Master's Programme in Data Science

Quantum computing and Grover's algorithm

Aleksi Suuronen

April 18, 2023

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Aleksi Suuronen

Työn nimi — Arbetets titel — Title

Quantum computing and Grover's algorithm

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Report	April 18, 2023	18

Tiivistelmä — Referat — Abstract

Grover's algorithm is a quantum algorithm, which can be used for searching unstructured databases quadratically faster on a quantum computer, than existing search algorithms on classical computers. Quantum algorithms are ran on quantum computers, which use a computing paradigm called quantum computing. Quantum computing takes advantage of quantum mechanical effects, such as quantum parallelism. This results in more computing power for certain computational problems, like integer factorization and database querying.

For this report, I have read 9 papers [1 – 9] about quantum search algorithms, such as Grover's algorithm, and about quantum computing and its relation to databases. During this report, I will tell about Grover's algorithm and also provide an introduction to the field of quantum computing. Furthermore, I have read a paper and done research on the internet, how to program quantum algorithms using an open-source framework for Python called Qiskit, and how to deploy and run them on a real quantum computer. In this report, I will compare running Grover's algorithm on IBM's cloud services on a quantum computer simulator versus on a real quantum computer on the same searching related task.

ACM Computing Classification System (CCS):
Hardware → Emerging technologies → Quantum technologies → Quantum computation
Theory of computation → Design and analysis of algorithms → Data structures design and analysis → Sorting and searching

Avainsanat — Nyckelord — Keywords

Grover's algorithm, Quantum computing, Quantum database search

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1. Introduction

Quantum computing is an emerging technology and computing paradigm, which combines quantum mechanics and computer science [1, 8]. Richard Feynman introduced the idea of quantum computing in the 1980's, by observing that quantum mechanical effects could not be efficiently simulated on a classical computer [7]. At first the scientific community were not interested in the field of quantum computing, as building quantum computers turned out to be difficult and there were no scientific proof that quantum computers would be much more faster and efficient than classical computers.

However, in 1994, Peter Shor introduced a quantum algorithm for efficient integer factorization called Shor's algorithm [7]. Shor's algorithm proved to run in polynomial time, when existing factorization algorithms ran in exponential time. This resulted in quantum computing gaining more interest. In 1996, Lov K. Grover introduced a quantum algorithm called Grover's algorithm for searching unstructured databases quadratically faster than known classical algorithms [4]. When the best search algorithm for classical computers need $O(N)$ queries in the worst case to find the right value from an unsorted database, Grover's algorithm needs only $O(\sqrt{N})$ queries in the worst case. There are also a lot of other quantum algorithms, like the Deutsch-Josza algorithm [2], but this report will focus on Grover's algorithm and occasionally mention Shor's algorithm.

These two quantum algorithms are the most well known algorithms in the field of quantum computing, as they were the first ones proven to be faster on a quantum computer than existing algorithms on a classical computer [1]. Both have been also succesfully experimented on a small amount of qubits, and they have applications and use cases in the real world. For example, if large enough and functioning quantum computers are built, these algorithms have disruptive results in the field of cryptography.

The rest of the report will be divided as follows. In Section 2, related work is discussed. I will give both a summary and explanation about Grover's algorithm, how it has been researched in the past and what are the application areas of it. I will also discuss about the state of the research combining quantum computing and databases. In Section 3, I will give a more detailed explanation about quantum computing and Grover's algorithm. In Section 4, I will discuss about the weak points I have found in the picked papers. In Section 5, I will present experiments regarding Grover's algorithm by

using Qiskit, an open-source programming framework for quantum computing. I will run Grover's algorithm on a real quantum computer through IBM's cloud services and on a quantum computer simulator, and compare and analyze the results. Finally in Section 6, I will give a brief conclusion to summarize this report.

2. Related work

In this Chapter, I will present the history of Grover's algorithm, how it has been researched and what are its applications outside of databases and data management.

2.1 Data searching

As previously mentioned, following the founding of Shor's algorithm, Grover's algorithm was first introduced by its creator Lov K. Grover in 1996 [4]. Grover's algorithm is a quantum algorithm, which provides a quadratic speed-up for searching an unsorted database. Let there be a database, which has N elements, and there is one exact element, which needs to be found without knowing anything about the database and its structure. With a classical computer the best running time algorithm to find the element would need in the worst case $O(N)$ queries. On a quantum computer, Grover's algorithm would find the wanted element in the worst case with $O(\sqrt{N})$ quantum queries. Grover's algorithm has been proven to be the fastest algorithm for searching an unsorted database [2, 7] and has been under extensive research in the field [2].

Grover's algorithm can be used both for searching a single entry or multiple entries from an unsorted database [2]. The optimal amount needed for finding M entries from a N–sized database is $\frac{\pi}{4}\sqrt{\frac{N}{M}} = O(\sqrt{\frac{N}{M}})$ quantum queries. Grover's algorithm does not only limit to a full database search, as it can be also used for a partial database search for efficiency reasons. However, it is case by case when this partial database search is more efficient than a full database search. This partial search algorithm is proven to need $(\frac{\pi}{4} + \frac{\beta(K)-\eta(K)}{\sqrt{K}})\sqrt{N}$ quantum queries. However, there is also an optimized version of this partial search algorithm, which is obtained by optimizing the coefficient $\beta(K) - \eta(K)$.

Other than using Grover's algorithm for quantum database search, there is not much of research or literature about combining quantum computing and data management [8]. Quantum computing has some applications in the field of optimizing database queries, but it will not be discussed further in this report.

2.2 Cryptanalysis

Both Shor's algorithm and Grover's algorithm have applications in the field of cryptanalysis [3], which has led to the development and research of post-quantum cryptography [8]. In theory, Shor's algorithm can be used for decrypting the encryption keys used in asymmetric* encryption, such as RSA, through efficient integer factorization. In reality, this would need a big enough and functioning quantum computer to be considered as a threat. Meanwhile, Grover's algorithm can be used for decrypting the encryption keys used in symmetric cryptography. In this case we are especially talking about the Advanced Encryption Standard (AES) scheme. Grover's algorithm can be used to gain quadratic speed-up in brute-force attacks related to exhaustive key search in AES. Exhaustive key search means that every possible solution for the encryption key is tried one at a time to decrypt the encryption key.

2.3 Other application areas

Grover's algorithm can be applied to some NP-complete problems, like *3-SAT*† [1] and *graph-coloring* [7], as they can be modeled as search problems.

Grover's algorithm can also be used as a subroutine for many larger algorithms, thus providing a speedup [2]. For example, in the area of quantum machine learning, quantum k-means clustering algorithm uses Grover's algorithm as a subroutine, providing a quadratic speedup [9].

*Also known as public-key encryption.

†Also known as boolean 3-satisfiability

3. Detailed methods and ideas in previous works

In this Chapter, I will introduce quantum computing in more detail. I'll present how information is handled and presented inside a quantum computer compared to a classical computer, why (in theory) quantum computers are more powerful than classical computers and what are the challenges in quantum computing and building (large-scale) quantum computers. We will also present the functioning of Grover's algorithm in more detail, such as how it finds the right element in the database and other properties.

3.1 Quantum Computing in Detail

3.1.1 Quantum information

In a classical computer, information is described and handled as bits. A single bit can only have a single value of 0 or 1 at a time. These bits are then manipulated through different gates, such as *NOT* and *NAND*, and these multiple gates form circuits.

However, a quantum computer uses quantum bits, also known as qubits, as its unit of information [7]. Qubits are usually presented by using the bra/ket-notation[†] and the classical bit values of 0 and 1 can be presented as qubits $|0\rangle$ and $|1\rangle$. What is special about qubits compared to classical bits, is that qubits can be in a linear combination of states $|0\rangle$ and $|1\rangle$ [6]. This state is called a superposition of states $|0\rangle$ and $|1\rangle$, e.g., $x\,|0\rangle + y\,|1\rangle$. In this case numbers x and y are complex numbers, also called probability amplitudes, such that $|x|^2 + |y|^2 = 1$ [8]. This can be interpreted as measuring the value $|0\rangle$ with the probability of $|x|^2$ and value $|1\rangle$ with the probability of $|y|^2$. These qubits are then controlled through quantum circuits, which consists of quantum gates [6]. Some examples of one-qubit gates are the *NOT* and *Hadamard* gates.

[†]Sometimes also referred as the Dirac notation

3.1.2 Quantum parallelism

The interest in quantum computing is due to the fact that quantum effects in quantum systems provide exponential parallelism, called quantum parallelism [7]. This can then be harnessed to solve certain computational problems faster than with a classical computer. Both Shor's and Grover's algorithms take advantage of this phenomenon, which results in them being so efficient.

A register of bits in a classical computer can store 2^n values, where n is the amount of bits, and only one of the values can be present at a time [7]. However, a register of qubits can be in a superposition of all 2^n values. This exponential growth of the state space is the reason why quantum computers are believed to be exponentially faster than classical computers. In other words, when a classical computer can produce a single output for one input value, a quantum computer can produce all of the possible outputs for the input in the same time.

On a classical computer, speeding up computations can be achieved through parallelization. Achieving exponential decrease of time used for computation requires an exponential increase in the amount of needed processors and hence, an exponential increase in physical space needed for the processors [7]. In quantum systems the situation is different, as parallelism increases exponentially in relation to the size of the system. Therefore, exponential increase in parallelism scales linearly in relation to the needed physical space.

3.1.3 Challenges in quantum computing

In quantum computing, there are some trade-offs related to the exponential computing power. When a quantum procedure is run and then the result is read, this disturbs the quantum state by destroying the rest of the quantum states [7]. Also, as quantum algorithms are probabilistic due to the nature of quantum mechanics, the measured result might not even be the one we are looking for. However, there are some techniques how to increase the probability of getting the right result by modifying quantum states. In the next Section 3.2, we will inspect how Grover's algorithm does this.

What comes to the implementation of quantum computers, a big challenge is decoherence [7]. Quantum states are very fragile and prone to distortions coming from the surrounding environment. Therefore, quantum computers need a very isolated environment, so that the quantum states do not interfere with the outside environment, making the computation and measuring reliable. Quantum error correction is a technique used for reconstructing the quantum states without errors.

3.2 Grover's Algorithm in Detail

Grover's algorithm can be formally defined as a function [6]

$$f : \{0, \ldots, N-1\} \to \{0, 1\}, \tag{3.1}$$

which recognizes:

$$f(i) = \begin{cases} 1 \text{ if } i \text{ is the searched element,} \\ 0 \text{ otherwise.} \end{cases} \tag{3.2}$$

Grover's algorithm uses two quantum registers: the first one is used to store n qubits, and the second one is used to store only one qubit [6]. Let $N = 2^n$ for some integer n, and U_f a quantum gate:

$$U_f : |x, 0\rangle \to |x, P(x)\rangle ,$$

where $P(x)$ is a function, which tells if x makes the statement true by returning 1 [7].

The first step of Grover's algorithm is to initialize the first register, which contains the superposition of all possible 2^n values $x_i \in \{|0\rangle , \ldots , |2^n - 1\rangle\}$. Hence, we get the register

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x\rangle . \tag{3.3}$$

Now due to quantum parallelism, when U_f is applied to the first register, P is computed for all x_i, which leads to state of the register being in a superposition state

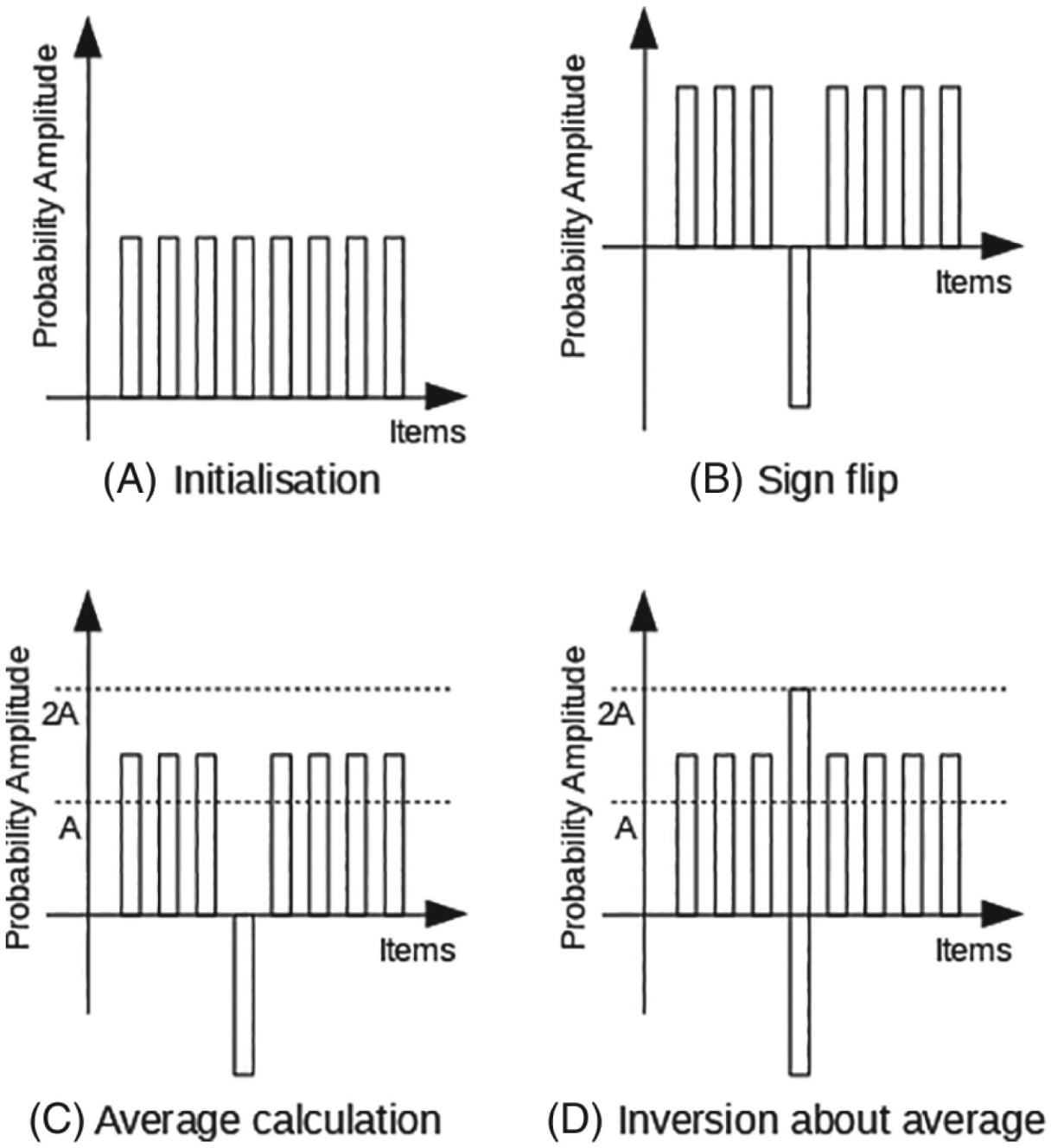
$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, P(x)\rangle . \tag{3.4}$$

Now any value x_0, which results in making $P(x_0)$ true, the quantum state $|x_0, 1\rangle$ is included in the superposition. As every quantum state has the amplitude of $\frac{1}{\sqrt{2^n}}$, the probability when randomly measured getting x_0 is $\frac{1}{2^n} = 2^{-n}$, which is significantly small. The technique here is to modify the quantum state in 3.4 so that the amplitude of vectors $|x_0, 1\rangle$, which make P true is significantly higher, than the amplitude of vectors $|x, 0\rangle$, which make P false. The computing of $P(x_i)$ and changing the amplitude is then iterated $\frac{\pi}{4}\sqrt{2^n}$ times, and then the result is obtained by measuring the last qubit. Therefore, as a result of the amplitude change, it has a high probability of being 1. This technique of changing the amplitude of the states, which have the desired result is called Amplitude Amplification and is a common technique when working with quantum algorithms [6]. I will not present Amplitude Amplification on any deeper level, however image 3.1 shows the flow of the Grover's algorithm, as presented in [9].

It has also been researched that Grover's algorithm is efficient only to a certain factor [7]. The most optimal factor is $\frac{\pi}{4}$, which provides a failure rate of 2^{-n}. More

iterations grow the failure rate, and after $\frac{\pi}{2}\sqrt{2^n}$ iterations the failure rate is almost 1. This behaviour differs from classical algorithms, where more iterations usually provide better results. When iterating quantum procedures, the results get better only up to a certain point, and after that they will get worse again. So to get useful results, it is important to know when to stop the iterating.

Figure 3.1: Flow of the Grover's algorithm



4. Weak points of papers and open challenges

I found a few weak points in the papers used in this report. The main weak point, which I noticed is that the research on quantum computing and quantum algorithms is mostly theoretical. The papers do not present any empirical benchmarking or exact numbers what comes to the computational power of quantum computing and efficiency of quantum algorithms compared to classical computers. It would be interesting to see comparison between classical and quantum computers, for example in the amount of needed qubits, memory, time or energy efficiency for a same search related task.

However, I think there are multiple reasons for the lack of exact benchmarks. First of all, quantum computing is a fairly new topic and some of the papers used for this report are fairly old. Also as mentioned in [8], there is not a lot of research, which combines databases and quantum computers. Second of all, quantum computing might be a discipline, which develops faster than what the scientific articles get published. As working quantum computers have groundbreaking consequenses in the field of cryptography, it might be that some of the research is kept under the radar on purpose.

Maybe one of the greatest open challenges is how powerful quantum computers can be realistically built and what they are capable of computing better than classical computers. One challenge is also to find an algorithm, which would give a greater speed-up than the Grover's quadratic one.

5. Ideas, algorithms and experiments

In this Chapter, I will describe and explain the experiment setup and the experiments related to Grover's algorithm. The experiment is quite trivial, but enough to demonstrate the functioning of Grover's algorithm in a search related problem. The same task will be run on IBM's cloud services on a quantum computer simulator and on a real quantum computer, after which the results are analyzed.

5.1 Problem and Experiment Setup

Qiskit is an open-source framework backed by a company called IBM, which can be used for coding quantum algorithms using Python without needing a broad background in quantum physics [5]. With Qiskit you can run your quantum algorithms on a simulator or on a real quantum computer through multiple providers, such as IBM or Amazon Web Services. In this report, Grover's algorithm is run on both situations, on a simulator and on a real quantum computer through IBM's cloud services. IBM was chosen as a provider due to extensive documentation and compatibility, as Qiskit is maintained by IBM.

For the experiment, a quantum computer called *ibm_oslo* was used. It was chosen due to being free of charge to use, and due to geographical location to minimize latency. However, as it can be used free of charge, this is shown as long queues to get your job running on the system. Also it has a lower quantum volume and less qubits than the ones included in the paid premium plan. More precisely, *ibm_oslo* has Falcon r5.11H processor, 7 qubits of computing power, quantum volume of 32 and can do around 2600 circuit layer operations per second (CLOPS).

For the simulations, the used simulator is called *QASM*. *QASM* is a general-purpose simulator for simulating quantum circuits ideally and subject to noise modeling, while supporting 32 qubit computations.

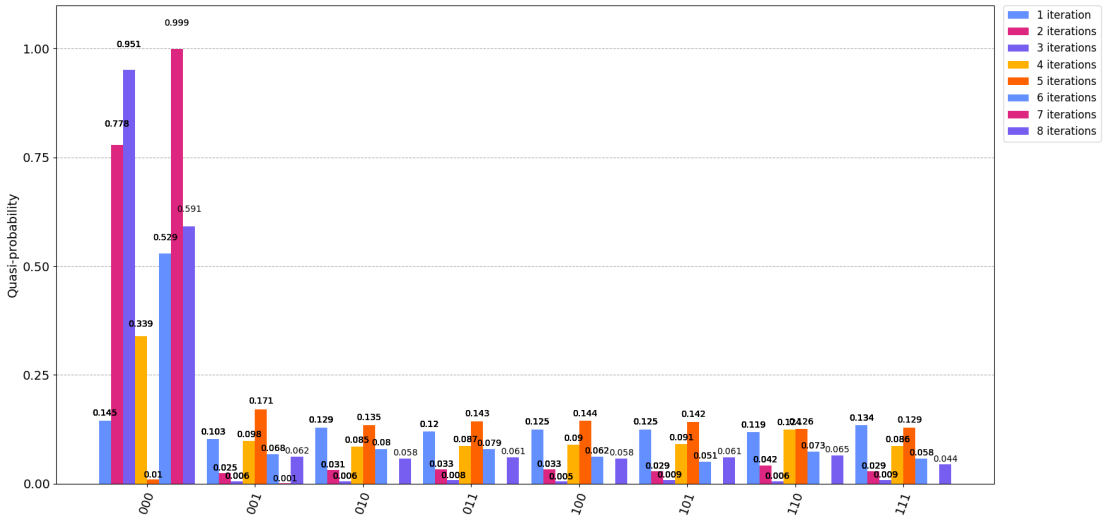
The problem used here is quite simple, but is enough to demonstrate the idea and functioning of the Grover's algorithm. Let there be an array, which will have 8 entries $[0, \ldots, 7]$, from which a random value will be picked. In this experiment the value to be found was 0 or $|000\rangle$. The random value is then given to an *oracle*, which is a black box telling if the guessed entry is correct or not. After that Grover's algorithm is used to

find the correct element, which is the one that has the highest probability. As $2^3 = 8$, 3 qubits is needed for the computation. Grover's algorithms is also iterated different amount of times to see how this affects to the results and probabilities. Visualization is done by using Python and matplotlib library. The graphs are also available through IBM's cloud services, and they are available for viewing in GitHub, see Appendix B. The whole program code can be found from behind the link in Appendix A.

5.2 Results of the experiments

5.2.1 Simulator

Figure 5.1: Results and visualization given by the simulator



The different colors of the bars indicate on how many iterations the Grover's search algorithm is run. The y-axis is the probability of getting a certain entry with a certain amount of iterations, and this probability can also be read from the top of the bars in the bar chart. The x-axis represents the entries in the database, in this case the values $[0, \ldots, 7]$, which can be encoded as qubits $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$ and $|111\rangle$.

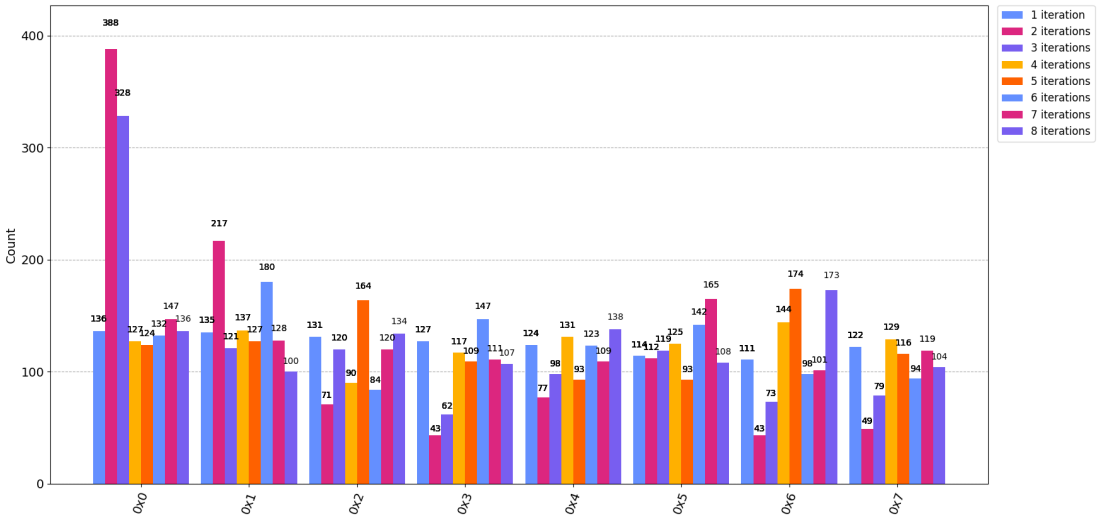
For example, the bar chart above shows that with 1 iteration, the probability of getting $|000\rangle$ is 0.145, $|001\rangle$ is 0.103 and so on. The probabilities, which are higher on the other values than on $|000\rangle$, are therefore wrong. So it can be seen that the simulator found the correct entry on all, but the 5 iterations. In 3.2 it was stated that the optimal amount of iterations is $\frac{\pi}{4}\sqrt{2^n} = \frac{\pi}{4}\sqrt{8} = 2.22$. However, it seems that with 3 iterations we

get the largest probability of 0.951. For some reason, with 7 iterations, the probability is almost 1, but it has not computed any probabilites for other values. This behaviour could be investigated more, as it is not clear if this is some programming error in my code or in the framework, or is this just natural behaviour of the algorithm.

More detailed results of this test run can be found from Appendix B.

5.2.2 Real quantum computer

Figure 5.2: Results and visualization given by the real hardware



It can be seen that the output is bit different with the real quantum computer than on the simulator, as the graph has now frequencies of different outcomes, rather than probabilities. However, the probabilities can be derived by dividing the frequency by the amount of all observations, which sums up to 1000. It can be seen that at first with iterations $1-3$, the algorithm finds the correct result $|000\rangle$. At 4 iterations and beyond, it gets the wrong result. In 3.2 it was stated that the algorithm should be in this case iterated $\frac{\pi}{4}\sqrt{2^n} = \frac{\pi}{4}\sqrt{8} = 2.22$ times. It is clearly shown in the graph that the largest frequency, 388 outcomes of $|000\rangle$, is achieved when the algorithm is iterated 2 times. Therefore, the experiments some what correlates with the researched papers.

More detailed results of the job executed on the quantum computer can be found from Appendix B.

5.2.3 Comparing the results

As stated in Chapter 3.2, there is an optimal factor for the amount of iterations for the algorithm, and beyond that factor the results will become worse. This observation seems to be correct when the Grover's algorithm is run on the real quantum computer, as after 3 iterations the algorithm gives wrong results. When run on the simulator, it only gave a wrong result once with 5 iterations. Therefore, it seems that the simulator does not follow the results in the cited papers.

I find it odd that the results differ so much between the two experiments, especially the simulator giving different results as proposed by the theory. This might just show how counterintuitive and unexpected the world of quantum mechanics can be.

6. Conclusion

Grover's algorithm is a quantum algorithm, which provides a quadratic speedup when using a quantum computer to search unstructured databases. Grover's algorithm is one of the well known and researched quantum algorithms, which development sped up the research in the field of quantum computing. Grover's algorithm has variations, as it can be used for partially searching a database instead of doing a full search. It has also applications outside of databases and data searching. Grover's algorithm can be applied to decrypt encryption keys used in symmetric cryptographic schemes like AES, to solve NP-complete problems, like the 3-SAT problem and as a subroutine for other algorithms, like for the quantum k-means clustering algorithm.

In this report, it was experimented how running Grover's algorithm on a simple search problem first in a simulator, and then on a real quantum computer differed. Conclusion was that the results gotten from the real quantum computer corresponded to the theoretical results in the papers cited on this report. It is shown that when the amount of iterations used in computing Grover's algorithm increases, the results get worse. However, in the simulator, when the iterations are increased, the gotten results do not show any change for the worse, excluding some individual ones. Therefore, it seems that the simulator does not obey the results in the researched papers.

Quantum computing is a new and emerging technology, which will have a significant influence to the field of computer science as we know it. If large enough and functioning quantum computers will be built, it would break the known public-key cryptography, which would have unfortunate consequenses. On the other hand however, the exponential computing power could possibly be used to solve some complex computing tasks, which would benefit various research fields. Most likely quantum computers will be used together with classical computers, in a hybrid manner. Same way as GPU's nowadays, quantum computers could be used to run only certain specific and complex computing problems. There is not yet a lot of research in the field combining quantum computing and databases, but this might change in the future when the development of quantum computing and quantum computers goes forward.

References

[1] A. Ambainis. Quantum search algorithms. *ACM SIGACT News*, 35(2):22–35, 2004.

[2] P. R. Giri and V. E. Korepin. A review on quantum search algorithms. *Quantum Information Processing*, 16:1–36, 2017.

[3] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt. Applying grover's algorithm to aes: quantum resource estimates. In *Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings 7*, pages 29–43. Springer, 2016.

[4] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[5] D. Koch, L. Wessing, and P. M. Alsing. Introduction to coding quantum algorithms: A tutorial series using qiskit, 2019.

[6] C. Lavor, L. Manssur, and R. Portugal. Grover's algorithm: Quantum database search. *arXiv preprint quant-ph/0301079*, 2003.

[7] E. Rieffel and W. Polak. An introduction to quantum computing for non-physicists. *ACM Comput. Surv.*, 32(3):300â335, sep 2000.

[8] V. Uotila. Synergy between quantum computers and databases. In *Proceedings of the VLDB 2022 PhD Workshop co-located with the 48th International Conference on Very Large Databases (VLDB 2022)*, CEUR Workshop Proceedings, International, Sept. 2022. CEUR. null ; Conference date: 05-09-2022 Through 05-09-2022.

[9] Y. Zhang and Q. Ni. Recent advances in quantum machine learning. *Quantum Engineering*, 2(1):e34, 2020.

Appendix A. Code used in running and analyzing Grover's algorithm

The code used in Section 5.2 related to running, experimenting and visualizing Grover's Algorithm can be found from my GitHub: https://github.com/AlluSu/Quantum-Algorithms

Appendix B. Results of the test runs used in experimenting

After each run, the results are saved in IBM's services where they can be viewed and analyzed later. The results from the experiments in Section 5.2 can be viewed in my GitHub: https://github.com/AlluSu/Quantum-Algorithms/tree/master/test-results/