \sum_{j=1}^{N} \sum_{i=1}^{N} \phi_{il} f_{ij} \phi_{jl} + \beta \sum_{l=1}^{N} \sum_{j=1}^{k} \sum_{i=1}^{k} \phi_{li} g_{ij} \phi_{lj} \tag{6.18b}$$

Therefore, in order to test and compare the results of our algorithm with the results found by the approach of the paper [Art+20], we just change the constant β in this last minimization problem (6.18). For the approach of the paper [Art+20] we use the following constants:

$$\alpha = \frac{max(D)}{2(N/k) - 1} \text{ and } \beta = max(D) \tag{6.19}$$

For our approach we use:

$$\alpha = \frac{max(D)}{2(N/k) - 1} \text{ and } \beta = [\beta_1, \ldots, \beta_N] \tag{6.20}$$

where

$$\beta_i = \frac{\sum_{j=1}^{N} d_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij}} \tag{6.21}$$

datasets

D-wave 2000Q handles a collection of qubits and a group of couplers between certain pairs of qubits. The challenge is that there are some qubits in the quantum computer that are not connected between them. The connectivity of those qubits is represented by a chimera graph, each node of this chimera graph is a qubit and each edge is a connexion between two qubits. This chimera graph has a very specific architecture [Bun+14]. An example of a chimera graph is shown in Figure 6.1 with 512 qubits, they are grouped by 8 and every group is connected to adjacent groups by 4 edges. The same architecture is maintained for D-wave 2000Q but with 2048 qubits. The architecture clearly shows that the chimera graph is not completely connected.

On the other hand, in the QUBO problem (6.18), all the coefficients of the quadratic part are not equal to zero, which means that the qubits of the QUBO (6.18) are completely connected between them. Therefore, the chimera graph of our problem is completely connected.

Consequently, running our problem directly in the D-wave 2000Q is not possible. In order to make it possible, we use several qubits to represent each variable of our problem. That means, we need to embed our chimera graph in the chimera graph of D-wave 2000Q.
In Figure 6.2, we illustrate the number of qubits, that we can use in D-wave 2000Q for our problem according to the number of data in the dataset and in the case of three clusters $K = 3$.

According to this constraint, we have chosen the following datasets:

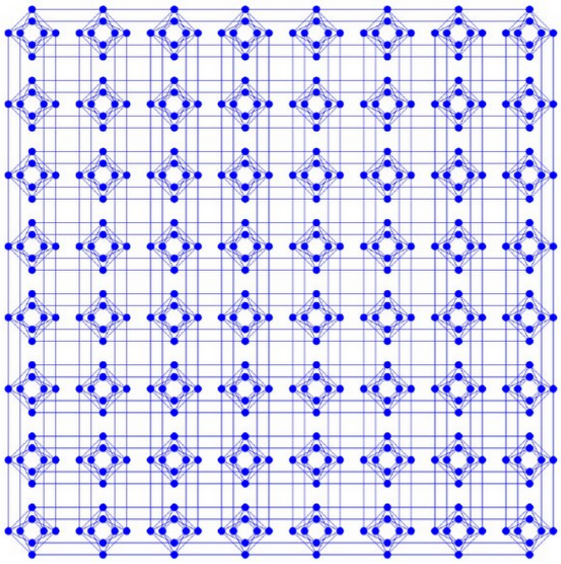


FIGURE 6.1: A chimeric graph that interconnects the qubits of a D-wave system. Every vertex of this graph represents a qubit and the edges between the vertices represent the connection between the qubits.

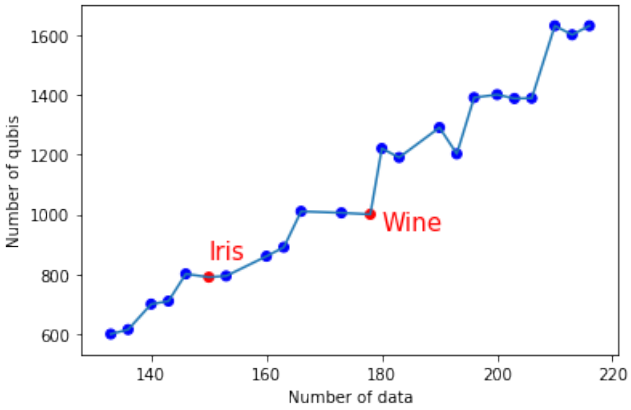


FIGURE 6.2: The number of used qubits in D-wave 2000Q after the embedding of our problem according to the number of data in the dataset processed in the case of 3 clusters.

- **Iris:** is a very well known dataset in the field of machine learning. This dataset contains three classes. Each one has 50 elements. One cluster is linearly separable but the other two are not.
- **Wine:** is a dataset of three classes, where each cluster represents the chemical analysis of a type of wine grown in the same region in Italy. This dataset has 178 instants with dimension 13.
- **Breast Cancer:** is a dataset of two classes and has 569 instants with dimension 32. Here we will use only a subset of this dataset.

Used metric

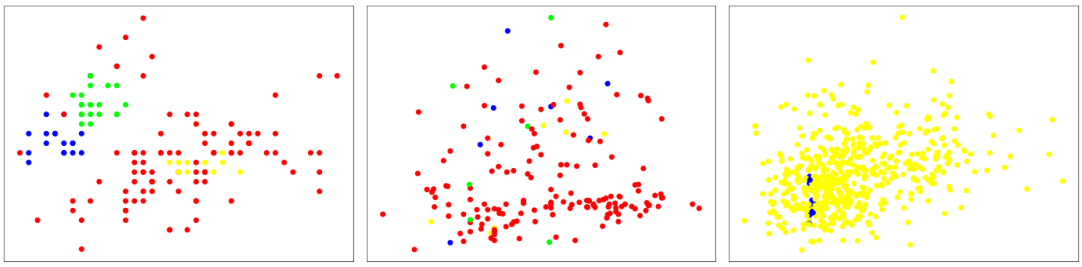
To evaluate the performance of our approach and other approaches, we use the Davies-Bouldin index [DB79]. This is a well-known index in the field of unsupervised machine learning to evaluate the quality of clustering. This index can be formulated as follows:

$$DB = \frac{1}{K}\sum_{k=1}^{K}\max_{k\neq k'}\frac{d_n(w_k)+d_n(w_{k'})}{d(w_k,w_{k'})} \tag{6.22}$$

where K is the number of clusters, d_n is the average distance of all elements from the cluster C_k to their cluster center w_k, $d(w_k, w_{k'})$ is the distance between clusters centers w_k and $w_{k'}$. More the value of DB is lower, more the clustering is better.

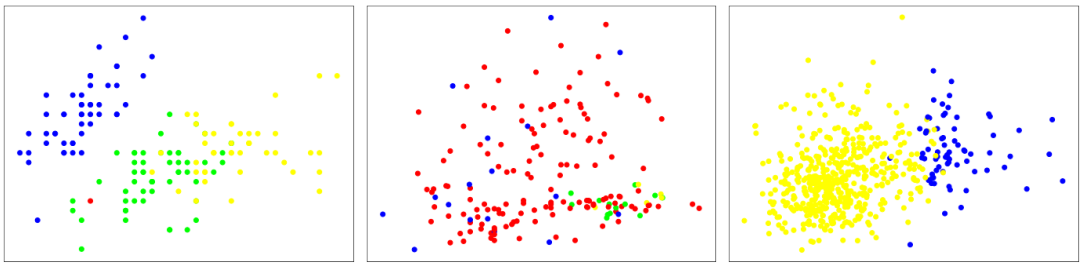
Assignment results

The first comparative study that we can do here is to determine which approach assigns the most data to clusters. After running the problem (6.18) in D-wave 2000Q with the two different β, we present the results obtained in the two Figures 6.3 and 6.4. In both Figures, clusters are represented by the three colors: green, yellow, and blue, and unassigned elements are represented by the red color.



((A)) ((B)) ((C))

FIGURE 6.3: The assignment results of the Iris dataset (A), Wine dataset (B) and Breast Cancer dataset (C) using the approach of [Art+20].



((A)) ((B)) ((C))

FIGURE 6.4: The assignment results of the Iris dataset (A), Wine dataset (B), and Breast Cancer dataset (C) using our approach.

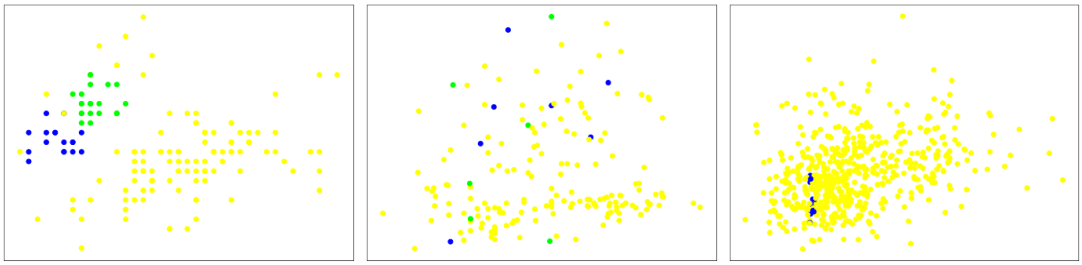
After analyzing the Figures 6.3 and 6.4, we see that our approach allows us to assign all the elements of the Iris dataset except one, and the approach proposed by [Art+20] gives back several elements not assigned. For the Wine dataset, we can see that our approach also gives back more non-assigned elements but not so many as the approach proposed by [Art+20]. For the breast cancer dataset, all data are assigned to

the clusters by the two different approaches.

Based on these results, we can assume that our approach is better in the case of data assignment to clusters.

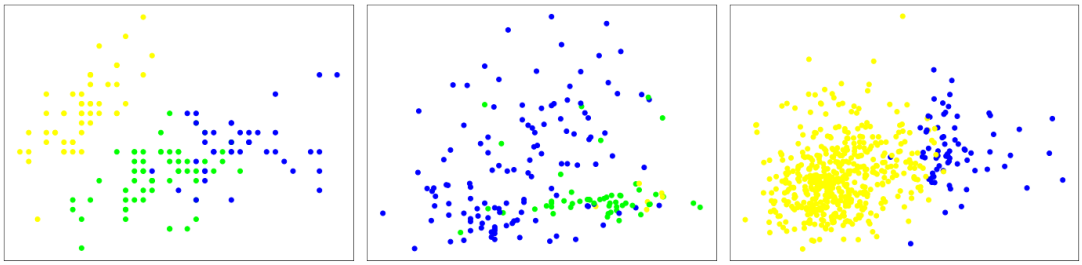
Clustering results

Figures 6.5 and 6.6 represent the clustering results after the execution in D-wave 2000Q quantum computer. Figure 6.5 shows the results of the approach proposed by the paper [Art+20] and Figure 6.6 shows the results of our approach.



((A)) ((B)) ((C))

FIGURE 6.5: The results of the clustering of the Iris dataset (A), Wine dataset (B) and Breast Cancer dataset (C) using the approach of [Art+20].



((A)) ((B)) ((C))

FIGURE 6.6: The results of the clustering of the Iris dataset (A), Wine dataset (B), and Breast Cancer dataset (C) using our approach.

To compare the results found by our approach and the approach of the paper [Art+20] and the classical approach of Balanced K-means we use the Davies-Bouldi index introduced in the paragraph 6.2.3. Table 6.1 shows the obtained results for the three datasets Iris, wine, and Breast cancer using the algorithms: Balanced K-means, classical K-means, Quantum Balanced K-means proposed by [Art+20] and our Balanced K-means.
In Table 6.1 we can confirm that our approach is the most efficient than the others. It has the smallest Davies-Bouldi index among the four treated approaches.

Concerning the execution time of the three approaches, our approach and the approach of [Art+20], no differences are apparent at the execution level in the D-wave 2000Q quantum computer, it's less than $20ms$ for the three datasets. On the other

Dataset	BK-means	K-means	QBk-means	Our Approach
Iris	0.69	0.66	0.73	0.67
Wine	0.525	0.53	0.93	0.44
Breast Cancer	0.75	0.50	1.04	0.48

TABLE 6.1: Davies-Bouldi results for Balanced K-means, K-means, Quantum balanced K-means proposed by [Art+20] and our Quantum balanced K-means

hand, concerning the classical approach, it's $565ms$ for Iris, $501ms$ for Wine, and $7.04s$ for Breast Cancer. This speed of the execution of our approach and [Art+20] approach comes back from the computational power of the quantum computer. Unfortunately, the number of qubits that can be used in D-wave 2000Q is limited, so we cannot test our approach on large datasets to see the real difference between the classical and the quantum approach.

6.3 Convex Non-negative Matrix Factorization Through Quantum Annealing

In this section, we provide the quantum version of the Convex Non-negative Matrix Factorization algorithm (Convex-NMF) by using the D-wave quantum computer. More precisely, we use D-wave 2000Q to find the low-rank approximation of a fixed real-valued matrix X by the product of two non-negative matrices factors W and G such that the Frobenius norm of the difference $X - XWG$ is minimized. In order to solve this optimization problem we proceed in two steps. In the first step, we transform the global real optimization problem depending on W, G into two QUBOs problems depending on W and G respectively. In the second step, we use an alternative strategy between the two QUBOs problems corresponding to W and G to find the global solution. The running of these two QUBOs problems on D-wave 2000Q need to use an embedding to the chimera graph of D-wave 2000Q, this embedding is limited by the number of qubits of D-wave 2000Q. We perform a study on the maximum number of real data to be used by our approach on D-wave 2000Q. The proposed study is based on the number of qubits used to represent each real variable. We test our approach on D-Wave 2000Q with several randomly generated datasets to prove that our approach is faster than the classical approach and also to prove that it gets the best results.

6.3.1 Classical and Convex NMF

We briefly describe below the classical version of non-negative matrix factorization and its convex version. Let $X = (X_1, X_2, \ldots, X_N) \in \mathbb{R}^{M\times N}$, be a data matrix with M rows and N columns, here $X_n \in \mathbb{R}^{M\times 1}$ represents the n^{th} column of X. In what follows $\|\cdot\|$ stays for the Euclidean norm and $\|\cdot\|_F$ for the Frobenius norm. Let K be a fixed input parameter. To refer to the (m, n) element of a matrix X, we either write x_{mn} or X_{mn}. Finally, subscripts or summation indices k will be understood to range from 1 to K, subscripts or summation indices n will range from 1 up to N (the number of data vectors), subscripts or summation indices m will range from 1 up to M (the dimension of data vectors) and subscripts or summation primes indices will be used to expand inner products between vectors or rows and columns of the same matrix.

Classical NMF

In the classical NMF we consider that the data matrix has only non-negative elements. The classical NMF gives a low-rank approximation of X by the product of two non-negative matrices FG where the factors are $F = (F_1, F_2, \ldots, F_K) \in \mathbb{R}_+^{M\times K}$, $G = (G_1, G_2, \ldots, G_K)^T \in \mathbb{R}_+^{K\times N}$ and T denotes the transpose operator. The NMF decomposition could be formulated as a constrained optimization problem by minimizing the following error function:

$$(F, G) = \underset{F,G\geq 0}{\operatorname{argmin}} \|X - FG\|_F^2. \tag{6.23}$$

The non-negativity constraints problem in the matrix form are $F, G \geq 0$.

Convex NMF

In the Convex-NMF we consider that the data matrix X is a real valued matrix. The Convex-NMF problem can be solved if we find the two matrices W and G that minimized the function:

$$(W, G) = \underset{W,G\geq 0}{\operatorname{argmin}} \|X - XWG\|_F^2 \tag{6.24}$$

where $X \in \mathbb{R}^{M\times N}, W \in \mathbb{R}_+^{N\times K}$ and $G \in \mathbb{R}_+^{K\times N}$. In other words, the matrix W represents the positive weights coefficients such that we have:

$$F_k = \sum_{n=1}^{N} w_{nk} X_n = XW_k.$$

All the elements in matrices W and G are non-negatives such that:

$$\sum_{n=1}^{N} w_{nk} = 1 \text{ and } \sum_{n=1}^{N} g_{kn} = 1. \tag{6.25}$$

NMF algorithms

The NMF decomposition has been studied early by Golub and Paatero [PT94],[XHP99]. Several families of algorithms are proposed to solve this matrix approximation problem. The approach proposed by Lee and Seung [LS01] is based on a gradient descent strategy that adaptively defines the gradient rates leading to multiplicative update rules. Another solution is to use an alternate least square strategy. We start with a random initialization of G and after that usually unfolds two phases: at first the functional is minimized with respect to F while in the second phase, the functional is minimized with respect to F, or W in the convex version. These algorithms could converge to a stationary point which is not necessarily global minima. There is no guarantee that we can exactly recover the original matrix from F and G or W and G so we will approximate it as best as possible in terms of the approximation error measured in the Frobenius norm. The work of Lee and Seung [LS01] reveals that NMF has an inherent clustering property, i.e., it automatically clusters the columns of input matrix X since the matrix columns vector factor $(F_1, F_2, \ldots, F_K)$ could be considered as cluster centroids while the the matrix rows vector factor $(G_1, G_2, \ldots, G_K)^T$ could be considered as cluster indicators. This fact brought much attention to NMF in machine learning and data mining communities. The aim of clustering is to cluster

the columns of X, so as to optimize the difference between X and the clustered matrix revealing significant block structure. When G is an orthogonal matrix $G^T G = I$, the resulting non-negative matrix factorization (NMF) is equivalent to relaxed K-means clustering (see [DHS05]). The K-means clustering is one of the most widely used clustering methods early developed by Lloyd [LLo82]. To summarize, the K-means clustering problem can be formulated as a matrix approximation problem [LS01] where the clustering aim is to minimize the approximation error between the original data X and the reconstructed matrix based on the cluster structures.

6.3.2 How we can deal with real values in QUBO problems?

In the QUBO problem, we use only binary variables, so we use the method described by D. Ottaviani and A. Amendola in [OA18] to switch from a real representation to a binary representation. In [OA18] a generic real element x_{mn} is represented as follows:

$$x_{mn} = \alpha \sum_{b=0}^{B} 2^b q_b \tag{6.26}$$

with α a constant and $B+1$ is the number of binary variables used to represent the element x_{mn}.

Therefore, to perform a general binary representation of a row of a matrix, we take the matrix $X \in \mathbb{R}^{M\times N}$ where each $X_m \in \mathbb{R}^{1\times N}$ is the m-th row of this matrix. In order to represent this vector with binary variables we represent each item of this vector by $B+1$ binary variables. To do that, we use the following set:

$$Q_n \equiv \{n(B+1), n(B+1)+1, \ldots, n(B+1)+B\} \tag{6.27}$$

where $n \in \{0, 1, \ldots, N\}$. If we take $n = 0$, then the first item of the vector X_m, is represented by $Q_0 \equiv \{0, 1, \ldots, B\}$ and if we take $n = 1$, then the second item of the vector X_m, is represented by $Q_1 \equiv \{(B+1), (B+2), \ldots, 2B+1\}$ and so on. So, to represent a row X_m we reformulate the general transformation as follows:

$$X_m = \sum_{n=0}^{N} \sum_{b=0}^{B} \beta_{nb}^m q_b \tag{6.28}$$

where

$$\beta_{nb}^m = \begin{cases} \alpha \cdot 2^{b-n(B+1)} & \text{if} \quad b \in Q_n, \\ 0 & \text{if} \quad b \notin Q_n. \end{cases} \tag{6.29}$$

Therefore, we use $(B+1)\times M \times N$ binary variables to represent a matrix $X \in \mathbb{R}^{M\times N}$. In the case where $B = 9$, $\alpha = 0.001$, we can represent every real value $x_{mn} \in [0, 1.023]$.

6.3.3 Modeling the Convex-NMF problem by QUBOs problems

In this subsection we present our strategy to find the two matrices W and G. We will use the *alternate optimization* idea to decompose the initial problem (6.24) into two different optimization independent problems. This strategy gives us the possibility to define two QUBOs problems which will be solved separately. Then we show the procedure to get the linear and quadratic coefficients for our problems, at the end we discuss the problem of embedding to the chimera graph of D-wave 2000Q and the maximum size of data that we can use.

Convex-NMF decomposition

Firstly, we fix the matrix W and we solve with respect to G the following minimization problem:

$$\min_{G\geq 0} \|X - XWG\|_F^2 \tag{6.30}$$

secondly, we fix the matrix G and we solve with respect to W the following minimization problem:

$$\min_{W\geq 0} \|X - XWG\|_F^2. \tag{6.31}$$

In order to satisfy the conditions described in (6.25), we add two constraints on the two problems described by (6.30) and (6.31) respectively. We rewrite these problems as follows:

$$\min_{G} \left(\|X - XWG\|_F^2 + \sum_{k=1}^{K} \left[1 - \sum_{n=1}^{N} g_{kn} \right]^2 \right) \tag{6.32}$$

and

$$\min_{W} \left(\|X - XWG\|_F^2 + \sum_{k=1}^{K} \left[1 - \sum_{n=1}^{N} w_{nk} \right]^2 \right) \tag{6.33}$$

Regarding the squared Frobenius norm of a matrix, we recall the following properties:

$$\|X\|_F^2 = \sum_{m=1}^{M}\sum_{n=1}^{N} x_{mn}^2 = \sum_{n=1}^{N} \|x_n\|^2 \tag{6.34a}$$

$$= \sum_{n=1}^{N} x_n^T x_n = \sum_{n=1}^{N} (X^T X)_{nn} = Tr(X^T X) \tag{6.34b}$$

By definition, note that the functional $\|X - XWG\|_F^2$ takes this form:

$$\|X - XWG\|_F^2 = Tr(X^T X - 2GX^T XW + W^T X^T XWGG^T) \tag{6.35a}$$

$$= Tr(X^T X) - 2Tr(X^T XWG) \tag{6.35b}$$

$$+ Tr(W^T X^T XWGG^T) \tag{6.35c}$$

In (6.35) the first term is constant with respect to G and W.

By direct computations of the second term, we get:

$$Tr(X^T XWG) = \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{n'=1}^{N} x_{mn} x_{mn'} w_{n'k} g_{kn} \tag{6.36}$$

The second term is a first order term with respect to G or W.

For the third term $Tr(W^T X^T XWGG^T)$, we obtain:

$$\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{n'=1}^{N}\sum_{k'=1}^{K}\sum_{n''=1}^{N} w_{nk} x_{mn} x_{mn'} w_{n'k'} g_{k'n''} g_{kn''} \tag{6.37}$$

Minimization problem with respect to G

In this case, we fix the matrix W and we are interested to rewrite our functional with respect to the matrix G as a variable. To this end, the equation (6.36) writes:

$$Tr(X^T XWG) = \sum_{n=1}^{N}\sum_{k=1}^{K}(X^T XW)_{nk}g_{kn} \tag{6.38}$$

while the equation (6.37) writes:

$$Tr(W^T X^T XWGG^T) = \sum_{k=1}^{K}\sum_{k'=1}^{K}\sum_{n=1}^{N}(W^T X^T XW)_{kk'}g_{k'n}g_{kn} \tag{6.39}$$

It follows that:

$$\|X - XWG\|_F^2 = Tr(X^T X) - 2\sum_{n=1}^{N}\sum_{k=1}^{K}A_{nk}g_{kn} \tag{6.40a}$$

$$+ \sum_{k=1}^{K}\sum_{k'=1}^{K}\sum_{n=1}^{N}B_{kk'}g_{k'n}g_{kn}. \tag{6.40b}$$

where $A = X^T XW$ and $B = W^T X^T XW$.

The constraint $\sum_{k=1}^{K}\left(1 - \sum_{n=1}^{N} g_{kn}\right)^2$ in (6.32) can be written as (6.41).

$$\sum_{k=1}^{K}\left(1 - 2\sum_{n=1}^{N} g_{kn} + \sum_{n=1}^{N}\sum_{\substack{n'=1\\ n'\neq n}}^{N} g_{kn}g_{kn'} + \sum_{n=1}^{N} g_{kn}^2\right) \tag{6.41}$$

Minimization problem with respect to W

In the case of the problem described by (6.33) we fix the matrix G and we are interested to rewrite our functional with the matrix W as a variable. To this end the (6.36) writes:

$$Tr(X^T XWG) = Tr(GX^T XW) \tag{6.42a}$$

$$= \sum_{n=1}^{N}\sum_{k=1}^{K}(GX^T X)_{kn}w_{nk} \tag{6.42b}$$

$$Tr(W^T X^T XWGG^T) = \sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{n'=1}^{N}\sum_{k'=1}^{K}D_{nn'}E_{k'k}w_{nk}w_{n'k'} \tag{6.43a}$$

$$= \sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{\substack{n'=1\\ n'\neq n}}^{N}\sum_{\substack{k'=1\\ k'\neq k}}^{K}D_{nn'}E_{k'k}w_{nk}w_{n'k'} \tag{6.43b}$$

$$+ \sum_{k=1}^{K}\sum_{n=1}^{N}D_{nn}E_{kk}w_{nk}^2 \tag{6.43c}$$

It follows that:

$$\|X - XWG\|_F^2 = Tr(X^T X) - 2\sum_{n=1}^{N}\sum_{k=1}^{K} C_{kn} w_{nk} \tag{6.44a}$$

$$+ \sum_{k=1}^{K}\sum_{n=1}^{N} D_{nn} E_{kk} w_{nk}^2 \tag{6.44b}$$

$$+ \sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{\substack{n'=1 \\ n'\neq n}}^{N}\sum_{\substack{k'=1 \\ k'\neq k}}^{K} D_{nn'} E_{k'k} w_{nk} w_{n'k'} \tag{6.44c}$$

where $C = GX^T X$, $D = X^T X$ and $E = GG^T$.

The constraint $\sum_{k=1}^{K}\left(1 - \sum_{n=1}^{N} w_{nk}\right)^2$ in (6.33) can be written as (6.45).

$$\sum_{k=1}^{K}\left(1 - 2\sum_{n=1}^{N} w_{nk} + \sum_{n=1}^{N}\sum_{\substack{n'=1 \\ n'\neq n}}^{N} w_{nk} w_{n'k} + \sum_{n=1}^{N} w_{nk}^2\right) \tag{6.45}$$

Construction of the QUBOs of our problems

In what follows we describe how to transform the two problems described in (6.32) and (6.33) to two independent QUBOs problems with respect to G and W. In both equations (6.32) and (6.33), W and G are matrices with real values. Therefore we use the method introduced in 6.3.2 to present these values with binary variables. After that, it is enough to find the two coefficients of each QUBO (linear coefficients $\psi(b)$ and quadratic coefficients $\psi'(b, b')$).

Binary minimization problem with respect to G

For the problem (6.32) we define the linear coefficients $\psi(b)$ and the quadratic coefficients $\psi'(b, b')$ as follows:

$$\psi(b) = \sum_{n=1}^{N}\sum_{k=1}^{K}(-2(A_{nk}+1)\beta_{nb}^k + (B_{kk}+1)(\beta_{nb}^k)^2) \tag{6.46a}$$

$$\psi'(b, b') = 2\sum_{k=1}^{K}\sum_{\substack{k'=1 \\ k'\neq k}}^{K}\sum_{n=1}^{N} B_{kk'}\beta_{nb}^{k'}\beta_{nb'}^k \tag{6.46b}$$

$$+ 2\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{\substack{n'=1 \\ n'\neq n}}^{N} \beta_{nb}^k\beta_{n'b'}^k \tag{6.46c}$$

$$+ 2\sum_{k=1}^{K}\sum_{n=1}^{N}(B_{kk}+1)\beta_{nb}^k\beta_{nb'}^k \tag{6.46d}$$

These formulations follow directly from (6.40) and (6.41).

Binary minimization problem with respect to W

For the problem (6.33) we define the linear coefficients $\psi(b)$ and the quadratic coefficients $\psi'(b,b')$ as follows:

$$\psi(b) = \sum_{n=1}^{N}\sum_{k=1}^{K} -2(C_{kn}+1)\beta_{kb}^{n} \tag{6.47a}$$

$$+ \sum_{k=1}^{K}\sum_{n=1}^{N}(D_{nn}E_{kk}+1)(\beta_{kb}^{n})^2 \tag{6.47b}$$

$$\psi'(b,b') = \sum_{k=1}^{K}\sum_{n=1}^{N} 2(D_{nn}E_{kk}+1)\beta_{kb}^{n}\beta_{kb'}^{n} \tag{6.47c}$$

$$+ 2\sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{\substack{n'=1\\ n'\neq n}}^{N}\Big(\sum_{\substack{k'=1\\ k'\neq k}}^{K} D_{nn'}E_{k'k}\beta_{kb}^{n}\beta_{k'b'}^{n'} + \beta_{kb}^{n}\beta_{kb'}^{n}\Big) \tag{6.47d}$$

These formulations follow directly from (6.44) and (6.45).

Embedding in D-WAVE 2000Q

After having built the two QUBOs of our problems, we notice that the coefficient for each pair of qubits b and b' is not zero, $\psi'(b,b') \neq 0$. This remark allows us to deduce that the qubits of our problem are completely connected between them. It means that the chimera graph which represents the connection of the qubits of our problem is completely connected. On the other hand, D-wave 2000Q processes a set of qubits and a set of couplers between some pairs of qubits. The problem here is that there exist qubits in the quantum processor of this computer that are not connected between them. So, the execution of our problem directly in D-wave 2000Q is not possible. To make it feasible we use multiple qubits to represent each variable b of our problems. That means we need to embed our chimera graph into the D-wave 2000Q chimera graph. On the other hand, to calculate the maximum number of real values that can be used in our problems, we consider these two conditions:

- According to 6.3.2, we use 10 qubits to represent all the real values $x_{mn} \in [0, 1.023]$.
- The chimeras graphs of our problems are completely connected.

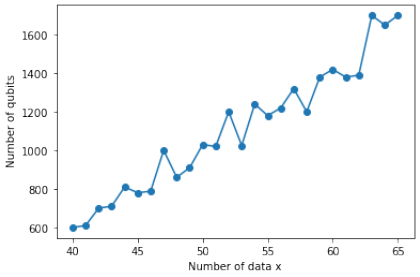


FIGURE 6.7: The number of qubits used in D-wave 2000Q after the embedding of a problem completely connected, in the case where each real number is represented by 10 qubits ($B = 9$ and $\psi'(b,b')! = 0$ for each $b \neq b'$).

The Figure 6.7 shows the maximum number of real values that we can process using our approach in D-wave 2000Q. Unfortunately, until now, we can't test our approach

on a large dataset. The matrices that we can factorize using our approach in the D-wave 2000Q quantum computer must not exceed 65 real values. In other words, if we have a dataset $X \in \mathbb{R}^{M \times N}$, it is necessary that $M \times N \leq 65$.

6.3.4 Results and test on D-wave's quantum computer

In order to test our approach we generate randomly a dataset $X \in \mathbb{R}^{M \times N}$ with $M = 20$ and $N = 3$. In this case we have $N \times M = 60 \leq 65$. Therefore we can use our approach in D-wave 2000Q to approximate the two matrices G and W. Firstly, we build the QUBO of the problem (6.32) and we execute this QUBO in D-Wave 2000Q to find the matrix G. Secondly, we use this matrix G to build the QUBO of the problem (6.33) and we execute this QUBO in D-wave 2000Q to find the matrix W. After finding the two matrices, we represent the two best centers by the red color in Figure 6.8 as follows:

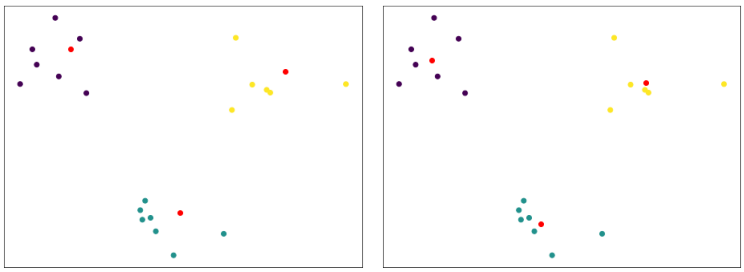


FIGURE 6.8: Results of the test in D-wave 2000Q, each figure represents one of the best results returned by D-wave 2000Q. The red color represents the centroids and each cluster is represented by a color.

In Figure 6.8, we can clearly see that our approach is able to find the right centroids of clusters. These results are very interesting, even if it is on a small dataset because it proves that our approach works well and that the D-wave quantum computer can find the right results that minimize the two optimization problems (6.30) and (6.31) with only one iteration for each problem.

The major limitation of our approach is the number of data that we can handle for each of the two problems defined in the Equations (6.30) and (6.31) respectively. Indeed the actual quantum computers cannot exceed 65 real values in the data matrix. This limitation is related to the number of qubits that we can manipulate in the D-wave quantum computer. However, our approach will be a very pertinent solution to solve the problem of computation time if we have quantum computers with a large number of qubits. This limitation will be a distant memory after a period of time. Because with the results of the paper [Vah+21], we can move to silicon quantum processors with millions of qubits instead of the current devices with a few qubits.

Run time analysis

In order to analyze and compare the computing time of our approach with the classical approach, we made several executions on several small randomly generated datasets $(X_1, X_2, \ldots, X_{12})$, each data of these datasets $x \in X_i$ is a vector of $\mathbb{R}^2$. The maximum number of real values that we can use in our approach is 65, so we have generated 12 random datasets with different sizes and containing at most 64 reals values. We have run our approach on these datasets and also the classical Convex-NMF approach to compare the computing time of each approach. The results of this

analysis are displayed in Figure 6.9 with the red color for classical Convex-NMF and green for Quantum Convex-NMF and the blue color presents the time to find the matrix G and orange to find W in D-wave 2000Q.

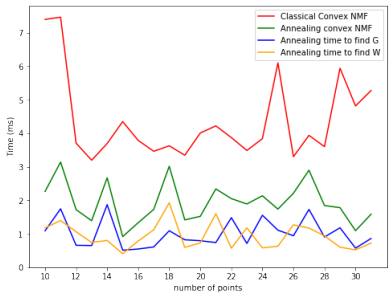


FIGURE 6.9: Total computing time of Classical Convex-NMF (red color) and Quantum Convex-NMF (green color) as the number of points of the dataset. The blue color is the total computing time to find G and the orange to find W.

The green curve in the Figure is simply the sum of the two curves orange and blue. This is intrinsically due to our alternative strategy, in order to find the results using our approach, we solve the first QUBO to find G and then the last QUBO to find W. Therefore, the total running time on D-wave 2000Q is simply calculated as the sum of the running times of these two problems. By comparing the total time spent to solve the problem with our approach (the green curve) and the time taken by the classical convex-NMF approach (the red curve), we can see clearly that our approach is very fast than the classical approach.

6.4 Conclusion

In this chapter, we have discussed the quantum version of the two algorithms Balanced K-means and Convex-NMF. We have modified the Quadratic Unconstrained Binary Optimization (QUBO) of the Balanced K-means algorithm proposed by Arthur and Davis in the paper [Art+20], we have proposed a new constants that handle the belonging of an element of data to two clusters. The constants proposed by the paper [Art+20] are very generic, which leads the approach to give many non-assigned elements. In our approach, we have proposed a vector of constants where each element of the dataset is associated with an element of this vector. Moreover, we have done a comparative analysis between the two approaches to prove that our approach is able to assign the largest number of data to clusters, and we have shown that our approach gives the best clustering by comparing the Davies-Bouldi index. We have proposed a new approach for the Convex-NMF algorithm in D-wave 2000Q. In this approach, we have proposed to decompose the Convex-NMF optimization problem on two QUBOs problems: the first to find the matrix G and the second to find the matrix W which minimizes the norm difference between the data matrix X and the matrix product XWG where all elements of G and W are non-negative and on the rows/columns they sum up to 1. To work with real values we used a transformation proposed in [OA18], this transformation allows us to find a binary representation of the problem from a problem with real variables. Before testing our approach on D-wave 2000Q, we made a study to find the maximum number of real data to use in our problems, this study is based on the number of qubits of D-wave 2000Q, the connection of these qubits (chimera graph architecture) and according to the number

of qubits used to represent each real variable in the QUBO. Our approach is tested on a small dataset generated randomly, to demonstrate that our approach works well and that the D-wave quantum computer is able to find the right clusters of the dataset. Also, we have made several tests to demonstrate that our quantum approach is faster than the classical method. Until today, we cannot test our approach on large datasets, because the number of qubits in D-wave quantum computers is limited. We hope that in the coming years the number of qubits in quantum computers will increase.

Chapter 7

Application case: Fuel Pool Cooling System

As an application, we will deal with the case of a small system of three trains, of 8 components, 5 pumps, 2 CCWS, and one CHRS. The diagram of reliability of this small system is presented in Figure 7.1, and the descriptions of its components are presented in Table 7.1 where the probabilities are randomly generated . The constraints of breakdowns of the components are the following: (i) components of train 2 can't break down except if train 1 is already broken down, (ii) components of train 3 can't break down except if the two first trains are already broken down.

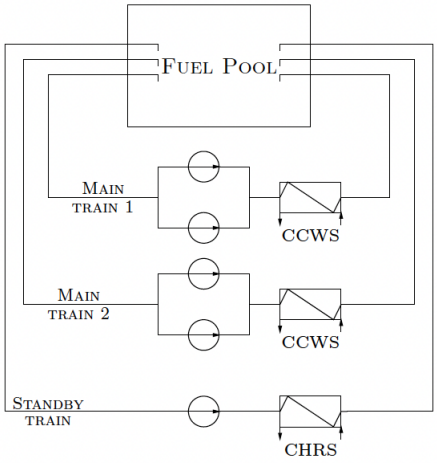


FIGURE 7.1: FPCS system reliability diagram

Index	Component	Probability of failure	Probability of being repaired
0	Fuel Pool In	0.12	0
1	Pomp 1	0.12	0
2	Pomp 2	0.312	0
3	Pomp 3	0.192	0
4	Pomp 4	0.122	0
5	Pomp 5	0.12	0
6	CCWS 1	0.12	0.12
7	CCWS 2	0.2	0
8	CHRS	0.13	0
9	Fuel Pool Out	0.21	0

TABLE 7.1: Description of the FPCS components

7.1 Research of combinations of basic events that can generate serious accidents

In order to find the events that can generate unacceptable consequences if they break down simultaneously, we need to find all the combinations of the basic events that can stop the flow between the Fuel Pool In (FPI) and the Fuel Pool Out (FPO) in the system. To do this, we represent the reliability diagram by an oriented graph using the paper [Hib13], and we use our quantum approach that we have done in the chapter **[ch:QVSP]** to find all the minimal cuts of this oriented graph. Using the reliability diagram (Figure 7.1), we find the graph in Figure 7.2, each vertex of this graph represents a system component, and each arc represents the flow between two components.

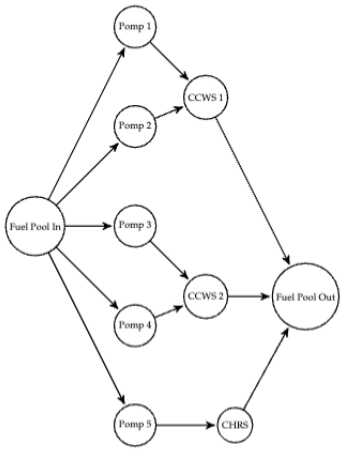


FIGURE 7.2: Representation of the system by a directed graph

In order to simplify the notations, we code each component of the system by an identifier v_i, and the source FPI by the vertex s and FPO by the vertex t, in the Table 7.2 we show each component with its identifier v_i.

Com	Pomp 3	Pomp 2	Pomp 1	Pomp 4	Pomp 5	CCWS 1	CCWS 2	CHRS
Id	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8

TABLE 7.2: Components identification

After this encoding, we show the graph in Figure 7.3. Finding the combinations of basic events that can cause a loss of flow between the FPI and FPO of the system is as simple as finding the subset of vertices that can break the flow between the vertex s and the terminal t of the graph 7.3.

In order to find the subset of vertices that can stop the flow between the source s and the terminal t, we use our approach proposed in chapter 3. We start by creating 12 qubits:

$$|\psi_0\rangle = |c_0, c_1\rangle \otimes |s, v_1, \ldots, v_8, t\rangle \tag{7.1}$$
$$= |0,0\rangle \otimes |0,0,\ldots,0,0\rangle \tag{7.2}$$
$$= |0\rangle^{\otimes 12} \tag{7.3}$$

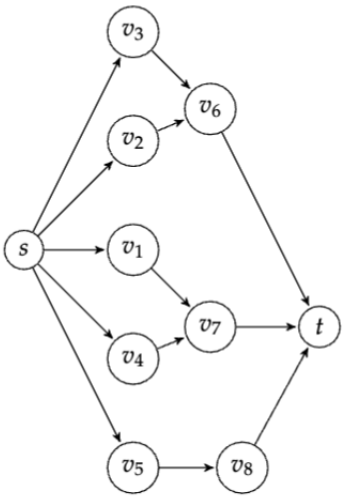


FIGURE 7.3: Directed graph of the system

the two first qubits $|c_0, c_1\rangle$ for the control, the qubit $|s\rangle$ represents the source, the qubit $|t\rangle$ represents the terminal, and each qubit $|v_i\rangle$ represents a vertex v_i.

We apply the X gate in the qubit $|s\rangle$ to represent the source in the following quantum state:

$$|\psi_1\rangle = \mathbb{I}^2 \otimes X \otimes \mathbb{I}^9 |\psi_0\rangle \tag{7.4}$$
$$= |c_0, c_1\rangle \otimes X|s\rangle \otimes |v_1, \ldots, v_8, t\rangle \tag{7.5}$$
$$= |0,0\rangle \otimes |1,0,\ldots,0,0\rangle \tag{7.6}$$

We apply the oracle O_s (see Figure 7.4) to make the movement to the successors of s, which gives the first cut:

$$|\psi_2\rangle = O_s |\psi_1\rangle \tag{7.7}$$
$$= O_s |c_1, c_2\rangle \otimes |1000000000\rangle \tag{7.8}$$
$$= |c_1, c_2\rangle \otimes |0111110000\rangle \tag{7.9}$$

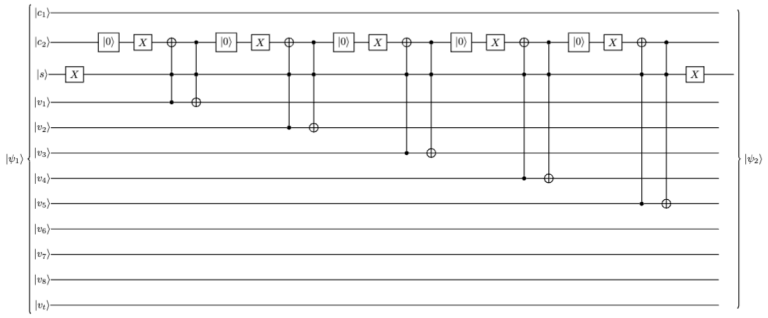


FIGURE 7.4: The first oracle O_s of the movement of s to its successors

This new state $|\psi_2\rangle$ is equivalent to $|v_i\rangle = |1\rangle$ for $i = 1,2,3,4,5$, $|v_i\rangle = |0\rangle$ for $i = 6,7,8$

and $|s\rangle = |t\rangle = |0\rangle$. If we note the set of cuts of the graph by C_s, in this step, with the state $|\psi_2\rangle$, we have a first cut in the set of cuts $C_s = \{\{v_1, v_2, v_3, v_4, v_5\}\}$.

The second step consists in applying a second oracle which allows to establish the movement of the vertex v_1 towards its successors starting with the state $|\psi_2\rangle$. So the oracle O_{v_1} (see Figure 7.5) is applied:

$$|\psi_3\rangle = O_{v_1} |\psi_2\rangle \tag{7.10}$$
$$= O_{v_1} |c_1, c_2\rangle \otimes |0111110000\rangle \tag{7.11}$$
$$= |c_1, c_2\rangle \otimes |0111110000\rangle + |c_1, c_2\rangle \otimes |0011110100\rangle \tag{7.12}$$

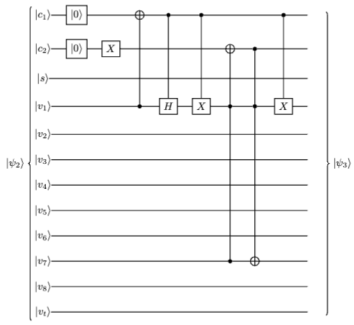


FIGURE 7.5: The Oracle O_{v_1} for the movement of v_1

After this oracle a new state is added to the superposition, which means that a new cut is added to the set of cuts, this state is $|s, v_0, \ldots, v_8, t\rangle = |0011110100\rangle$ which translates into the new cut $\{v_2, v_3, v_4, v_5, v_7\}$. Then, the set of found cuts becomes: $C_s = \{\{v_1, v_2, v_3, v_4, v_5\}, \{v_2, v_3, v_4, v_5, v_7\}\}$. We apply the remaining oracles on $|\psi_3\rangle$, all the remaining oracles are represented in the circuit of Figure 7.6.

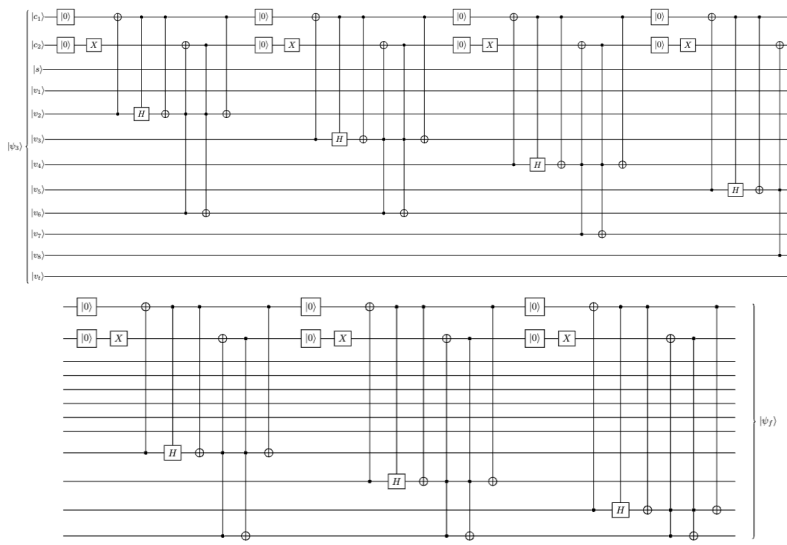


FIGURE 7.6: Oracles $O_{v_8} O_{v_7} O_{v_6} O_{v_5} O_{v_4} O_{v_3} O_{v_2}$

The final superposition $|\psi_f\rangle$ is obtained as follows:

$$
\begin{aligned}
|\psi_f\rangle &= O_{v_8}O_{v_7}O_{v_6}O_{v_5}O_{v_4}O_{v_3}O_{v_2}\,|\psi_3\rangle && (7.13)\\
&= O_{v_8}O_{v_7}O_{v_6}O_{v_5}O_{v_4}O_{v_3}O_{v_2}\,|c_1,c_2\rangle \otimes (|0111110000\rangle + |0011110100\rangle) && (7.14)\\
&= |c_1,c_2\rangle && (7.15)\\
&\otimes (|0111110000\rangle + |0011110100\rangle + |0101111000\rangle + |0001111100\rangle && (7.16)\\
&+ |0110111000\rangle + |0010111100\rangle + |0100111000\rangle + |0000111100\rangle && (7.17)\\
&+ |0111010100\rangle + |0011010100\rangle + |0101011100\rangle + |0001011100\rangle && (7.18)\\
&+ |0110011100\rangle + |0010011100\rangle + |0100011100\rangle + |0000011100\rangle && (7.19)\\
&+ |0111100010\rangle + |0011100110\rangle + |0101101010\rangle + |0001101110\rangle && (7.20)\\
&+ |0110101010\rangle + |0010101110\rangle + |0100101010\rangle + |0000101110\rangle && (7.21)\\
&+ |0111000110\rangle + |0011000110\rangle + |0101001110\rangle + |0001001110\rangle && (7.22)\\
&+ |0110001110\rangle + |0010001110\rangle + |0100001110\rangle + |0000001110\rangle) && (7.23)\\
& && (7.24)
\end{aligned}
$$

By filtering the non-minimal cuts, we found the following minimal cuts: $C_s = \{\{v_1, v_2, v_3, v_4, v_5\}, \{v_6, v_1, v_4, v_5\}, \{v_2, v_3, v_7, v_5\}, \{v_1, v_2, v_3, v_4, v_8\}, \{v_6, v_1, v_4, v_8\}, \{v_2, v_3, v_7, v_8\}, \{v_6, v_7, v_5\}, \{v_6, v_7, v_8\}\}$. According to these results, we replace each identifier with its corresponding name, which is the same as the one shown in Table 7.2. The final results obtained are:

- CCWS 2, CCWS 1, and Pomp 5
- CHRS, CCWS 2 and CCWS 1
- CHRS, CCWS 2, Pomp 2, and Pomp 1
- CCWS 2, Pomp 5, Pomp 2 and Pomp 1
- CHRS, CCWS 1, Pomp 4, and Pomp 3
- CCWS 1, Pomp 5, Pomp 4 and Pomp 3
- Pomp 5, Pomp 4, Pomp 3, Pomp 2 and Pomp 1
- CHRS, Pomp 4, Pomp 3, Pomp 2 and Pomp 1

7.2 Research of failure scenarios

In order to search the failure scenarios of the FPCS system, we use the above-mentioned failure constraints to build the graph of states of the whole system (the components of train 2 do not break down unless train 1 is down, and the components of train 3 do not break down unless both train 1 and 2 are down). This graph of states has 144 vertices (144 possible states of the system) and 434 edges between these vertices (434 possible actions in the system). Each vertex of this graph represents a state of the system, each edge represents an action in the system (a failure or a repair of a component of the system), and the states that correspond to the minimal cuts are indicated in the graph by marked vertices. We take the current state, as the starting state, where all the elements of the system are in good condition and working well, and we look for all the paths from this initial state to the marked states.

In order to perform this research, we use the approach that we have proposed in **??**, we compare the obtained results with the results from the classical random walk approach and we show them in Figure 7.7.

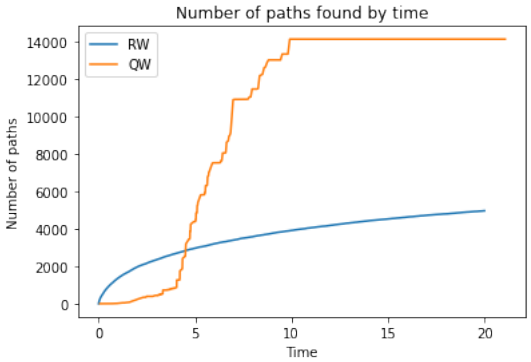


FIGURE 7.7: Number of found scenarios as a function of time

In these results, we found with our approach in total 14135 scenarios and 4964 with the classical random walk approach. In Figure 7.7, we can see clearly that with our approach we have found this number of scenarios only with 10s, on the other hand, with the classical approach we have found less even if with 20s, and after 10s the number of paths found with our approach is still stable which shows the convergence of our approach to a stable set of scenarios and that we have found all possible scenarios.

7.3 How we can find the most similar scenarios?

In the previous section, we have found the scenarios that can be the cause of the minimal cuts found in section 7.1. We have found 14135 scenarios. In this section, we will use the two algorithms 8, 9 that we have proposed in 4.2.3, 4.2.2 and the k-NN algorithm, to classify these scenarios. At present, it's not possible to use large quantum computers, in this work, we use IBM simulator with 32 qubits, which doesn't allow us to run tests on the whole dataset (14135 scenarios), so, in order to test the DTW approaches that we have proposed in the previous chapter, we split these scenarios into several datasets and use each one of them for the test. For each dataset, we take a part for learning and the other for testing. These partitions are made by a random choice:

Dataset	SC_1	SC_2	SC_3	SC_4	SC_5	SC_6
Length of each scenario	8	7	9	8	9	9
Length of training dataset	44	44	44	64	16	16
Length of test dataset	24	24	16	16	12	9
Number of classes	3	3	3	3	3	3

TABLE 7.3: 6 datasets of scenarios

The results that we found are shown in Table 7.4.
In this table, we can see that our approach has the highest metric in all datasets, which demonstrates that using our approach to calculate the distance between scenarios in our particular problem is the best choice.

Method	SC_1	SC_2	SC_3	SC_4	SC_5	SC_6
Classical DTW	0.25	0.25	0.25	0.81	0.25	0.58
Quantum DTW	0.25	0.25	0.25	0.81	0.25	0.58
Faster DTW-2	0.67	0.44	0.88	0.50	0.44	0.50
Faster DTW-3	0.46	0.25	0.38	0.31	0.19	0.42
Faster DTW-4	0.29	0.25	0.19	0.25	0.19	0.33
Faster DTW-5	0.25	0.25	0.19	0.25	0.25	0.33
QDTW-w	0.75	0.67	0.88	0.81	0.75	0.67

TABLE 7.4: Results of k-NN algorithm using QDTW

7.4 FPCS system scenario generator

In the section 7.2, we searched all the scenarios from the initial state of the FPCS system, this initial state is represented by the empty set, i.e. all the elements of the system are working correctly. If this initial state changes, we must search again for all scenarios from the new state. This search process is very long for large systems. In the industry, it is necessary to take a decision as fast as possible to avoid the deterioration, we will wait for the end of the research of all the possible scenarios. So, how can get a general idea about the effects of the change of state? To answer this question, we have proposed in 4, to learn Quantum Hidden Markov Models (QHMMs) able to detect a probable or not probable scenario for a given objective and to generate failure scenarios from any initial state of the system.

Consider the critical states of the system: $C_1, C_2, \ldots, C_8$. Assume that the datasets $X_1, X_2, \ldots, X_8$, have all possible failure scenarios that can make the system go to the critical states $C_1, C_2, \ldots, C_8$ respectively. In order to detect whether a scenario can generate a specific severe accident based on the system history, we learn QHMMs, for each dataset X_i we learn a model M_i with $i = 1, \ldots, 8$.

After that, we use these models and at each change of state of the system, we calculate the probability by each model, and with the model that gives the highest probability, we can detect the most possible critical state from this transition.

We learn these 8 models using datasets X_i and we process the sequence $[0, 7, 102, 2, 24, 114, 90, 55]$ step by step, i.e. we start with the state where all is working well and follow the system at each switching of states. In the normal case of the system, all the elements are in the operating state, so the state of the system is represented by 0. Let's assume that the system switches to state 7. Then, we have a sequence represented by $[0, 7]$. We use the 8 models, calculate the probability of this sequence $[0, 7]$ and display the results in Figure 7.8.

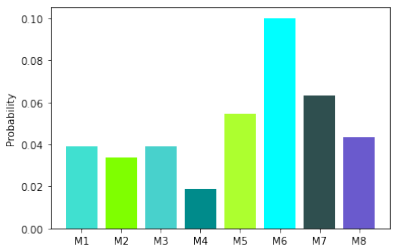


FIGURE 7.8: The probability of the sequence $[0, 7]$ using the 8 models

In Figure 7.8, we can see that all models give a non-null probability, which indicates that this switch of state can lead to the 8 various critical states. Also, we can see that the model 6, gives the highest probability, which signifies that the most probable critical state following this switch of the state is the one that is represented by the model 6. We take the processed sequence and process each change of state, the sequences become as follows:

- **A**: $[0, 7, 102]$
- **B**: $[0, 7, 102, 2]$
- **C**: $[0, 7, 102, 2, 24]$
- **D**: $[0, 7, 102, 2, 24, 114]$
- **E**: $[0, 7, 102, 2, 24, 114, 90]$
- **F**: $[0, 7, 102, 2, 24, 114, 90, 55]$

The results of each state transition are shown in Figure 7.9:

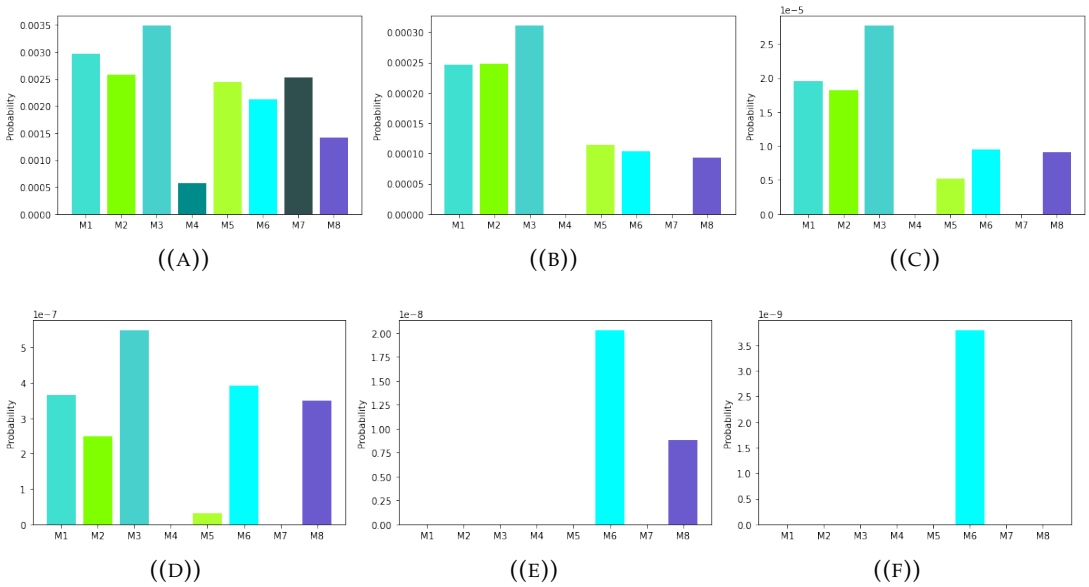


FIGURE 7.9: The probability found by each model after each state transition.

In Figure 7.9, we can see the change of probabilities after each change of state, we can see that both models M_4 and M_7 disappear from the iteration B, which means that these changes of state can not generate the two critical states represented by the two models M_4 and M_7. At iterations A, B, C, and D, we can see that the most probable critical state is the one represented by the model M_3 and can see that during these iterations the probability by the model M_6 increases compared to the other models. At iteration E, we can see that all the models have disappeared except the two models M_6 and M_8. At the last iteration, only the model M_6 remains which classifies the sequence at the critical state represented by model M_6.

7.5 Conclusion

In this chapter, we have treated the Fuel Pool Cooling System as a case study, we have shown in detail how to determine the combinations of basic events that can generate severe accidents in this system, and also showed how we can find all the possible failure scenarios of this system to reach all the possible severe accidents by using our quantum algorithm. We have used the QDTW-β method to classify the sequences. We have shown how we can use the Quantum Hidden Markov Models to track the changing state of the system in real time.

Conclusion

The main objective of this dissertation is to propose quantum algorithms able to solve complex problems in the field of Probabilistic Safety Assessment (PSA) of nuclear power plants and to propose quantum algorithms equivalent to Machine Learning algorithms but with less complexity and more performance.

In the first chapter, we have described some concepts related to PSA problems to provide a good basis to start this dissertation. We have started by giving some generality and history of this field and present the three main levels of it. We have presented the operational methods used in this field conventionally today, the problems, and some motivations for treating this model in the light of quantum computing.

Quantum computing is the principal foundation of this dissertation, to introduce its basics and to give a general idea about the difference between classical and quantum information processing. In chapter two, we have given a mathematical review of quantum computing in a general way and defined what is a quantum system of one qubit, two qubits, and n qubits. In each case, we have presented how we can apply transformations to the states of qubits. We have discussed the Quantum Annealing with the QUBO problem and presented the D-wave system to find the results with it. We have given the different classes of classical and quantum complexity. Some well-known algorithms in the field of quantum computing are presented: Deutsch-Jozsa algorithm, Quantum Fourier transform, Quantum phase estimation, Grover search algorithm, Quantum Walks, and others algorithms. Several quantum machine learning algorithms are presented in this chapter, including the Quantum k-means algorithm, Quantum k-medians algorithm, Quantum Support Vector Machines, Quantum Principal Component Analysis, and Quantum Neural Networks.

The objective of the static PSA problem is to find all the combinations of basic events of a system that can generate serious accidents like the fusion of the core of the nuclear power plant. To find these combinations, we can represent the system by a directed graph and look in it for minimal cuts. So, in the third chapter, we have proposed a quantum algorithm to find all the minimal cuts of a directed graph. More precisely, we propose a quantum algorithm that uses movement oracles to generate as output a superposition of all states that represent the minimal cuts. In this chapter, cuts are represented by a set of vertices, which can separate the source and the terminal of the graph, and they are minimal if they contain just the minimal number of vertices to represent a cut. The complexity of our algorithm is linear, because: it uses only $N + 2$ qubits, N to represent all the possible combinations of vertices and 2 for the control, and it uses N oracles of movements, N being the number of vertices of the graph.

The dynamic part of PSA consists in managing the failure scenarios of the system. These scenarios are represented by sequences of states of the system. The search and process of these sequences in a reasonable time is a very complex problem. Therefore,

in the fourth chapter, we use the graph of states of the system to find all failure scenarios, in it, we look for paths between the current state of the system and the critical failure states. We proposed a quantum algorithm that finds all the paths between two vertices in a DAG with N qubits and M gates (N is the number of vertices of the graph and M is the number of edges). This algorithm allows us to search for all the paths to several destinations from a source vertex at the same time. Unfortunately, due to the size of the quantum computers available today, it is not yet possible to use this algorithm alone with a single quantum circuit. This is why we have proposed a second hybrid algorithm that allows us to call the first one to deal with large graphs. These two algorithms are tested in an IBM quantum simulator with 32 qubits to show how well our algorithms work and to show the effect of our algorithms converging to a set of fixed paths as opposed to the classical random walk.

To calculate the similarity between sequences. We proposed two quantum algorithms, the first one QDTW is a quantum version of the DTW algorithm to compute the similarity between two sequences, by treating the sequences element by element. This algorithm uses $N \times M$ qubits and $3N \times M$ gates to build the circuit that finds the Warping Path between two sequences of sizes N and M. The second algorithm QDTW-β allows us to find this similarity between two sequences using in addition to the case of one to one the case of treating the sub-sequences. These sub-sequences can take different sizes in each sequence and the matching can be passed by two sub-sequences of different sizes. The biggest advantage of this algorithm is, we use only $N \times M$ qubits which allows us to have a speedup in finding the results compared to the classical approach (NP-complete approach). The number of gates increases depending on the β value. Both algorithms are tested and compared with the classical approaches. QDTW gives the same results as the classical approach, except that here it is a quantum algorithm. QDTW-β gives better results than the classical one, either for finding the best matching or for classification.

Also, we have proposed a strategy to learn failure scenarios for a PSA system. We have proposed to use QHMM models to generate failure scenarios from a given state of the system and to identify the probable and no-probable failure scenarios of a system. This strategy gives us several advantages compared to the current approaches used in the field of PSA, among them, by doing the learning only once, we can find the failure scenarios from any state of the system. In addition, it allows us to generate new scenarios that we do not have in the dataset. To test this approach, we used four datasets for two small real systems to show that the QHMMs are more efficient than the HMMs. Furthermore, it shows that the models can detect probable and no-probable failure scenarios.

Staying in problems that have high complexity, clustering algorithms are known that are NP-complete, therefore it's a good idea to find equivalent quantum algorithms that can do these tasks with less complexity. For this purpose, in the fifth chapter, we propose a quantum version of the k-means algorithm with logarithmic complexity. It is an improvement of an existent quantum approach. This reduction of the complexity comes after proposing a strategy to compute the distance between the observations and centers of the clusters at the same time with a single quantum circuit, as well as, the use of Grover's algorithm to search the nearest centroid. We have analyzed different methodologies to build quantum states from classical data to compute distances, we have proposed a quantum version of the Davis-Bouldin index. The quantum version of k-means gives results as in the classical case, except in the

quantum version with less complexity. This algorithm belongs to the algorithms that use universal quantum computers.

On the other hand, for the quantum annealing, in chapter six, we have discussed the quantum version of the two algorithms Balanced k-means and Convex-NMF. We have modified the Quadratic Unconstrained Binary Optimization (QUBO) of the Balanced k-means algorithm proposed recently in the state of the art. We have proposed new constants that handle the belonging of observation of data to two clusters. The constants of the last proposed approach are very generic, which leads the approach to give many non-assigned elements. In our approach, we have proposed a vector of constants where each element of the data set is associated with an element of this vector. Moreover, we have done a comparative analysis between the two approaches to prove that our approach can assign the largest number of data to clusters, and we have shown that our approach gives the best clustering by comparing the Davies-Bouldin index. We have proposed a new approach for the Convex-NMF algorithm in D-wave 2000Q. We have proposed to decompose the Convex-NMF optimization problem on two QUBOs problems: the first to find the matrix G and the second to find the matrix W which minimizes the norm difference between the data matrix X and the matrix product XWG where all elements of G and W are non-negative and on the rows/columns they sum up to 1. To work with real values, we used a transformation to find a binary representation of the problem from a problem with real variables. We have made several tests to demonstrate that our quantum approach is faster than the classical one. Until today, we can't test our approach on large datasets, because the number of qubits in D-wave quantum computers is limited. We hope that in the coming years the number of qubits in quantum computers will increase.

In the seventh and last chapter, we have treated the Fuel Pool Cooling System as a case study, we have shown in detail how to determine the combinations of basic events that can generate severe accidents in this system, and showed how we can find all the possible failure scenarios of this system to reach all the possible severe accidents by using our quantum algorithm. We have used the QDTW-β method to classify the sequences. We have shown how we can use the QHMMs to track the changing state of the system in real-time.

Future work

An important number of tracks are opened to be treated in the near future, either in the static part of our problem or in the dynamic part. Among them, we mention the following:

1. The problem of finding the combinations of basic events that can generate unacceptable consequences in an installation is the essential basis for the safety analysis of the installations in nuclear power plants. This problem is NP-complete, we have proposed in chapter 3 a strategy based on directed graphs to find all these combinations. Unfortunately, today, we can't use this algorithm to deal with large systems (systems with a large number of components) due to the size of the available quantum computers (the number of available qubits). Following this issue, the main question that appears is: how can we benefit in our problem from the power of quantum computers even with a reduced number of qubits? In answering this question, several other issues arise:

- How can we make our algorithm Hybrid to handle large systems part by part?
- How can we split the large system into several small systems and process them separately? If we use the balanced K-means algorithm proposed in chapter 6 to split the large directed graph, how can we aggregate the results of each part to find the final results?

2. If we take the original representation of our problem with fault trees, the question to be asked is: How to find a quantum circuit able to find all the minimal cuts of a fault tree with the smallest number of qubits?

3. In the dynamic part of our issue, from the system failure scenarios, how can we find the weaknesses in the system? and with the calculation of the similarity between these scenarios, how to choose the most important repairs that should be done quickly to avoid the greatest number of highly probable scenarios?

4. Testing all the algorithms on real quantum computers with an advanced number of qubits, such as the IBM quantum computer with 127 qubits, thanks to the possibility of a partnership between EDF R&D and IBM.

Publications

- A. Zaiou, Y. Bennani, B. Matei and M. Hibti, "Balanced K-means using Quantum annealing," IEEE Symposium Series on Computational Intelligence, SSCI 2021, Orlando, FL, USA, December 5-7, 2021.
- A. Zaiou, Y. Bennani, B. Matei and M. Hibti, "Convex Non-negative Matrix Factorization Through Quantum Annealing," IEEE 7th Int Conf on Data Science Systems 20-22, 2021.
- A. Zaiou, Y. Bennani, M. Hibti and B. Matei, "Quantum Approach for Vertex Separator Problem in Directed Graphs," Artificial Intelligence Applications and Innovations - 18th International Conference, 2022, Hersonissos, Crete, Greece, June 17-20, 2022,
- K. Benlamine and Y. Bennani and A. Zaiou and M. Hibti and B. Matei and N. Grozavu, "Distance Estimation for Quantum Prototypes Based Clustering," Neural Information Processing - 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12-15, 2019
- A. Zaiou, Y. Bennani, B. Matei and M. Hibti, "A quantum learning approach based on Hidden Markov Models for failure scenarios generation," in 2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML), Hangzhou, China, 2022 pp. 62-67.
- A. Zaiou, Y. Bennani, B. Matei and M. Hibti, "Quantum approach to find all paths between two vertices on an a-cyclic directed graph,"
- A. Zaiou, Y. Bennani, B. Matei and M. Hibti, "Quantum approaches to calculate the similarity between sequences,"
- K. Benlamine and Y. Bennani and A. Zaiou and M. Hibti and B. Matei and N. Grozavu, "Clustering Quantique à base de prototypes," Conférence Internationale Francophone sur la Science des Données (CIFSD) Actes de la 9e édition, 2021
- A. Zaiou, Y. Bennani, M. Hibti and B. Matei, "Algorithme quantique pour trouver les séparateurs d'un graphe orientÉ," Conférence Internationale Francophone sur la Science des Données (CIFSD) Actes de la 9e édition, 2021
- A. Zaiou, Y. Bennani, M. Hibti and B. Matei, "Marches quantiques pour déterminer les scénarios de défaillance d'un systéme," Congrès de maîtrise des risques et de sûreté de fonctionnement, $\lambda\mu$23 , du 10 au 13 octobre 2022 à EDF Lab - Paris Saclay
- M. Hibti, Y. Bennani, A. Zaiou and B. Matei, "Les EPS et les algorithmes quantiques," Congrès de maîtrise des risques et de sûreté de fonctionnement, $\lambda\mu$23 , du 10 au 13 octobre 2022 à EDF Lab - Paris Saclay

Bibliography

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[ABG06] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. "Machine learning in a quantum world". In: *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer. 2006, pp. 431–442.

[ABG07] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. "Quantum clustering algorithms". In: *Proceedings of the 24th international conference on machine learning*. 2007, pp. 1–8.

[ACDF89] Bruno Apolloni, C Carvalho, and Diego De Falco. "Quantum stochastic optimization". In: *Stochastic Processes and their Applications* 33.2 (1989), pp. 233–244.

[ADFCB88] Bruno Apolloni, D De Falco, and N Cesa-Bianchi. *A numerical implementation of" quantum annealing"*. Tech. rep. 1988.

[Adh+20] Sandesh Adhikary et al. "Expressiveness and learning of hidden quantum markov models". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 4151–4161.

[ADZ93] Yakir Aharonov, Luiz Davidovich, and Nicim Zagury. "Quantum random walks". In: *Physical Review A* 48.2 (1993), p. 1687.

[AK99] Ashish Ahuja and Sanjiv Kapoor. "A quantum algorithm for finding the maximum". In: *arXiv preprint quant-ph/9911082* (1999).

[AL94] Cleve Ashcraft and Joseph WH Liu. "A partition improvement algorithm for generalized nested dissection". In: *Boeing Computer Services, Seattle, WA, Tech. Rep. BCSTECH-94-020* (1994).

[Alb83] Peter M Alberti. "A note on the transition probability over C*-algebras". In: *Letters in Mathematical Physics* 7.1 (1983), pp. 25–32.

[Alo+09] Daniel Aloise et al. "NP-hardness of Euclidean sum-of-squares clustering". In: *Machine learning* 75.2 (2009), pp. 245–248.

[Ang+03] Davide Anguita et al. "Quantum optimization for training support vector machines". In: *Neural Networks* 16.5-6 (2003), pp. 763–770.

[Art+20] Davis Arthur et al. "Balanced k-means clustering on an adiabatic quantum computer". In: *arXiv preprint arXiv:2008.04419* (2020).

[AZCN21] Ali Al Zoobi, David Coudert, and Nicolas Nisse. "On the complexity of finding *k* shortest dissimilar paths in a graph". PhD thesis. Inria; CNRS; I3S; UCA, 2021.

[Bar+15] Jens Barth et al. "Stride segmentation during free walk movements using multi-dimensional subsequence dynamic time warping on inertial sensor data". In: *Sensors* 15.3 (2015), pp. 6419–6440.

[Bau+18] Christian Bauckhage et al. "Adiabatic Quantum Computing for Kernel k= 2 Means Clustering." In: *LWDA*. 2018, pp. 21–32.

[Bau+19] C. Bauckhage et al. "A QUBO Formulation of the k-Medoids Problem". In: *LWDA*. 2019.

[BB04] Claus Bahlmann and Hans Burkhardt. "The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.3 (2004), pp. 299–310.

[BCA03] Todd A Brun, Hilary A Carteret, and Andris Ambainis. "Quantum to classical transition for random walks". In: *Physical review letters* 91.13 (2003), p. 130602.

[Ben+19] Kaoutar Benlamine et al. "Distance Estimation for Quantum Prototypes Based Clustering". In: *Neural Information Processing*. Ed. by Tom Gedeon and al. Springer International Publishing, 2019, pp. 561–572.

[BI80] Robert D Burns III. "WASH 1400-reactor safety study". In: *Prog. Nucl. Energy* 6.1-3 (1980), pp. 117–140.

[Big86] Norman Biggs. *The traveling salesman problem a guided tour of combinatorial optimization*. 1986.

[BJ92] Thang Nguyen Bui and Curt Jones. "Finding good approximate vertex and edge partitions is NP-hard". In: *Information Processing Letters* 42.3 (1992), pp. 153–159.

[BK59] Richard Bellman and Robert Kalaba. "A mathematical theory of adaptive control processes". In: *Proceedings of the National Academy of Sciences of the United States of America* 45.8 (1959), p. 1288.

[BR93] Avrim L Blum and Ronald L Rivest. "Training a 3-node neural network is NP-complete". In: *Machine learning: From theory to applications*. Springer, 1993, pp. 9–28.

[Bra+00] Gilles Brassard et al. "Quantum Amplitude Amplification and Estimation". In: 2000.

[Bun+14] Paul I. Bunyk et al. "Architectural Considerations in the Design of a Superconducting Quantum Annealing Processor". In: *IEEE Transactions on Applied Superconductivity* 24.4 (2014), pp. 1–10. DOI: 10.1109/TASC.2014.2318294.

[BV97] Ethan Bernstein and Umesh Vazirani. "Quantum complexity theory". In: *SIAM Journal on computing* 26.5 (1997), pp. 1411–1473.

[BW96] Beate Bollig and Ingo Wegener. "Improving the variable ordering of OBDDs is NP-complete". In: *IEEE Transactions on computers* 45.9 (1996), pp. 993–1002.

[Cai+15] X-D Cai et al. "Entanglement-based machine learning on a quantum computer". In: *Physical review letters* 114.11 (2015), p. 110504.

[Cat15] Maria Catrina. "Railway station routing algorithm using the backtracking method". In: *University Politehnica of Bucharest Scientific Bulletin* 77.4 (2015), pp. 335–346.

[Che+09] Yueguo Chen et al. "Efficient processing of warping time series join of motion capture data". In: *2009 IEEE 25th International Conference on Data Engineering*. IEEE. 2009, pp. 1048–1059.

[CKH95] Pierluigi Crescenzi, Viggo Kann, and M Halldórsson. *A compendium of NP optimization problems*. 1995.

[Cor01] A. Corradini. "Dynamic time warping for off-line recognition of a small gesture vocabulary". In: *Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*. 2001, pp. 82–89. DOI: 10.1109/RATFG.2001.938914.

[DAGa] Finding all paths between a set of vertices in a DAG. "Computer Science Stack Exchange." In.

[DAGb] Number of paths between two nodes in a DAG. "Stack Overflow." In.

[DAPN21] Prasanna Date, Davis Arthur, and Lauren Pusey-Nazzaro. "QUBO formulations for training machine learning models". In: *Scientific Reports* 11.1 (2021), pp. 1–10.

[Dav+19] Timothy A Davis et al. "Algorithm XXX: Mongoose, a graph coarsening and partitioning library". In: *ACM Trans. Math. Software* (2019).

[DB79] David L Davies and Donald W Bouldin. "A cluster separation measure". In: *IEEE transactions on pattern analysis and machine intelligence* 2 (1979), pp. 224–227.

[DG17] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[DH96] Christoph Durr and Peter Hoyer. "A quantum algorithm for finding the minimum". In: *arXiv preprint quant-ph/9607014* (1996).

[DHS05] Chris Ding, Xiaofeng He, and Horst D Simon. "On the equivalence of nonnegative matrix factorization and spectral clustering". In: *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM. 2005, pp. 606–610.

[Din+08] Hui Ding et al. "Querying and mining of time series data: experimental comparison of representations and distance measures". In: *Proceedings of the VLDB Endowment* 1.2 (2008), pp. 1542–1552.

[Dix+20] Vivek Dixit et al. "Training and classification using a restricted boltzmann machine on the d-wave 2000q". In: *arXiv preprint arXiv:2005.03247* (2020).

[DJ92] David Deutsch and Richard Jozsa. "Rapid solution of problems by quantum computation". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (1992), pp. 553–558.

[DP20] Prasanna Date and Thomas Potok. "Adiabatic Quantum Linear Regression". In: (Aug. 2020).

[EFV07] Alon Efrat, Quanfu Fan, and Suresh Venkatasubramanian. "Curve matching, time warping, and light fields: New algorithms for computing similarity between curves". In: *Journal of Mathematical Imaging and Vision* 27.3 (2007), pp. 203–216.

[ER08] Waiyawuth Euachongprasit and Chotirat Ann Ratanamahatana. "Efficient multimedia time series data retrieval under uniform scaling and normalisation". In: *European Conference on Information Retrieval*. Springer. 2008, pp. 506–513.

[Fel+20] Sebastian Feld et al. "The Dynamic Time Warping Distance Measure as Q U BO Formulation". In: *2020 5th International Conference on Computer and Communication Systems (ICCCS)*. 2020, pp. 946–950. DOI: 10.1109/ICCCS49078.2020.9118469.

[Fin17] Klint Finley. "Quantum computing is real, and d-wave just open-sourced it". In: *Wired (magazine). Condé Nast.(11 January 2017)* (2017).

[Fin+94] Aleta Berk Finnila et al. "Quantum annealing: A new method for minimizing multidimensional functions". In: *Chemical physics letters* 219.5-6 (1994), pp. 343–348.

[FM82] Charles M Fiduccia and Robert M Mattheyses. "A linear-time heuristic for improving network partitions". In: *19th design automation conference*. IEEE. 1982, pp. 175–181.

[FS87] Steven J Friedman and Kenneth J Supowit. "Finding the optimal variable ordering for binary decision diagrams". In: *24th ACM/IEEE Design Automation Conference*. IEEE. 1987, pp. 348–356.

[GHZ12] Hongyu Guo, Dongmei Huang, and Xiaoqun Zhao. "An algorithm for spoken keyword spotting via subsequence DTW". In: *2012 3rd IEEE International Conference on Network Infrastructure and Digital Content*. IEEE. 2012, pp. 573–576.

[GJ06] Jie Gu and Xiaomin Jin. "A simple approximation for dynamic time warping search in large time series database". In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer. 2006, pp. 841–848.

[GJ79] Michael R Garey and David S Johnson. "Computers and intractability". In: *A Guide to the* (1979).

[Gro96] Lov K Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219.

[Gro98] Lov K Grover. "A framework for fast quantum mechanical algorithms". In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 1998, pp. 53–62.

[Gu10] Shi-Jian Gu. "Fidelity approach to quantum phase transitions". In: *International Journal of Modern Physics B* 24.23 (2010), pp. 4371–4458.

[Has20] Matthew B. Hastings. "Classical and quantum algorithms for tensor principal component analysis". In: *Quantum* (2020). ISSN: 2521327X. DOI: 10.22331/q-2020-02-27-237. arXiv: 1907.12724.

[HH15] William W Hager and James T Hungerford. "Continuous quadratic programming formulations of optimization problems on graphs". In: *European Journal of Operational Research* 240.2 (2015), pp. 328–337.

[HHS18] William W Hager, James T Hungerford, and Ilya Safro. "A multilevel bilinear programming algorithm for the vertex separator problem". In: *Computational Optimization and Applications* 69.1 (2018), pp. 189–223.

[Hib13] Mohamed Hibti. "What if we revisit evaluation of PSA models with network algorithms?" In: vol. 2. Jan. 2013, pp. 1387–1398.

[Hic83] JW Hickman. "PRA procedures guide: a guide to the performance of probabilistic risk assessments for nuclear power plants". In: *NUREG/CR-2300* (1983).

[HLT11] Minh Hoai, Zhen-Zhong Lan, and Fernando De la Torre. "Joint segmentation and classification of human actions in video". In: *CVPR 2011*. IEEE. 2011, pp. 3265–3272.

[HR98] Bruce Hendrickson and Edward Rothberg. "Improving the run time and quality of nested dissection ordering". In: *SIAM Journal on Scientific Computing* 20.2 (1998), pp. 468–489.

[Inta] *Procedures for Conducting Probabilistic Safety Assessments of Nuclear Power Plants (Level 1)*. Safety Series 50-P-4. Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY, 1992. ISBN: 92-0-102392-8. URL: https://www.iaea.org/publications/3759/procedures-for-conducting-probabilistic-safety-assessments-of-nuclear-power-plants-level-1.

[Intb] *Procedures for Conducting Probabilistic Safety Assessments of Nuclear Power Plants (Level 3): Off-Site Consequences and Estimation of Risks to the Public: A Safety Practice*. Safety Series 50-P-12. Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY. ISBN: 92-0-103996-4. URL: https://www.iaea.org/publications/5127/procedures-for-conducting-probabilistic-safety-assessments-of-nuclear-power-plants-level-3-off-site-consequences-and-estimation-of-risks-to-the-public-a-safety-practice.

[Jae07] Gregg Jaeger. *Quantum information*. Springer, 2007.

[Joz94] Richard Jozsa. "Fidelity for mixed quantum states". In: *Journal of modern optics* 41.12 (1994), pp. 2315–2323.

[KAVL19] Sumsam Ullah Khan, Ahsan Javed Awan, and Gemma Vall-Llosera. "K-means clustering on noisy intermediate scale quantum computers". In: *arXiv preprint arXiv:1909.12183* (2019).

[KD16] S. Kolodziej and T. Davis. "Vertex separators with mixed-integer linear optimization". In: *17th SIAM Conference on Parallel Processing for Scientific Computing* (2016).

[KD20] Scott P Kolodziej and Timothy A Davis. "Generalized Gains for Hybrid Vertex Separator Algorithms". In: *2020 Proceedings of the SIAM Workshop on Combinatorial Scientific Computing*. SIAM. 2020, pp. 96–105.

[Kem03] Julia Kempe. "Quantum random walks: an introductory overview". In: *Contemporary Physics* 44.4 (2003), pp. 307–327.

[Ker+19] Iordanis Kerenidis et al. "q-means: A quantum algorithm for unsupervised machine learning". In: *Advances in Neural Information Processing Systems* 32 (2019).

[Ker80] Deutsche Risikostudie Kernkraftwerke. "Der Bundesminister für Forschung und Technologie, Verlag T ÜV Rheinland". In: (1980).

[KK98] Ton Kloks and Dieter Kratsch. "Listing all minimal separators of a graph". In: *SIAM Journal on Computing* 27.3 (1998), pp. 605–613.

[KL70] Brian W Kernighan and Shen Lin. "An efficient heuristic procedure for partitioning graphs". In: *The Bell system technical journal* 49.2 (1970), pp. 291–307.

[KN98] Tadashi Kadowaki and Hidetoshi Nishimori. "Quantum annealing in the transverse Ising model". In: *Physical Review E* 58.5 (1998), p. 5355.

[Kol19] Scott Parker Kolodziej. "Computational Optimization Techniques for Graph Partitioning". PhD thesis. 2019.

[KS01] Tamer Kahveci and Ambuj Singh. "Variable length queries for time series data". In: *Proceedings 17th International Conference on Data Engineering*. IEEE. 2001, pp. 273–282.

[KSG02] Tamer Kahveci, Ambuj Singh, and Aliekber Gurel. "Similarity searching for multi-attribute sequences". In: *Proceedings 14th International Conference on Scientific and Statistical Database Management*. IEEE. 2002, pp. 175–184.

[Kum+18] Vaibhaw Kumar et al. "Quantum annealing for combinatorial clustering". In: *Quantum Information Processing* 17.2 (2018), pp. 1–14.

[KZ07] Ana Kuzmanic and Vlasta Zanchi. "Hand shape classification using DTW and LCSS as similarity measures for vision-based gesture recognition system". In: *EUROCON 2007 - The International Conference on "Computer as a Tool"*. 2007, pp. 264–269. DOI: 10.1109/EURCON.2007.4400350.

[Llo10] Seth Lloyd. "Quantum algorithm for solving linear systems of equations". In: *APS March Meeting Abstracts*. Vol. 2010. 2010, pp. D4–002.

[LLo82] S.P. LLoyd. "Least Squares Quantization in PCM". In: *Special Issue on Quantization, IEEE Tr. on Information Theory* 28.2 (1982), pp. 129–137.

[LMR13] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum algorithms for supervised and unsupervised machine learning". In: *arXiv preprint arXiv:1307.0411* (2013).

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum principal component analysis". In: *Nature Physics* 10.9 (2014), pp. 631–633.

[LR76] Hyafil Laurent and Ronald L Rivest. "Constructing optimal binary decision trees is NP-complete". In: *Information processing letters* 5.1 (1976), pp. 15–17.

[LS01] Daniel D Lee and H Sebastian Seung. "Algorithms for non-negative matrix factorization". In: *Advances in neural information processing systems*. 2001, pp. 556–562.

[LT79] Richard J Lipton and Robert Endre Tarjan. "A separator theorem for planar graphs". In: *SIAM Journal on Applied Mathematics* 36.2 (1979), pp. 177–189.

[LT80] Richard J Lipton and Robert Endre Tarjan. "Applications of a planar separator theorem". In: *SIAM journal on computing* 9.3 (1980), pp. 615–627.

[Mac+67] James MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[Man15] Nicholas Mancuso. "Algorithm that finds the number of simple paths from s to t in G". In: *Computer Science Stack Exchange. URL: https://cs. stackexchange. com/q/3087* (2015).

[McG14] Catherine C McGeoch. "Adiabatic quantum computation and quantum annealing: Theory and practice". In: *Synthesis Lectures on Quantum Computing* 5.2 (2014), pp. 1–93.

[Mon05] Ashley Montanaro. "Quantum walks on directed graphs". In: *arXiv preprint quant-ph/0504116* (2005).

[Mot+04] Mikko Mottonen et al. "Transformation of quantum states using uniformly controlled rotations". In: *arXiv preprint quant-ph/0407010* (2004).

[MRR80] C. Myers, L. Rabiner, and A. Rosenberg. "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.6 (1980), pp. 623–635. DOI: 10.1109/TASSP.1980.1163491.

[Mül07a] Meinard Müller. "Dtw-based motion comparison and retrieval". In: *Information Retrieval for Music and Motion* (2007), pp. 211–226.

[Mül07b] Meinard Müller. "Dynamic time warping". In: *Information retrieval for music and motion* (2007), pp. 69–84.

[Mur89] Tadao Murata. "Petri nets: Properties, analysis and applications". In: *Proceedings of the IEEE* 77.4 (1989), pp. 541–580.

[NC02] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. 2002.

[NC11] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 10th. New York, NY, USA: Cambridge University Press, 2011. ISBN: 1107002176, 9781107002173.

[NDW16] Philipp Niemann, Rhitam Datta, and Robert Wille. "Logic synthesis for quantum state generation". In: *2016 IEEE 46th International Symposium on Multiple-Valued Logic (ISMVL)*. IEEE. 2016, pp. 247–252.

[NR07] Vit Niennattrakul and Chotirat Ann Ratanamahatana. "On clustering multimedia time series data using k-means and dynamic time warping". In: *2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*. IEEE. 2007, pp. 733–738.

[NVDS18] Florian Neukart, David Von Dollen, and Christian Seidel. "Quantum-assisted cluster analysis". In: *arXiv preprint arXiv:1803.02886* (2018).

[OA18] Daniele Ottaviani and Alfonso Amendola. "Low rank non-negative matrix factorization with d-wave 2000q". In: *arXiv preprint arXiv:1808.08721* (2018).

[Ort+91] NR Ortiz et al. "Use of expert judgment in NUREG-1150". In: *Nuclear Engineering and Design* 126.3 (1991), pp. 313–331.

[O18] Daniel O'Malley et al. "Nonnegative/binary matrix factorization with a d-wave quantum annealer". In: *PloS one* 13.12 (2018), e0206653.

[PE10] Claire Pagetti-ENSEEIHT. "Module de sûreté de fonctionnement". In: (2010).

[Ped+19a] Edwin Pednault et al. *Leveraging secondary storage to simulate deep 54-qubit sycamore circuits*. 2019. arXiv: 1910.09534.

[Ped+19b] Edwin Pednault et al. "Leveraging secondary storage to simulate deep 54-qubit sycamore circuits". In: *arXiv preprint arXiv:1910.09534* (2019).

[PKG11] François Petitjean, Alain Ketterlin, and Pierre Gançarski. "A global averaging method for dynamic time warping, with applications to clustering". In: *Pattern recognition* 44.3 (2011), pp. 678–693.

[Pre12] John Preskill. "Quantum computing and the entanglement frontier". In: *arXiv preprint arXiv:1203.5813* (2012).

[PT94] Pentti Paatero and Unto Tapper. "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values". In: *Environmetrics* 5.2 (1994), pp. 111–126.

[RH03] Marvin Rausand and Arnljot Hoyland. *System reliability theory: models, statistical methods, and applications*. Vol. 396. John Wiley & Sons, 2003.

[Ros+17] Marcelo Rosa et al. "An anchored dynamic time-warping for alignment and comparison of swallowing acoustic signals". In: *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2017, pp. 2749–2752.

[RS15] Enno Ruijters and Mariëlle Stoelinga. "Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools". In: *Computer science review* 15 (2015), pp. 29–62.

[SC78] H. Sakoe and S. Chiba. "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26.1 (1978), pp. 43–49. DOI: 10.1109/TASSP.1978.1163055.

[Sch95] Benjamin Schumacher. "Quantum coding". In: *Physical Review A* 51.4 (1995), p. 2738.

[SFY07] Yasushi Sakurai, Christos Faloutsos, and Masashi Yamamuro. "Stream monitoring under the time warping distance". In: *2007 IEEE 23rd International Conference on Data Engineering*. IEEE. 2007, pp. 1046–1055.

[SGB18] Siddarth Srinivasan, Geoff Gordon, and Byron Boots. "Learning hidden quantum Markov models". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 1979–1987.

[Sho99a] Peter W Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". In: *SIAM review* 41.2 (1999), pp. 303–332.

[Sho99b] Peter W Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". In: *SIAM review* 41.2 (1999), pp. 303–332.

[SMR08] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge, 2008.

[Tal+09] A Talon et al. "Analyse de risques: Identification et estimation: Démarches d'analyse de risques-Méthodes qualitatives d'analyse de risques". In: *Module de l'Université Numérique Ingénierie et Technologie _ Fondation unit. Available via http://webdav-noauth. unitc. fr/files/perso/hniandou/cyberrisques2/etage_3_aur html* (2009).

[Uhl76] Armin Uhlmann. "The "transition probability" in the state space of a-algebra". In: *Reports on Mathematical Physics* 9.2 (1976), pp. 273–279.

[UMNM17] Hayato Ushijima-Mwesigwa, Christian FA Negre, and Susan M Mniszewski. "Graph partitioning using quantum annealing on the d-wave system". In: (2017), pp. 22–29.

[VA12] Salvador Elías Venegas-Andraca. "Quantum walks: a comprehensive review". In: *Quantum Information Processing* 11.5 (2012), pp. 1015–1106.

[Vah+21] Ensar Vahapoglu et al. "Single-electron spin resonance in a nanoelectronic device using a global field". In: *Science Advances* 7.33 (2021), eabg9158. DOI: 10.1126/sciadv.abg9158.

[Val79] Leslie G Valiant. "The complexity of enumeration and reliability problems". In: *SIAM Journal on Computing* 8.3 (1979), pp. 410–421.

[Wan+17] Kwok Ho Wan et al. "Quantum generalisation of feedforward neural networks". In: *npj Quantum information* 3.1 (2017), pp. 1–8.

[Wer+18] Kamil Wereszczyński et al. "Quantum computing for clustering big datasets". In: (2018), pp. 276–280.

[WHB19] Daochen Wang, Oscar Higgott, and Stephen Brierley. "Accelerated variational quantum eigensolver". In: *Physical Review Letters* (2019). ISSN: 10797114. DOI: 10.1103/PhysRevLett.122.140504. arXiv: 1802.00171.

[Wil+20] Dennis Willsch et al. "Support vector machines on the D-Wave quantum annealer". In: *Computer physics communications* 248 (2020), p. 107006.

[WJ94] Lusheng Wang and Tao Jiang. "On the complexity of multiple sequence alignment". In: *Journal of computational biology* 1.4 (1994), pp. 337–348.

[WKS18] Nathan Wiebe, Ashish Kapoor, and Krysta M Svore. "Quantum nearest-neighbor algorithms for machine learning". In: *Quantum Information and Computation* 15 (2018).

[XHP99] Y.-L. Xie, P.K. Hopke, and P. Paatero. "Positive matrix factorization applied to a curve resolution problem". In: *Journal of Chemometrics* 12.6 (1999), pp. 357–364.

[XXR09] Haiping Xu, Liudong Xing, and Ryan Robidoux. "Drbd: Dynamic reliability block diagrams for system reliability modelling". In: *International journal of computers and applications* 31.2 (2009), pp. 132–141.

[Zai+21a] Ahmed Zaiou et al. "Balanced K-means using Quantum annealing". In: *IEEE Symposium Series on Computational Intelligence, SSCI 2021, Orlando, FL, USA, December 5-7, 2021*. IEEE, 2021, pp. 1–7. DOI: 10.1109/SSCI50451.2021.9659997. URL: https://doi.org/10.1109/SSCI50451.2021.9659997.

[Zai+21b] Ahmed Zaiou et al. "Convex Non-negative Matrix Factorization Through Quantum Annealing". In: *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Haikou, Hainan, China, December 20-22, 2021*. IEEE, 2021, pp. 1253–1258. DOI: 10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00191. URL: https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00191.

[Zai+22] Ahmed Zaiou et al. "Quantum Approach for Vertex Separator Problem in Directed Graphs". In: *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer. 2022, pp. 495–506.

[Zbi+20] Stefanie Zbinden et al. "Embedding algorithms for quantum annealers with chimera and pegasus connection topologies". In: *International Conference on High Performance Computing*. Springer. 2020, pp. 187–206.

[Zha+19] Jian Zhao et al. *State preparation based on quantum phase estimation*. 2019. arXiv: 1912.05335.

[Zho+20] Han-Sen Zhong et al. "Quantum computational advantage using photons". In: *Science* 370.6523 (2020), pp. 1460–1463.

[ZJ10] Ming-Jie Zhao and Herbert Jaeger. "Norm-observable operator models". In: *Neural computation* 22.7 (2010), pp. 1927–1959.