Faculteit Bedrijf en Organisatie

How will quantum computing affect the mainframe environment and its applications?

Lukas Marivoet

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Martijn Saelens
Co-promotor:
Francis Harkins

Instelling: IBM

Academiejaar: 2019-2020

Tweede examenperiode

Faculty of Business and Information Management

How will quantum computing affect the mainframe environment and its applications?

Lukas Marivoet

Thesis submitted in partial fulfilment of the requirements for the degree of
professional bachelor of applied computer science

Promotor:
Martijn Saelens
Co-promotor:
Francis Harkins

Institution: IBM

Academic year: 2019-2020

Second examination period

Preface

Why did I choose to explore the quantum realm without any specific pre-knowledge/ education? Quantum computing is being transformed to a real buzzword much like data science used to be. The field has been opened up from just highly specialised academics to an open source community willing to teach outsiders from the very basics.

Also the mere fact that the field of quantum computing is developing to a profitable and sustainable business so rapidly has astonished me from my very first contact with the environment.

I would like to thank Frank Harkins from IBM for always being available to have a discussion about quantum computing and how it will influence our societies and even our very nature of problem-solving.

But most of all I would like to congratulate the research environment around quantum computing on how accepting and supportive they are in all interested parties. In the next decade this will only become more apparent and obvious to a point where quantum computing becomes an essential part of solving anything in a reasonable time scheme.

So that is exactly why this paper will serve as a great starting tool for a computer scientist that is interested in the multiple ways Quantum Computers could influence the general sector.

Samenvatting

Zoals het onderzoek aantoont is het onderzoeksgebied van kwantum computers nog in zijn beginjaren en moeten we kritisch blijven ten opzichte van elke nieuwe uitgave in verband met nieuw onderzoek naar kwantum computers. Echter is het ook belangrijk dat we buiten het puur theoretische deel ook effectief op zoek gaan naar de praktische toepassingen en/ of inzichten in onze huidige processen met evt. de toepassing van kwantum verwerking in deze bestaande processen.

In het onderzoek proberen we een duidelijk beeld weer te geven aan de lezer, zodat hij/ zij zelfstandig kan nadenken over toepassingen en/ of zelf toevoegingen kan maken aan de vele open source gemeenschappen op Github. Dit hebben we proberen te bereiken door enkele praktische weergaves te maken met de hulp van het Python-framework Qiskit over de uitvoering op een kwantum systeem. De resultaten wijzen inderdaad erop dat we bestaande problemen ook kunnen oplossen met kwantum algoritmes, maar zoals te zien aan de werkelijke uitvoeringen op de echte kwantum computers van IBM is het moeilijk om deze nieuwe technologieën al meteen te gebruiken in bestaande productiesystemen.

In de paper is er ook een focus gelegd op de mogelijke samenwerking van de mainframe in het verhaal over kwantum computers. De verwachtingen zijn al enorm over de samenwerking tussen onze huidige supercomputers en een kwantum systeem, maar dit betekent dus ook dat er een enorm potentieel bestaat tussen een mainframe machine en een kwantum systeem. De potentiële winst van kwantum wordt voorzien in enorme versnellingen van data verwerkingen en dat is nu exact waar een mainframe machine zo krachtig in is, het genereren van enorme hoeveelheden data.

Er ligt ook een sterke nadruk op het tonen van alle kanten van kwantum computers om er zeker van te zijn dat de lezer een volledig beeld heeft van het complete onderwerp.

Abstract

The field of quantum computers is still in its infancy and we must remain critical of any new publication related to new insights. However, it is also important that beyond the purely theoretical part, we also effectively look for practical applications and/or insights into our current processes with the possible application of quantum processing in them.

In the research we try to present a more clear picture to the reader, so that he or she can independently think about applications and/or make additions to the many open source communities on Github. We have tried to achieve this by making some practical showcases with the help of the Python framework Qiskit for quantum computing. The results indeed point to the possible execution of quantum algorithms on already existing computer science problems. But as can be seen from the actual results on IBM's real quantum computers, it is difficult to use these new technologies in existing production systems at this point in time.

In the paper there is also a focus on the cooperation of the mainframe in the story of quantum computers. The expectations are huge about the cooperation between our current supercomputers and a quantum system, but this also means that there is a huge potential between a mainframe machine and a quantum system. The benefits of quantum are mostly expected in huge accelerations of data processing and that is exactly what a mainframe machine is so powerful at, generating huge amounts of data.

There is also a strong emphasis on showing all sides of quantum computing as to make sure the reader has a full picture of the whole field.

Contents

1 Introduction ... 15

1.1 Problem Statement 16

1.2 Research question 16

1.3 Research objective 16

1.4 Structure of this bachelor thesis 17

2 Quantum Essentials ... 19

2.1 The Qubit and its representations 19

2.2 Superposition and entanglement 21

2.2.1 Superposition ... 21

2.2.2 Entanglement ... 22

2.2.3 Quantum advantage ... 23

2.3 Quantum decoherence 23

2.4 Quantum gates **25**
2.4.1 Hadamard gate 25
2.4.2 Pauli-X gate 26
2.4.3 CNOT gate 26
2.4.4 Toffoli, CCNOT gate 26

3 Real-world solutions with Quantum 29
3.0.1 Quantum computing and traditional computing 29
3.0.2 Quantum computing and the mainframe 30
3.0.3 Quantum computing and Machine Learning 31

4 Practical demonstration with Qiskit 33
4.0.1 Grover's search algorithm in a practical fashion 33
4.0.2 Data-encoding in QC 36
4.0.3 Mainframe computing with QC 37
4.0.4 Future prospects 38

5 Discussion 39
5.0.1 Quantum computing for now 39
5.0.2 Quantum computing and its myths 40
5.0.3 Quantum computing as an addition 40
5.0.4 Future works 41

A Using Qiskit yourself 43
A.1 Setting up your Python-Qiskit environment **43**
A.2 Connecting your Python environment with IBMQ **44**
A.3 Running the quantum decoherence example **44**

A.4 Running the practical 3-SAT example 45

B Research Proposition 49

B.1 Introduction 49

B.1.1 Situating the subject 49

B.2 State-of-the-art 50

B.2.1 Prior knowledge 50

B.2.2 Recent developments 50

B.3 Methodology 50

B.4 Expected results 51

B.5 Expected conclusions 51

Bibliography 53

List of Figures

2.1 A Bloch sphere representation of a qubit in the elevated $|1\rangle$ state. The Bloch sphere clearly indicates that the state of a qubit has a complex aspect to it. By representing it in a sphere, we are able to showcase how the direction of the arrow has a probability with it. In this case the arrow has a probability of 1.00 for the elevated $|1\rangle$ state. 20

2.2 This is the quantum circuit that puts a qubit in an elevated state using the Pauli-X gate followed up with a measurement, to show off how quantum decoherence influences the results. 24

2.3 These are the percentages after performing the circuit on a real device from which it is visible that quantum decoherence has taken place on the initial $|1\rangle$ state to the $|0\rangle$ state. 24

4.1 The 3-SAT problem that we have processed manually, using the quantum simulator and using the real quantum device. 34

4.2 These are the results of executing the algorithm for the 3-SAT problem on a **quantum simulator** that comes with Qiskit. The encoding refers to the TRUE/FALSE value of the x,y,z respectively 35

4.3 These are the results of executing the algorithm for the 3-SAT problem using 15 qubits on a **real quantum device** that comes with Qiskit. The encoding refers to the TRUE/FALSE value of the x,y,z respectively 36

4.4 The filled-in example for showing how performing this boolean equation on the real device has been influenced too much by quantum decoherence. 36

5.1 The platform we used for executing the 3-SAT algorithm on a real quantum device. © Copyright 2020, Qiskit Development Team Last updated on 2020/05/14. 41

A.1 If you are able to execute this command in your Python environment, you have installed Qiskit successfully. 43

A.2 The necessary code to connect your Python environment to the external quantum devices. 44

A.3 The simulated results plotted directely in a Jupyter notebook. 47

A.4 This is the visual result of running the practical version of this paper and displaying it on the ibmq website. 48

1. Introduction

Why does everyone suddenly jump on the subject of quantum computing **(QC)** and why would it concern anyone at this point in time? Well we are rapidly reaching the limits of how small we are able to create the transistors on a chip, Moore's Law may very well be about to end. Currently we are able to create transistors so small that they themselves start being influenced by the quantum world which would undermine the whole point of building smaller and smaller components that are faster than its predecessors. (Hartnett, 2019)

The quantum field itself is also rapidly expanding due to the practical executions of QC, much in the same way data science has been expanding for the last 2 decades. Only 20 years ago data was something nice to have in a business to gain a possible edge over opponents, now data is the lifeblood for many of those companies. Data is what drives research, competitive advantage and various innovations. QC could offer our way of data handling and processing a surprising speed boost and expansion into regions we just were not able to even understand due to its quantum nature e.g. in sectors like chemistry, astronomy, physics... And this is exactly why the mainframe environment could tie in so nicely into the research towards a classical and quantum computational combined environment. Mainframes drive the big enterprises who in turn drive the smaller ones that our societies are built upon. If QC could aid these big enterprises they would in turn let this information flow through into the lower sectors of our global economy. All this is why the paper will try and expose why we should start keeping track of QC in the very same everyone suddenly started keeping track of data research with classical computing. (Arute et al., 2019) (Amico et al., 2019)

1.1 Problem Statement

What can QC actually solve as of this very moment, is a question every interested enterprise is trying to figure out first. The field has shown even in this very early state with the limited amount of computational resources much promise. Inside the fields of data processing there is a clear trend that as we further develop quantum computational expertise the possible business impacts are generated exponentially. Many big enterprises have finally figured out the most lucrative and easy-to-apply ways of capturing important data that could have business value. Now the actual issue that most definitely is worth addressing, is that the processing of the shear amount of data has become unbearable in realistic time schemes.(Rieffel & Polak, 1998)

Business needs all these data results as soon as possible to gain the edge over competitors and industry leaders. Quantum could in theory exponentially aid classical computers with the processing of this large amount of data. Mainframes especially are able to generate so much I/O with all sorts of data e.g. credit card spending, production analysis, transport optimization, that the mainframe together with QC could very well become the power couple of the 21st century. With platforms such as Qiskit everyone is able to contribute towards quantum research even within a mainframe minded environment. (Abraham et al., 2019) and (McClean et al., 2017)

1.2 Research question

The question of "How will QC affect the mainframe environment and its applications?" can be a really useful question to solve because it would allow the highly expertised environment of the mainframe to be able to think of possible applications of the quantum research with their mainframe systems. In this moment of time practical QC has come such a long way that this question could possibly give added business value to the mainframe industry and all of their users. Exploring this domain can provide valuable insights in all different kinds of sectors that make use of the mainframe's high-data capabilities.

1.3 Research objective

The paper is designed to allow individuals that are interested in QC and general computer science to get a better grasp on the real business impacts of the quantum realm. We expect that we can evaluate the real value of QC inside a business, but also we want to find actual value and applications inside the sector as a whole. Following up with the real practical showcase of these new quantum technologies using a Python framework that is freely available for anyone to download and use. The framework is called Qiskit and is at the moment of writing the only framework that allows the connection to real quantum devices. This connection is free of charge, where you are able to create quantum circuits and test them out on simulators of the framework itself to then follow these circuits up by sending them off to a quantum device and get back the results from it. By being able to look at real executions you instantaneously receive a better grasp of the whole aspect of certain quantum phenomena.

1.4 Structure of this bachelor thesis

The paper will consist of the following chapters:

In chapter 2, we will introduce the very basic usage of QC to make sure every reader is able to understand the basic necessary principles to understand this paper.

In chapter 3, we will expose how classical and QC could offer a valuable partnership in their effort to speed up all research.

In chapter 4, we will finally show the real usages of quantum algorithms of the future through simulations or even executions on real devices with Qiskit. The algorithm that we will use will be an adaptation of Grover's algorithm for a unstructured search, more specifically an algorithm that is able to solve the 3-SAT problem in an quantum manner.

In chapter 5, there will be a discussion where we take in the results of the practical compartment of this paper. Furthermore we would still like to take a critical look at how QC has its benefits but also its disadvantages.

2. Quantum Essentials

To make sure everyone starts off on the same baseline to understand the full potential of this paper, we will introduce a few of the basic quantum principles. This paper is not targeting these specific principles but it they are essential to be able to learn further about quantum computing and its potential. If there is any specific interest regarding these ground fundamentals, we would refer you to the following papers, Rieffel and Polak (1998) and Shor (2000).

2.1 The Qubit and its representations

The foundation of any quantum related paper is and will always be the **qubit**. A qubit is just like a classical computing **bit** the foundational unit of the quantum computer. Whilst a bit can either be on or off, a qubit has a certain statistical measurement to it. This statistical aspect is derived from the way we interact with a measurement of a qubit. Now keeping this in mind, to be able to program on a quantum computer you need to be able to encode classical problems in a quantum manner, which as it turns out is not an easy task.

A qubit is not an infinite on or off, meaning that any type of computation needs to happen during the time frame that qubit remains stable without being thrown of its state by decoherence (2.0.3) or any other external factors. In other words defining a qubit as being on or off is simply not enough, if we want to keep the time-factor in the equation. One of the biggest issues with qubits is that they can behave unstable when influenced by the slightest of external influences, which also means the influence of an external observer. Once a qubit reaches an unstable state it has lost its quantum advantages and becomes a determined particle, which is not available for calculations any more. Meaning that during the execution of your program you are simply not able to look at the intermediary results as this would affect the final result, which would make the whole computation worthless.

Now more defined to comprehend the nature of a qubit we need to understand that representing a qubit is only possible in a complex field, which visualises a certain amplitude of the state of the qubit at a certain point in time. When we think about a classical bit, we like to imagine a switch being turned on or off. With a qubit you would have to think about a sphere that is being transformed and shifted around its axes to transform its state. *Felix Bloch* was the individual that came up with the Bloch sphere that we currently use to clearly represent what state a qubit is in at a certain point in time. Looking at figure 2.1 you are able to see a representation of the state of a qubit in its elevated state. You need to think about this representation in a complex field on which transformations and rotations can be applied. During runtime a qubit is in a probabilistic state where it can not be fully determined in which state the qubit currently resides, which makes the whole execution harder. The moment we measure the state of the qubit it will degrade back to a determined result that shows us exactly the state of the qubit, but in turn loses the quantum potential initially gained.

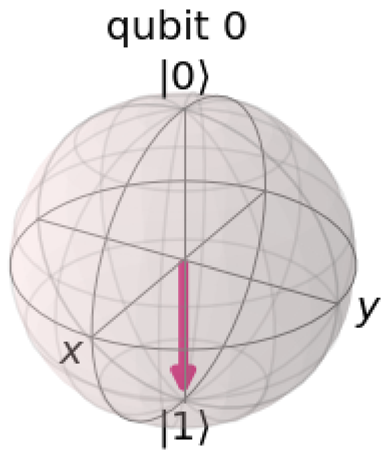

Figure 2.1: A Bloch sphere representation of a qubit in the elevated $|1\rangle$ state. The Bloch sphere clearly indicates that the state of a qubit has a complex aspect to it. By representing it in a sphere, we are able to showcase how the direction of the arrow has a probability with it. In this case the arrow has a probability of 1.00 for the elevated $|1\rangle$ state.

Another way of demystifying what a qubit exactly means is by representing it through the use of matrices and the Bra-ket notation. By using these matrices and using matrix transformations we can more easily expose the way a qubit can be influenced during execution. If you take a look at the way of representing a qubit in its elevated $|1\rangle$ state or ground $|0\rangle$ state in the formulae below, we are able to look at qubits in a more mathematical manner. Also by representing the qubits as a matrix we are able to more clearly show how operations on a quantum device have an impact on the state of the qubit.

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \, and \, |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

This image is mostly preferred by computer scientist because it gives them a clear image of the transformations, much in the same way an ordinary logic gate can influence an electrical signal. The combination of qubit basis states can be achieved by the utilisation of a Kronecker product. In the formulae below you are able to see how a 2-qubit system is represented through their matrix-representation. So look at the following transformations in much the same way one would look at an electrical signal would flow through a set of gates.

$$|00\rangle = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \quad |01\rangle = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$$

$$|10\rangle = \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \quad |11\rangle = \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

To sum it up a bit can be only be in a state of on or off and this can be checked throughout execution, whilst a qubit is in a uncertain state during execution much like Schrödinger's cat. So once we observe the qubit it becomes just as determined as a normal bit would be. But determining the state during execution will affect the rest of the experiment and will remove the potential benefits of quantum in much the same way if Schrödinger went on with the experiment after observation that the cat died, he would be certain that cat would still be dead at a later point.

2.2 Superposition and entanglement

Now the next 2 principles are fully responsible of giving quantum computing its exponential speed-up compared to classical computing in certain tasks like factorisation and database searches. However these principles only offer that powerful advantage when they operate together in solving a certain quantum algorithm.

2.2.1 Superposition

Superposition is a term a lot of people have heard about and how it could achieve major breakthroughs in the scientific world, but what it exactly represents is the real question.

We should think about qubits in superposition in a statistical way to receive a clearer image. When a qubit is put into a state of superposition, the qubit operates between its elevated state $|1\rangle$ and its ground state $|0\rangle$. By referring back to the representation of the Bloch sphere the qubit lives in a three dimensional complex field, where the only knowledge we can keep track of is its statistical chance of the state of the qubit. Until we have truly observed the qubit and have measured it extensively, the state of a qubit remains a statistical probability.

This all meaning that quantum particle in superposition can remain in both states at once statistically whilst it has yet not been observed. To explain this more clearly from a computer science perspective, a qubit in superposition is during execution behaving as 0 and 1 at the same time. A concept that seems impossible within a classical frame of mind but a concept that can also be very advantageous. E.g. if you are processing a big array of data through your classical processor, your processor will take one item of the array, process, convert and output it before it will take another item of the array and perform the same thing. A quantum processor could go about this process in a similar yet much more ingenious way. It would put an amount of qubits in superposition to represent the full array as input, perform the needed amount of quantum gates and receive the output in a single go instead of needing to loop over the full array. (Draper, 2000) And yes using a graphical processing unit does allow classical devices to perform in a likewise parallel matter, but it still does not do this through 1 computational cycle because you still need to combine the results at the end.

2.2.2 Entanglement

Entanglement is another interesting principle within the realm of quantum physics. This principle truly offers the quantum advantage that is predicted over every single sector. It refers to the correlation between entangled qubits where the state of one qubit influences the state of the correlated qubit in a way that it can be exploited and theoretically exponentially speed up computation. This entanglement can be achieved inside a quantum computer by the use of quantum gates on qubits in a state of superposition. You are able to put every single qubit available in a entangled state with the qubits available in the system.

E.g. When we have a 3-qubit system, we are able to entangle every single qubit with each other and put them in a superposition state as to be able to play around with the quantum capabilities of our system. Now let us add a single qubit to this system, by adding this single qubit we are able to entangle this qubit with the 3 other qubits. So this addition of the qubit has added the qubit itself as a new possible data-point into the system but it has also added these 3 new states of entanglement which in itself are new data-points.

This phenomenon is the reason why a quantum computer has an exponential factor to any added qubits, it becomes more clear where the quantum advantage can truly be gained. Such a quantum system can represent an immense amount of classical bits by using this phenomenon. The deterministic result of the computation with the qubits at the end of an experiment will always reflect the correlations of the qubits in the results. Furthermore we must keep in mind that enough experiments are to be performed in order to defend against the mistakes of external influences on a quantum system. (Brandao et al., 2016)

2.2.3 Quantum advantage

These two principles are constantly being used by a quantum processor to achieve a quantum advantage as the one that Google showed of in their latest showcase of their 'quantum supremacy', Arute et al. (2019). Superposition and entanglement are able to exponentially increase the computing power of a quantum computer, as you add more and more qubits you are exponentially increasing the available data items such a processor could handle.

E.g. to be able to simulate the biggest medicine of the 20th century, penicillin, you would need 286 functional qubits, which in turn would be able to generate the 2^{286} bits of memory. It straight up would be impossible to gather enough RAM in the world to be able to simulate this drug exactly. Now this sounds amazing if we were able to simulate any type of drug exactly, but as of now quantum computers are not capable of holding this amount of stable and reliable qubits. So actually getting to such a stable amount of qubits in itself would need a new breakthrough in the way we can add stable qubits to a system. The whole limitation of 'just adding qubits' to the system runs into a barrier by a term called Quantum decoherence.

2.3 Quantum decoherence

QC is not solely composed of benefits, the biggest downside is that as of now technology has not yet progressed far enough to actually provide the necessary amount of stable qubits to perform trustworthy calculations with. For example the simulations of penicillin, a system would need 286 qubits that remain stable for a prolonged period. But as of now Google has only been able to keep 53 qubits stable for a prolonged period of time using quantum error correction throughout the calculations. The loss of these quantum aspects during execution is called *quantum decoherence*, it is the phenomenon that describes how a qubit falls in an unstable state after being influenced by external forces or even internal influences from the qubits around it inside the system.

Referring to figures 2.2 and 2.3 below you are able to see that quantum decoherence is even a measurable phenomenon. The circuit below is especially built to show that quantum decoherence even shows up in the smallest of computations. In figure 2.2 a qubit gets thrown in an elevated state of $|1\rangle$ and then gets measured and pictured on a classical bit. Looking at the results you are able to see that quantum decoherence has occurred and de-elevated the state of the qubit back to $|0\rangle$ in 8% of the cases. In this experiment, were we have taken 1024 shots performing the circuit on the real quantum device, around 8 percent or 81 of the shots, were influenced to its ground state due to the quantum decoherence. For the code behind this figure, we would refer to appendix A.

Taking all of this into account the whole field has one giant, non-circumventable downside that goes with it. The larger our quantum systems become, the more internal decoherence we receive from the higher concentration of qubits near each other. This all could mean that there is a limit to how big we are able to make quantum machines, because at a certain point, without proper error correction, the internal decoherence would make every single calculation useless because of the high probability of faulty data throughout. However if we are able to find a solution to this internal decoherence, the amount of qubits inside a system could be limitless and our data processing with it could also become limitless. (Hartnett, 2019)

It is of utmost importance that we add qubits in a controlled manner where we are able to perform better and better error-correction before we just add more qubits into the system. If everyone started adding qubits to their systems without performing any research regarding quantum error correction, we would only receive an increase in quantity of computing power not in its reliability. And in this case quality is a way more important factor compared to quantity.

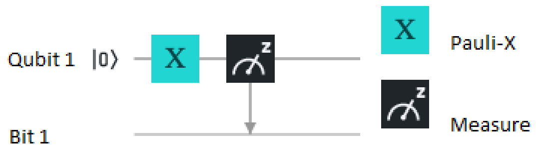

Figure 2.2: This is the quantum circuit that puts a qubit in an elevated state using the Pauli-X gate followed up with a measurement, to show off how quantum decoherence influences the results.

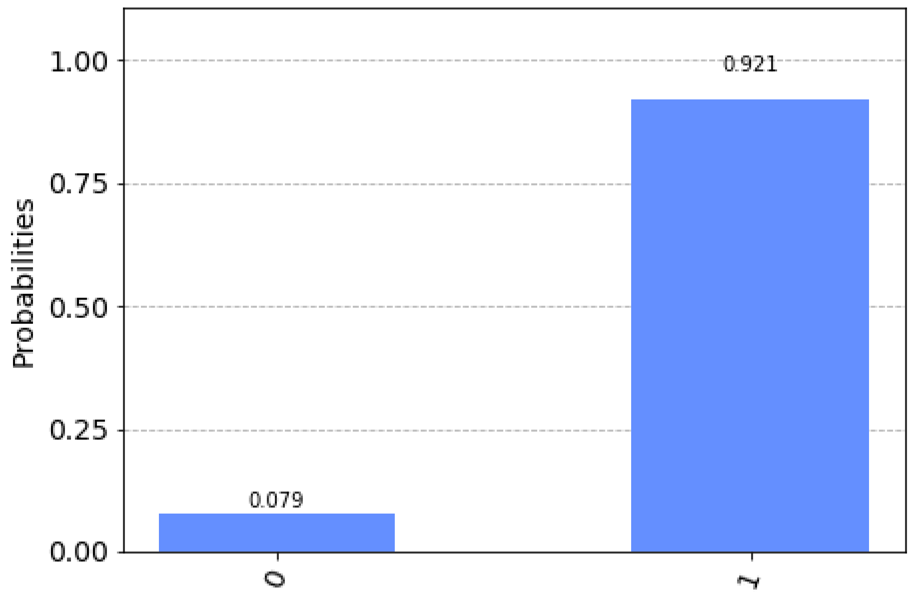

Figure 2.3: These are the percentages after performing the circuit on a real device from which it is visible that quantum decoherence has taken place on the initial $|1\rangle$ state to the $|0\rangle$ state.

2.4 Quantum gates

Now one might wonder, how do we create calculations with particles that are not observable and not tangible at a specific point in time. Quantum gates offer the solution to this question, a quantum gate affects one or more qubits during execution so that a programmer is able to perform changes to the state of the qubit but does not necessarily create an unstable qubit. From a programmers perspective they function in a similar way that a normal logic gate functions on an electrical signal inside a regular processor.

A major rule that a quantum computer needs to adhere to when it comes to its gates, is that quantum computations have to be 100 percent reversible whilst a classical computer does not have to deal with this limitation. This is clearly shown when we use the matrix representations of a qubit that passes through a specific quantum gate. Because by the very nature of quantum mechanics operations on these matrices are unitary and by that also reversible.

E.g. think of an OR-gate where you are able to only see the outcome of the signal with an on or an off signal. Now you can not know just from the outcome alone which initial signal influenced the OR-gate to be activated. This shows that a classical device is not bound by this gate-reversibility limitation. Now you can look at these gates in two different perspectives. First in the three dimensional perspective, we could apply a rotation over the z-axis of +90 to the qubit, now by the very nature of a rotation we could apply the reverse of this previous operation and receive the same state of the qubit initially. Secondly we have the option of representing these qubits by the Bra-ket notation and its accompanied matrix notation. A quantum gate would apply its unitary transformation mathematically to the state of a qubit. And by the very definition of an Unitary transformation, they are reversible. Both of these perspectives are viable to explain quantum aspects.

To understand the application of quantum gates during runtime on a quantum computer, we would like to refer solely back to the representation of matrices where unitary transformations are applied by performing a Kronecker product on the qubit in question.

2.4.1 Hadamard gate

The Hadamard gate is the single most important gate for creating a quantum computation. This gate is responsible for putting a qubit inside a state of superposition and is also is the one to get it out of this state. So in turn without this gate, quantum advantage would simply not exist. It maps $|0\rangle$ to $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|1\rangle$ to $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$, which are both superposition states.

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

2.4.2 Pauli-X gate

Performs in a similar way a classical NOT-gate performs on an electrical signal or an absence of it. A qubit in $|0\rangle$ state going through a Paul-X gate would go in a $|1\rangle$ state. We also showcase an example on how we apply these unitary transformations when the qubit would pass through a gate of this type.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$PauliX * |0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

$$PauliX * |1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$$

2.4.3 CNOT gate

This gate works as a flip for a qubit. It is a gate that connects two qubits together, where one is the control qubit and the other the target. If the control qubit is in an activated state it will flip the target qubit.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

2.4.4 Toffoli, CCNOT gate

The Toffoli gate works in the same way as a CNOT gate, but instead it has 2 control qubits. So both of them need to be activated to actually flip the target qubit, which logically requires at least 3 qubits in your system. This gate is the connector of the whole quantum system where we are able to entangle states in superposition.

$$CCNOT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

This is just a listing of the most prevalent quantum gates. These quantum gates form the basis for performing the simplest of quantum algorithms. These can be implemented through a programming interface or through many of the UI-editors where you can add gates along the circuit in a dynamic way. You are even able to send off these generated circuits through the UI tools to any real devices or simulators to analyse and compare your results.

3. Real-world solutions with Quantum

Once you start looking into quantum theory and everything it could possibly encompass for your scientific projects, you would find yourself in one of the deepest rabbit holes you could ever possibly find. The true value of quantum research is how we can actually use it for real-life solutions. One could easily imagine that being able to simulate an exact medicine within a couple of days instead of the many months it takes at the moment, would save a numerous amount of lives. So keeping this same train of thought throughout, it is of great importance that we actually focus our attention on what current developments could possibly mean for existing projects and research.

3.0.1 Quantum computing and traditional computing

QC is and will never be the sole solution to a problem. This new form of computing is made to be an addition to points where classical computing fails, e.g. searching through an extremely large dataset without having a clear index within a polynomial time frame such as in Terhal and Smolin (1998). QC also has its limits as it takes a lot longer to actually set up your computation than it would take on a regular machine. But it could be able to solve a couple of non polynomial problems we are currently facing in computer science like factorization. Some problems have been left NP-complete even with quantum attempts like this paper has tried, Wang et al. (2007). Quantum computing must not be looked at as the single solution for every problem, it still needs the help of classical computing to be able to perform its more advantageous tasks.

Classical computing is still great at organising and shuffling data around and performing parallel actions on your device, but with the help of QC we would be able to shift the heavy long term calculations over to devices especially made for long term and hard calculations like a quantum processor. Calculating a machine learning model Schuld et al. (2014) or performing an accurate simulation of a new medicine could be exponentially reduced in time, which would in turn return the value of these calculations to the business side in a much faster way. (Schuld et al., 2015) (Troyer & Wiese, 2005)

3.0.2 Quantum computing and the mainframe

First of all we need to clarify what a mainframe is and what its main use is in our current business environments. A mainframe is a type of supercomputer that is different from other supercomputers because it is not specialised in solving 1 really hard problem, like simulations or factorisation, it is specialised to have the highest possible amount of secure throughput for smaller calculations. The mainframe is widely used within the banking, production and logistical sector as it offers the most reliable way of managing your data that is generated by a certain business practice.

To clarify let us look at an example where a mainframe computer like the IBM Z15 shines. When millions of users throughout the world want to buy their flight tickets towards France around the end of April, a huge bottleneck is created at the end point of the booking system of the particular airport. A mainframe handles these types of atomic transactions quickly to make sure every single booking will come through with the correct data. If the data were to be corrupted along the way, the mainframe would be able to spot out these irregularities and discard this data so that the user receives a proper notification as soon as possible. So look at a mainframe computer as a really good processor of input and output of small tasks.

To quickly compare a normal super computer is used because of its high-speed processors that are able to create simulations, ML-models... They are designed to perform one synchronous task really well and if necessary brute force such a task.

IBM has released the new mainframe Z15 in 2019, with a broad future perspective, because as one of the top researchers in quantum technology they have a clear image of how a quantum computer could influence themselves and others within their sector. They are emphasising on 2 very different aspects to make sure their devices are the most likely to take the biggest market share, modernisation and security. With modernisation IBM is trying the incorporate the mainframe in as much areas as possible to keep on attracting new developers so that their devices don't fall behind. With this modernisation a lot of opportunities are opening up to connect widely different departments such as Machine learning and quantum research.

They have also emphasized on creating new security measures which focusses more on digital signing compared to the current RSA factorisation algorithm, this could secure the mainframe security status indefinitely. Quantum would in the future indeed be able to brute force these RSA based algorithms (Shor (2000)) and that is why data-security has become such a high importance area at the moment for everyone in computer science that knows the potential hazard of powerful quantum devices.

So now that you are able to view what role the mainframe plays, we can more clearly look at how quantum computers could offer major benefits as a complementary service for solving the harder problems just like a super computer works with the mainframe in much the same way. Nowadays all the data generated from the billions of transactions from the mainframe are preserved so that afterwards a supercomputer would be able to process all this information inside a reasonable time frame to get as most as possible business value out of it. If the quantum computer would be able to help process this data exponentially faster, the business value of this data would also exponentially increase. Mainframes offer this great amount of throughput of data that could offer this exponential force of QC the data-items it needs to become viable for business. Again this is something that could show how well QC can fit in our existing model of computation, to further enhance the processes that drive already our economies. QC will become the most helpful ally to make sure all our classical systems become even more valuable.

Mainframes are here to stay, with the dawn of fully data-driven worlds that operate on an autonomous platform and if QC could add to such an industry important device it will most definitely boost both devices in exposure and value. But for now there most certainly are questions of when QC would be able to reliably process and/ or handle the high throughput of a mainframe in a manner that would add to its business value. This will form another benchmark for QC where full cooperation of these devices can transform into something practical and valuable.

3.0.3 Quantum computing and Machine Learning

Another area where QC could have a major impact is the area of Machine Learning **(ML)**. At this moment machine learning is running into a bottleneck where the amount of data has become so intense that ordinary classical computers are not able to process the data in time so that its value can be exploited to its maximum potential. QC could help with this issue in a couple of major aspects, like data model training and data capturing. This would greatly improve the impact of ML on the business side, because the relay of the captured information through the models could indeed be shortened in exponential ways. (Biamonte et al.,)

At this moment research is becoming quite prevalent in ML with a combination of QC-technology. Qiskit has also seen this opportunity opening up and they too try and attract businesses with these advantages. We are able to enhance supervised learning algorithms as well as unsupervised learning, with time series or without. Algorithms such as linear regression, k-means clustering and even neural networks can be enhanced during its training phases with QC. Because due to superposition and entanglement, these algorithms could train a model theoretically through one loop instead of having multiple epochs that contain a certain batch size, which obviously speeds up the models that require a large amount of data to become valuable.

The utilisation of QC with ML would not aid the accuracy of ML in the short run because of the uncertainty of solving the quantum decoherence issue for now. But the time frame of processing a complete machine learning model could be exponentially decreased.

4. Practical demonstration with Qiskit

For two decades now people have been receiving fully blown quantum mechanics courses where they are able to experiment with the thought of quantum experiments in a theoretical type of way, but never were truly interested parties able to perform their experiments in a free and fluid manner. QC is at a point where we are able to effectively experiment with the technology as a broader community. Platforms like Qiskit are excellent in their reach towards interested parties and are more than welcoming towards new developments that could aid the whole community in its research and adaption. The service is open source which truly pushes the whole movement of research out of this shroud of high costs and large enterprises. This will obviously influence other branches to follow in the same footsteps as to allow every party that is interested or has a passion to be able to participate in a costless and open manner. To remain objective and fair towards other companies outside of IBM, Google is also participating in the open source community with platforms like Cirq, McClean et al. (2017).

In the following part, we will lay out how any interested parties are able to perform their own executions on real devices and start applying what some of them have been learning theoretically for over 20 years.

4.0.1 Grover's search algorithm in a practical fashion

Grover's search unstructured database search

We have chosen for an adaptation of an algorithm that could prove extremely useful for any implementation together with a mainframe, which is the Grover Search algorithm applied for the boolean satisfiability problem. The whole premise of the original algorithm is that

$f(x,y,z) = (\neg x \lor y \lor \neg z) \land (x \lor \neg y \lor \neg z) \land (x \lor \neg y \lor z) \land (\neg x \lor \neg y \lor z) \land (x \lor y \lor z)$

Figure 4.1: The 3-SAT problem that we have processed manually, using the quantum simulator and using the real quantum device.

we are able to speed up the search time in an unstructured database. This all meaning when a computer needs to find an item with an unique attribute that differentiates itself from the other items in the list, QC could become the main solution to a brute force task like that. The whole algorithm uses something called "amplitude amplification" where the algorithm influences the probabilities in such a manner that the specific item has the highest probability after the quantum computation. (Grover, 1996)

To clarify amplitude amplification further, this is a principle that affects qubits in their superposition and entangled states where they would interfere with each other to eliminate the least likely outcomes and amplify the statistical chance of the desired item. By applying this to an algorithm after the correct transformations, you would be able to decompose all the different results back to a highly probable result that does not collapse all the qubits in superposition like any normal measurement would.

Grover's search algorithm in an applied form

For the experiment itself, we have chosen for the specific "Boolean satisfiability problem" which uses Grover's way of amplitude amplification to find the correct results of a boolean problem. This computer science question goes as follows, given a boolean comparison of multiple parts are we able to determine the outcome of this function to where the result of the boolean calculation equal TRUE. Being able to solve this comparison in a single execution could abuse the fact of superposition and entanglement and could prove useful when we scale out the problem towards thousands or even millions of factors for other functions. For now the 3-SAT problem has been chosen to be performed using Qiskit to show off the performance of QC that is available to us.

You are able to view the boolean function in figure 4.1 as the problem that we will try and solve using Quantum technology. The algorithm now needs to find which solutions are possible by interchanging x,y,z with TRUE/FALSE.

Executing the quantum algorithm

Using a simulator of an ideal quantum computer we are able to show the results in figure 4.2. The probabilities have been amplified to where there are multiple results for this boolean expression. Figure 4.3 shows the gathered results when we sent off the identical circuit towards one of IBM's real quantum devices (IBMQMelbourne16 in this case).

The reason for choosing this specific experiment is to show that even problems that just require us to encode boolean statements we needed 694 quantum gates. IBMQ transpiles the sent-off circuit to the necessary amount of gates needed for this specific calculation. It does not keep in account that having this much gates on a single line of computation

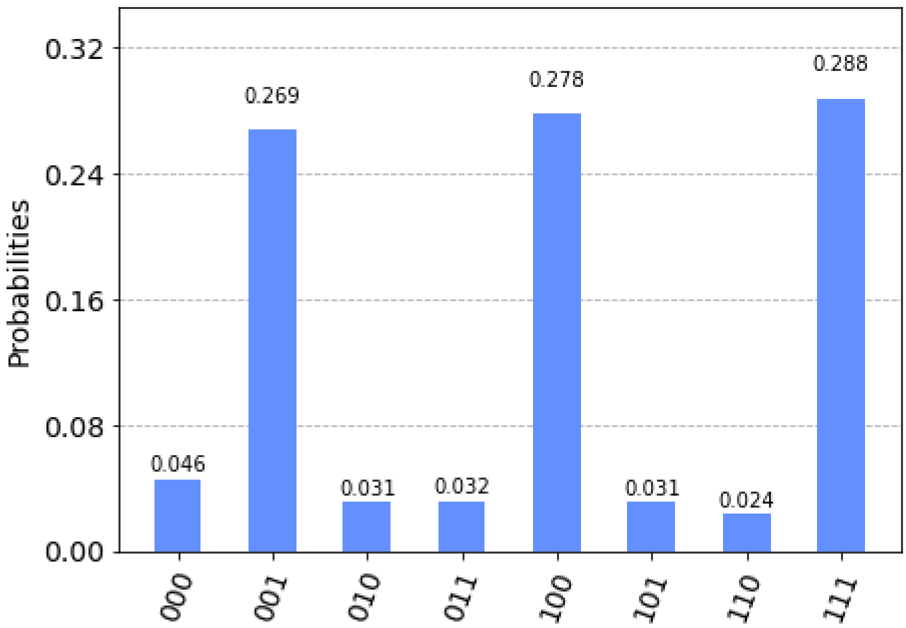

Figure 4.2: These are the results of executing the algorithm for the 3-SAT problem on a **quantum simulator** that comes with Qiskit. The encoding refers to the TRUE/FALSE value of the x,y,z respectively

invites a multitude of quantum decoherence issues during runtime. (Appendix A)

Comparing figures 4.2 and 4.3, you are able to see that decoherence for now breaks the probabilities of a computation too much to reliably trust any computation of this size out of a quantum computer. The values became distorted over time by all types of interference.

If we compare the manually gathered results from table 4.1 above across the simulated version and the real version, we are clearly able to see that decoherence has played too big of a role to be certain of any reliable output for these types of large computations. By having these big differences in results we once again prove that simply adding qubits will not be helpful for the reliability in QC. The results differ too much to ever use these results in business cases where speed is as critical as accuracy in reporting towards externals.

Let us work out a clear example (figure 4.4) to make sure our probabilities generated by the real quantum device are incorrect. If we take the highest probability of the real execution which is the configuration of 101. Meaning that the quantum computer determined that when X and Z are true the whole boolean expression will result in a returned value of TRUE. This is simply not a valid option for this boolean expression. If we look at the first part of this boolean expression we can see that this configuration would return a FALSE resulting in the whole expression being FALSE because all the parts are connected with a logical AND.

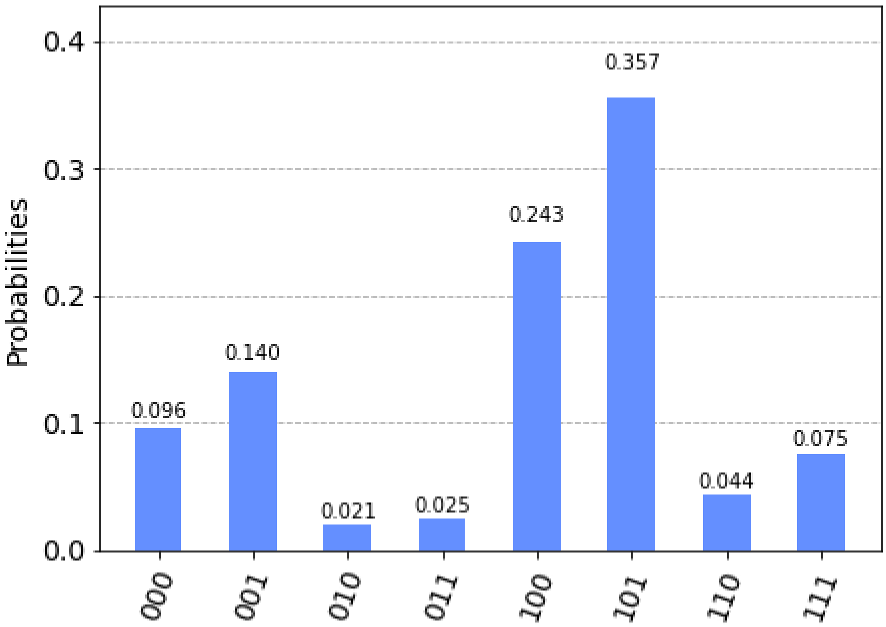

Figure 4.3: These are the results of executing the algorithm for the 3-SAT problem using 15 qubits on a **real quantum device** that comes with Qiskit. The encoding refers to the TRUE/FALSE value of the x,y,z respectively

$$f(1,0,1) = (0 \vee 0 \vee 0 \vee) \wedge (1 \vee 1 \vee 0) \wedge (1 \vee 1 \vee 1) \wedge (0 \vee 1 \vee 1) \wedge (1 \vee 0 \vee 1)$$

Figure 4.4: The filled-in example for showing how performing this boolean equation on the real device has been influenced too much by quantum decoherence.

4.0.2 Data-encoding in QC

As the experiment shows we are able to use quantum algorithms to solve classical computer problems. Of course quantum decoherence has played a great part in the computation and solving it would be greatly beneficial for the whole sector, but there is also a different problem that arises with defining our classical way of problems in a quantum way. The way we represent data in a quantum circuit quickly becomes overly complicated for any large database structure. This is visible from appendix A, where you are able to see how many gates were needed to perform this rather small boolean equation. (694 gates)

So for now quantum computers are not as powerful as classical machines in the way they are able to solve classical problems. But they can have value in the sectors that are the most halted by the use of classical devices. E.g. using them for use-cases like simulating quantum effects we can easily imagine that using a qubit to actually simulate quantum effects is way more effective than simulating a qubit using classical bits.

The issue lies in the fact that we want to input classical database structures/ data-items into a quantum device and hopefully receive the results in a readable classical solution. So if we

$f(x,y,z)$	TRUE/FALSE
0 0 0	FALSE
0 0 1	TRUE
0 1 0	FALSE
0 1 1	FALSE
1 0 0	TRUE
1 0 1	FALSE
1 1 0	FALSE
1 1 1	TRUE

Table 4.1: These are the results of our 3-SAT problem, calculated in a manual classical manner.

want to solve existing classical problems we need to find a performant manner of encoding these problems in a quantum device. As appendix A shows, we would need 694 quantum gates to encode and solve this small boolean problem. The generation and application of these gates itself bring a lot of potential decoherence and risk in the reliability of the results, which in turn downgrade the experience of working with the quantum devices as of now.

All of this shows an issue we are facing with the encoding of our classical data to quantum data and back. For the proof-of-concept experiments it does not matter as the encoding time does not influence the runtime of the experiment as a whole. On a normal classical machine, it took 2 seconds to solve this boolean expression and on the quantum device the execution was 7.3 seconds. But once we start scaling out the issue where we would want to find a specific item through the use of Grover's algorithm, we would run into the issue that the encoding and decoding of the input and output could take up a great amount of computational time. If however QC develops in such a way that we are able to gain the full benefits of qubits in superposition, this encoding time could be overcome. But for now it remains a crucial factor in solving the whole feasibility of QC.

4.0.3 Mainframe computing with QC

As the paper has previously stated having QC together with the power of a mainframe could become extremely advantageous for the whole industry to provide the power of data crunching this immense layer of internal data that companies have collected over the years with their mainframes. So we needed to find a circuit that could show off where quantum computing indeed could benefit in the crunching of data in a better/ faster way than classical computing can at the moment. Soon it became clear that simulating anything of a mainframe is impossible for now, we can simulate how a new form of database search could work with Grover. But we are not able to simulate the main advantage of a mainframe device, which is performing quick, stable and secure input and output transformations. And as shown by the experiment it is obvious that having a stable output of a specific input is not one of the main strengths of QC for now. Then when we take into account the encoding and decoding of classical computations and problems it only decreased the feasibility for running mainframe-sized data-sets.

So for now there is no clear advantage when we would use the current developments of QC with the existing mainframe technology, because as stated above we ran into the issue of data-encoding. The potential for QC and mainframe to still become the data-crunching powerhouse of the future is still there. But for this potential to become viable, a couple of major hurdles have to be overcome first.

4.0.4 Future prospects

As for now we are able to play around with the greater problems of quantum computing but to be able to *reliably* solve real-world solutions in a beneficial way remains an uncertainty.

With the current state of engineering, computer scientists will have to wait to fully utilise the system in a reliable fashion. But as engineering develops the power of quantum computing will increase exponentially as stated by Neven's law (Hartnett, 2019) with each added qubit to the system, which would make algorithms like this extremely valuable for data-crunching. When we find a way to circumvent the interference of quantum decoherence and the data-encoding issue, a quantum system could become an essential tool for every sector willing to innovate.

5. Discussion

Quantum computing will most definitely become one of the great buzzwords of the next decade. With the release of Google's paper Arute et al. (2019) around their interpretation of 'Quantum supremacy', the whole field was catapulted to the forefront of research. All the big players in Quantum research have been given a tremendous spotlight for the future of profitable quantum computing. And this is precisely why we need to make the whole concept more approachable for anybody interested.

5.0.1 Quantum computing for now

From the research around the subject, there are a couple of interesting conclusions to draw. Beginning with one of the most important ones, Quantum computing is here to stay. The technology has proven too much promise in too many sectors that have a strong financial backbone. At this point it is important to understand that the world around us that is visible with the naked eye is completely homogeneous with the 'quantum world', meaning that all research to exploring our world around is can only be profitable towards the future.

To return back to the more concrete conclusions of this paper. While executing this specific version of Grover's algorithm, there was a clear trend visible between the theory and the practical example that does show there are some growing pains that come with the expansion of our quantum devices. Yes, the theory can be fully implemented at this point to visualise its results when we would work in the most perfect of environments like a simulator. But this perfect environment simply does not exist in the real world or at least for now it doesn't. Meaning that we will have to be creative to reach this edge of perfect conditions to get to a point that these simulated algorithms can become reliable and profitable towards the future. There are two ways of trying to fix the issue of quantum decoherence. The first would be to actually create devices that are not influenced by any internal/ external interferences that could provide a stable platform for algorithms like Shor's encryption breaking algorithm Gidney and Ekerå (2019). The other option would be to account for these interferences to happen anyway and try and compensate against them in a software way, much in the same way a computer does error correction for downloaded files. The latter seems to be the more reasonable option where we are already trying to implement these quantum error correction in to provide better results (Cory et al., 1998). For now there is no clear technique behind the whole principle except to play around with the length that a qubit needs to stay in its elevated $|1\rangle$ state and the length of the circuit as a whole

But this all can be tried out, because at the moment of writing we are able to extensively experiment with real or simulated quantum devices. There is a multitude of platforms available, where most of them are open-source and you are free to contribute to them as you want to expand their feature-set. Frameworks like Qiskit allow you to design quantum circuits and test them using their built in simulators, which are easy to pick up and hard to master. As of now IBMQ is the only service that allows you to push up your circuits to really test them on a real device owned and managed by IBM. By granting people the privilege to pay around with the real devices and notice the shortcomings through raw data results, really shows off how dedicated the whole community of computer science is on pushing this technology to the forefront.

5.0.2 Quantum computing and its myths

As shown in the experiment, QC will not break our entire world in the next year in any drastic manner. So theories that quantum computing could break our entire encryption standard in a matter of years seem absurd once you look at the real executions of these needed algorithms on real quantum devices. Nevertheless we do want to work proactively to have solutions ready-to-go once these machines do become powerful enough to brute force the RSA-encryption by finding the factors of the prime numbers that represent the private keys of encrypted files. There is also the notion that quantum decoherence will prove to have an insurmountable mountain the more qubits we start adding to the systems. This indeed is a major hurdle that needs to be overcome to make quantum computing the new standard for solving really hard to solve problems in the computer science community.

5.0.3 Quantum computing as an addition

The one thing we should take away from the dawn of quantum computing the following decade, is that quantum computing is not the one solution for every single issue in computing. It has its advantages and disadvantages just like classical computing. We should strive to make these two technologies as complementary as possible so that they can cancel out each others disadvantages and amplify their advantages respectively.

With this work, we have tried to inspire people to learn more about the subject of quantum computing and to hopefully entice them into writing their own 'Hello-world-Applications' on any of the freely available frameworks. For the actual executions and setup of the used experiments, we refer to appendix A.

5.0.4 Future works

As to conclude back with the main subject of this paper, the potential powerhouse of a mainframe device and a quantum computer. If IBM keeps up it doubling pace of releasing quantum computational resources into the world each year and the quantum error correction evolves to an acceptable percentage, we could start thinking about implementing QC machine learning in our mainframes or even the speeding up the database structures in the devices. But for now QC and mainframe are not able to cooperate in an useful manner to give back business value.

It is good to explore these options upfront so that we know when the time comes to expand QC towards the mainframe we have a clear scope to all the potential applications of these two devices.

Figure 5.1: The platform we used for executing the 3-SAT algorithm on a real quantum device. © Copyright 2020, Qiskit Development Team Last updated on 2020/05/14.

A. Using Qiskit yourself

For the practical execution, we preferred to use Anaconda 1.9.12 for all our experiments. By using an environment that was just created for using Qiskit, we could debug way easier as there only were a handful of addition needed.

You will need to pip install, Qiskit, matplotlib, jupyter and Pyscience to have an all-round work environment for creating your own circuits.

Qiskit.org will give a full explanation, if the basics would not suffice.

A.1 Setting up your Python-Qiskit environment

Make sure you are able to execute the commands as shown in figure A.1 to ensure that you are capable to create your own circuits.

```
In [1]: import qiskit
        qiskit.__version__

Out[1]: '0.14.1'
```

Figure A.1: If you are able to execute this command in your Python environment, you have installed Qiskit successfully.

```
In [ ]: from qiskit import IBMQ
        IBMQ.save_account('YOUR_API_KEY')
```

Figure A.2: The necessary code to connect your Python environment to the external quantum devices.

A.2 Connecting your Python environment with IBMQ

If you would like to use the actual devices to perform your tests, you need to create an account on . Then you need to find your private key and connect it to the environment in a Jupyter notebook as seen in figure A.2. This key will be used each time you send off a circuit to the IBMQ quantum devices, the use of these devices is *free of charge*.

A.3 Running the quantum decoherence example

This section will show the simple way that you can show quantum decoherence taken an effect on your circuits. We will put 1 qubit in an elevated $|1\rangle$ state followed up with a measurement and then send it off to a quantum device.

This are the needed imports for the basic example.

```
[frame=single]
from qiskit import(
QuantumCircuit,
execute,
IBMQ,
Aer)
import numpy as np
from qiskit.visualization import plot_histogram
```

Here we create a quantum register of 1 qubit and apply a Pauli-X gate, followed up by a measurement to a classical register.

```
[frame=single]
circuit = QuantumCircuit(1,1)
circuit.x(0)
circuit.measure(0,0)
```

Using matplotlib will result in a much cleaner output of the circuit.

```
[frame=single]
circuit.draw(output= "mpl")
```

These are the needed actions to actually set up your device. You will need to choose your ibmq-device on the website where you have signed up for an IBMQ-account.

```
[frame=single]
IBMQ.load_account()
prov = IBMQ.get_provider(hub='ibm-q')
device = prov.get_backend('ibmq_armonk')
```

Here we actually send off the circuit towards our device.

```
[frame=single]
job = execute(circuit, backend = device, shots = 1024)
from qiskit.tools.monitor import job_monitor
job_monitor(job)
result = job.result()
plot_histogram(result.get_counts())
```

A.4 Running the practical 3-SAT example

In this part, we will showcase the code of the practical portion of this paper where we solved a 3-SAT problem using Qiskit.

First we will perform the algorithm through the use of the simulator, then we will send the circuit off to an IBM device capable of handling the large circuit. (13 qubits needed at the least)

```
[frame=single]
import numpy as np
import matplotlib as plt
from qiskit import *
from qiskit import BasicAer
from qiskit.visualization import plot_histogram
%config InlineBackend.figure_format = 'svg'
from qiskit.aqua import QuantumInstance
from qiskit.aqua.algorithms import Grover
from qiskit.aqua.components.oracles import LogicalExpressionOracle, TruthTableOracl
```

This is the specific encoding that is used for encoding a Boolean satisfiability problem into the Grover search algorithm.

```
[frame=single]
```

```
input_3sat = '''
c example DIMACS-CNF 3-SAT
p cnf 3 5
-1 2 -3 0
1 -2 -3 0
1 -2 3 0
-1 -2 3 0
1 2 3 0
'''
```

```
[frame=single]
oracle = LogicalExpressionOracle(input_3sat)
```

Using the previously mentioned encoding and inputting this into Grover's search algorithm is really straightforward when we use the Qiskit-framework.

```
[frame=single]
grover = Grover(oracle)
```

Here we are setting up the simulator and retrieving the useful data from it.

```
[frame=single]
backend = BasicAer.get_backend('qasm_simulator')
quantum_instance = QuantumInstance(backend, shots=1024)
result = grover.run(quantum_instance)
print(result['result'])
```

We can easily plot this data to receive a clear image of the results after the simulation.

```
[frame=single]
plot_histogram(result['measurement'])
```

Now we will send off this circuit to an actual device. Make sure you pick an device from the IBMQ website that supports at the least 13 qubits.

```
from qiskit import IBMQ
IBMQ.load_account()
provider = IBMQ.get_provider(hub='ibm-q')
backend = provider.get_backend('ibmq_16_melbourne')
```

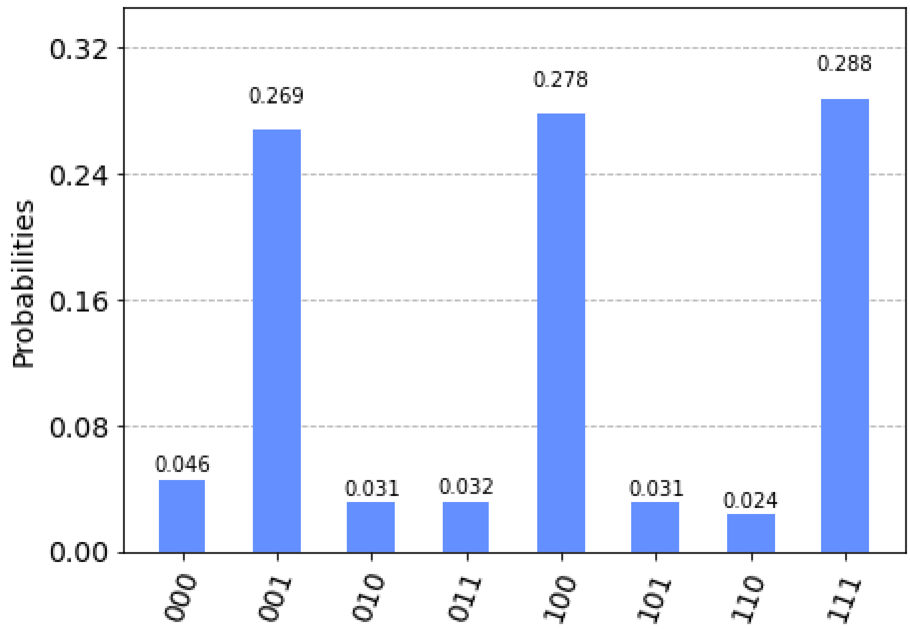

Figure A.3: The simulated results plotted directely in a Jupyter notebook.

Here we perform a manual transpile, which turns our algorithms in the needed gates to perform the Grover search algorithm for example. In this case, you are able to see the results of this transpile in text format under the code.

```
from qiskit.compiler import transpile


grover_compiled =transpile(result['circuit'],backend=backend,
optimization_level=3)

print('gates = ', grover_compiled.count_ops())
print('depth = ', grover_compiled.depth())
```

```
gates =  OrderedDict([('cx', 397), ('u2', 176), ('u1', 68), ('u3', 48),
('measure', 3), ('barrier', 2)])
depth =  431
```

Once the circuit has run successfully you are able to view the result on the website of ibmqexperience in nicely put together format as visible from figure.

```
job = execute(result['circuit'], backend = backend , shots = 1024)
from qiskit.tools.monitor import job_monitor
job_monitor(job)
```

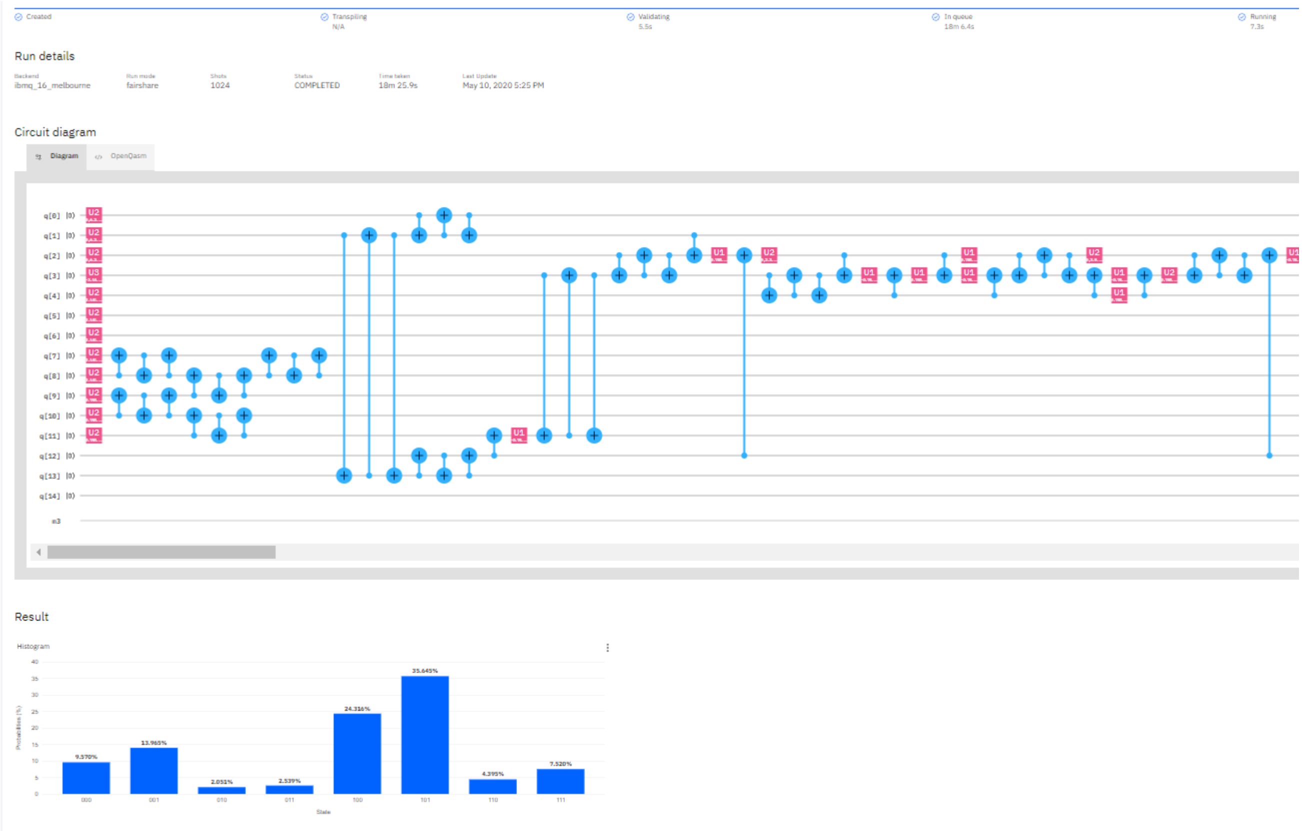

Figure A.4: This is the visual result of running the practical version of this paper and displaying it on the ibmq website.

Hopefully you will install Qiskit yourself and try these experiments out free-of-charge.

B. Research Proposition

Under this section you are able to view the original proposition for this paper to introduce the subject with schooling officials and technical promotors. This section can also serve as an introduction to this paper for any further interested readers.

To address the whole reason why this paper was created, the subject has become more and more influential in the Computer Science world. We have officially come at a point where we are able to think of real world utilisations of quantum computers to further our research in various subjects. Quantum has become a buzz word at this point, but not everyone that throws it around has a real grasp on what it exactly means. That is why this paper has been created to aid interested people in the subject to gain a real understanding of what quantum actually is and what it can do.

B.1 Introduction

B.1.1 Situating the subject

There has been a strong believe over the last 30 years that quantum computing can and will influence our sophisticated environment more than we think. In case of the mainframe environment it will maybe be the most influenced sector in *computer science*, because of its immense creation of data. Data will become or has already been the driving factor inside our societies, think of how much our daily lives are already controlled by data (e.g. online shopping, social media etc.). With the usage of mainframes we are able to create a sense of logic in this almost infinite pile of data. Now with *theoretical* utilisation of quantum computing, data can be searched more thoroughly and faster. (Grover, 1996)

If we are able to find and explore quantum applications for our current high-transactional business applications, a new wave of investment in research will open itself up. Which would obviously boost both fields at once. In this paper we will try and find these general applications that can prevail through the use of quantum technology.

B.2 State-of-the-art

B.2.1 Prior knowledge

Inside the paper a couple of physics specific terms will be utilised. If you are not familiar with basic quantum physics notations and or terms, it would be highly recommended to read one or both of the following papers, (Rieffel & Polak, 1998) or (Shor, 2000). For the general quantum notation that are used throughout the field, we refer towards Dirac (1939). It is also possible to read this paper as an informational piece without the implications of the mathematics and physics surrounding the subject. As previously stated the paper will not be going in depth technologically, because the scope is more focused on exposing the practical usages of quantum computing compared to classical computing or the combination of them both.

B.2.2 Recent developments

As of now Google has claimed to have won the *Quantum Supremacy race* (Arute et al., 2019) against IBM. They have realised this through the creation of their 54-Qubit quantum computer (53 functional qubits), that is able to perform a calculation exponentially faster than a classical system could ever hope to perform. In this case the *Sycamore* (Quantum processor) was able to perform a calculation within 200 seconds that could only be performed by a classical computer over 10.000 years (theoretically). Although it most definitely was an experimental calculation that has no real value in the business world, it does however prove the potential of quantum computing. It has been rumoured that IBM will release its counterpart of research in 2020. The fact that these 2 conglomerates are competing so fiercely will only further the technological developments in the realm of quantum mechanics. IBM has not been sitting idly either, they have released a paper regarding quantum algorithms applications. (Amico et al., 2019)

B.3 Methodology

While the field of practical quantum computing is still in its infancy, there are a lot of different possible angles to approach the subject with. First of all we will be introducing the guiding principles of quantum computing, as to all start on the same footing. Then we will explore the realistic potential that quantum computing can offer for economic gain, especially for mainframe development. This will mainly be comprised of an extensive literature study that will set its focus on economic applications of quantum computation and thereby on the mainframe environment. Concretely the paper will use real-life economical batch data and will process this data through the use of quantum algorithms and classical algorithms. If there are any advantages in processing the nightly batch load by using quantum algorithms, it will become provable that quantum computing can also be extremely profitable. There will also be demonstrations of quantum computation software such as Qiskit by IBM (Abraham et al., 2019), Cirq by Google (McClean et al., 2017) and Q Sharp by Microsoft (Svore et al., 2018). Qiskit stands out because it is an IBM Python framework that solely offers the opportunity to actually execute your quantum circuits on real quantum devices as of today. (with limited qubits however)

B.4 Expected results

The paper will try and create a more concrete point of view on the possible features quantum computation can offer. Through the analysis of multiple papers, we are hoping to find certain points of contest. These points indicate the highly debated subjects within quantum computing and are therefore extremely valuable. We will be trying to locate and display the business potential within these points of conflict. Currently IBM has created an extremely stable and performant business environment with its mainframe, Z15 and its older versions. Anything that can/ will affect this stable business platform can form a great threat or opportunity to the way we currently create and process our data. To protect this stable platform, we will be trying to index all the threats and opportunities that come with the introduction of quantum computation in our current computational environment. The second part of the paper will be more software-orientated, where we will be creating an application that processes the typical nightly batch data. This application will be performed on the different quantum platforms an on a classical device. The paper will visualise these probabilistic and timing differences between results of the different software platforms and will try to show attention points with simulating quantum computers compared to effectively executing on one. Through the demonstration of quantum computation we hope that readers are going to be personally inspired to be creative with the new technology and start developing their first 'Hello World' with their quantum circuits.

B.5 Expected conclusions

We are expecting to *debunk* the more absurd ideas of quantum computing. (e.g. destroying all our encryptions and our society) Concretely, we are going to put the whole subject inside a more realistic 'future' vision. This will hopefully offer readers ideas of possible applications of quantum computation inside their departments (e.g. Chemistry, Economics, Astronomy etc.) Also With software being so readily available for the general public, we expect that quantum computing applications will be created exponentially faster than with the dawn of classical computing 70 years ago. With this train of thought, we are hoping that real economical value can be available within the next decade. By processing our example night batch load we hope to find this necessary business value. Frameworks like Qiskit will be developed further and more powerful quantum computers will be made available towards the public to boost the research in the subject. And with these thoughts we can be certain that interest in quantum computers will only increase in the future.

Bibliography

Abraham, H., Akhalwaya, I. Y., Aleksandrowicz, G., Alexander, T., Alexandrowics, G., Arbel, E., Asfaw, A., Azaustre, C., Barkoutsos, P., Barron, G., Bello, L., Ben-Haim, Y., Bevenius, D., Bishop, L. S., Bosch, S., Bucher, D., CZ, Cabrera, F., Calpin, P., … yotamvakninibm. (2019). Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2562110

Amico, M., Saleem, Z. H., & Kumph, M. (2019). Experimental study of Shor's factoring algorithm using the IBM Q Experience. *Phys. Rev. A*, *100*, 012305. https://doi.org/10.1103/PhysRevA.100.012305

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G. S. L., Buell, D. A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., … Martinis, J. M. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, *574*(7779), 505–510. https://doi.org/10.1038/s41586-019-1666-5

Biamonte, P., Jacob andWittek, Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (). Quantum machine learning. *Nature*.

Brandao, F. G. S. L., Christandl, M., Harrow, A. W., & Walter, M. (2016). The Mathematics of Entanglement.

Cory, D. G., Price, M. D., Maas, W., Knill, E., Laflamme, R., Zurek, W. H., Havel, T. F., & Somaroo, S. S. (1998). Experimental Quantum Error Correction. *Phys. Rev. Lett.*, *81*, 2152–2155. https://doi.org/10.1103/PhysRevLett.81.2152

Dirac, P. A. M. (1939). A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, *35*(3), 416–418. https://doi.org/10.1017/S0305004100021162

Draper, T. G. (2000). Addition on a Quantum Computer.

Gidney, C., & Ekerå, M. (2019). How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits.

Grover, L. K. (1996). A Fast Quantum Mechanical Algorithm for Database Search, In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, Philadelphia, Pennsylvania, USA, ACM. https://doi.org/10.1145/237814.237866

Hartnett, K. (2019). A New "Law" Suggests Quantum Supremacy Could Happen This Year. *Scientific American*.

McClean, J. R., Sung, K. J., Kivlichan, I. D., Cao, Y., Dai, C., Fried, E. S., Gidney, C., Gimby, B., Gokhale, P., Häner, T., Hardikar, T., Havlíček, V., Higgott, O., Huang, C., Izaac, J., Jiang, Z., Liu, X., McArdle, S., Neeley, M., … Babbush, R. (2017). OpenFermion: The Electronic Structure Package for Quantum Computers.

Rieffel, E., & Polak. (1998). An Introduction to Quantum Computing for Non-Physicists. *ACM Computing Surveys*, *32*. https://doi.org/10.1145/367701.367709

Schuld, M., Sinayskiy, I., & Petruccione, F. (2014). Quantum Computing for Pattern Classification (D.-N. Pham & S.-B. Park, Eds.). In D.-N. Pham & S.-B. Park (Eds.), *PRICAI 2014: Trends in Artificial Intelligence*, Cham, Springer International Publishing.

Schuld, M., Sinayskiy, I., & Petruccione, F. (2015). An introduction to quantum machine learning. *Contemporary Physics*, *56*(2), https://doi.org/10.1080/00107514.2014.964942, 172–185. https://doi.org/10.1080/00107514.2014.964942

Shor, P. W. (2000). Introduction to Quantum Algorithms.

Svore, K., Roetteler, M., Geller, A., Troyer, M., Azariah, J., Granade, C., Heim, B., Kliuchnikov, V., Mykhailova, M., & Paz, A. (2018). QSharp. *Proceedings of the Real World Domain Specific Languages Workshop 2018 on - RWDSL2018*. https://doi.org/10.1145/3183895.3183901

Terhal, B. M., & Smolin, J. A. (1998). Single quantum querying of a database. *Phys. Rev. A*, *58*, 1822–1826. https://doi.org/10.1103/PhysRevA.58.1822

Troyer, M., & Wiese, U.-J. (2005). Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations. *Phys. Rev. Lett.*, *94*, 170201. https://doi.org/10.1103/PhysRevLett.94.170201

Wang, X.-Y., Feng, Y.-X., Huang, D.-B., Pu, W.-G., Zhou, Y.-C., Liang, C.-G., & null Zhou. (2007). Quantum swarm evolutionary algorithm, Quantum-inspired evolutionary algorithm, Particle swarm optimization, Knapsack problem, Traveling salesman problem. https://doi.org/10.1016/j.neucom.2006.10.001