

ECM3428: Algorithms that changed the World

Grover's Algorithm

A fast quantum database search

650043500

13 December 2018

Abstract

Quantum algorithms are proving important to many fields as they repeatedly display, often exponential, speed up over their classical counterparts. In this paper, we discuss Grover's algorithm which provides quadratic speed up over its classical alternative of searching for an element in an unstructured database. It provides the mathematical grounding necessary to understand the algorithm while touching on the proof for the quadratic speed up in time complexity. With the main limitations of quantum algorithms coming from hardware specific decoherence problems, this paper hopes to generate excitement about the future prospects of this young yet growing area of Computer Science. There is no question that this algorithm will be instrumental in changing the future world.

I certify that all material in this dissertation which is not my own work has been identified.

1 Introduction

In 1992, Deutsch and Jozsa developed the first instance by which a quantum algorithm was faster than a classical one for the same task [1]. 6 years later, Simon devised an algorithm for Simon's Problem; the first with exponential speed up over its classical counterpart [2]. The Deutsch-Jozsa algorithm and Simon's algorithm have limited applications besides acting as proof of speedup. Shor's factoring algorithm for factoring integers and finding discrete logarithms built off of Simon's algorithm and was the first example with notable applications that had exponential speedup [3]. As most of modern cryptography is based off factoring prime numbers this algorithm would render public-key cryptography impractical, provided quantum computers could operate without succumbing to quantum noise and other quantum-decoherence phenomena. Quantum algorithms are important because of the wide range of applications they have to numerous fields, often offering exponential speed up. This paper covers a brief overview of the concepts necessary for understanding Grover's algorithm which provides quadratic speedup over classical algorithms for searching an unstructured database.

2 Classical Computing

Computers in the classical sense input and output data encoded as zeros and ones. On a physical level these zeros and ones, called bits, are the most basic unit of information corresponding to the quantity of electrical potential stored in capacitors in semiconductor memory, such as RAM. Gates perform logical operations on binary inputs, producing a single binary output. They are implemented in classical computers using transistors that act as electronic switches. The universal gates - AND, OR and NOT - from Fig. 1. can be combined to create classical circuits to perform functions on input data.








Name	NOT	AND	NAND	OR	NOR	XOR	XNOR																																																																																																
Alg. Expr.	\overline{A}	AB	\overline{AB}	$A + B$	$\overline{A + B}$	$A \oplus B$	$\overline{A \oplus B}$																																																																																																
Symbol																																																																																																							
Truth Table	<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	X	0	1	1	0	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	B	A	X	0	0	0	0	1	0	1	0	0	1	1	1	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	B	A	X	0	0	1	0	1	1	1	0	1	1	1	0	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	0	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	0	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	1
A	X																																																																																																						
0	1																																																																																																						
1	0																																																																																																						
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	1																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	1																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	1																																																																																																					

Figure 1: Classical Gates

Since AND and OR gates are irreversible, meaning given an output we cannot determine the input, any circuit consisting of these gates is also irreversible. Circuits of this kind cannot be easily generalised to quantum gates.

Obviously a classical search algorithm for an unstructured array of N elements will take, on average, $\frac{N}{2}$ attempts to find the searched item. In the worst case it may even take N attempts giving it a time complexity of $O(N)$. A classical implementation of Grover's algorithm would require a Boolean function that outputs a 1 if the searched element is found and 0 otherwise.

3 Quantum Computing

3.1 Linear Algebra of Qubits

In quantum computers the bits 0 and 1 become qubits denoted in Dirac notation as $|0\rangle$ and $|1\rangle$, respectively. Unlike classical bits that must be either zeros or ones, qubits can be in a coherent superposition of multiple states at the same time. Physically, they resemble classical bits via either the spin of an electron or polarisation of a photon, depending on the quantum computer architecture. Furthermore, qubits are represented by vectors that follow all eight axioms of vector spaces. Thus, a qubit $|\psi\rangle$ can be a linear combination of two states $|0\rangle$ and $|1\rangle$, such that:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

$|\psi\rangle$ is the superposition between the states with amplitudes $|\alpha|^2$ and $|\beta|^2$. Quantum mechanics states that if we measure $|\psi\rangle$ we get $|0\rangle$ or $|1\rangle$ with probabilities $|\alpha|^2$ and $|\beta|^2$, respectively. There are two ways in which qubits can interact: measurement and linear transformations. The act of measurement produces a nondeterministic collapse of the superposition to either of the states. Hence, in the case of two qubits, there are two possibilities for the superposition to collapse to, which ensures the norm of $|\psi\rangle$ is given by:

$$\| |\psi\rangle \| = \sqrt{|\alpha|^2 + |\beta|^2} = 1 \quad (2)$$

More generally, as superpositions can occur between multiple qubits, it follows that the sum of their corresponding probabilities must equal 1. Upon measurement, a state must collapse to one of the qubits that makes up the superposition. Measurement does not provide information about the amplitudes. However, the second interaction is a linear transformation, which causes the amplitudes to change while holding the constraint of Eq. 2. An example of this is the Unitary transformation U that take unit vectors into unit vectors. We will see more of these in the following section. This can be visualised best with the Bloch sphere from Fig. 2. If we imagine β increasing by the same amount that α decreases then $|\psi\rangle$ will move around the surface of the Bloch sphere accordingly.

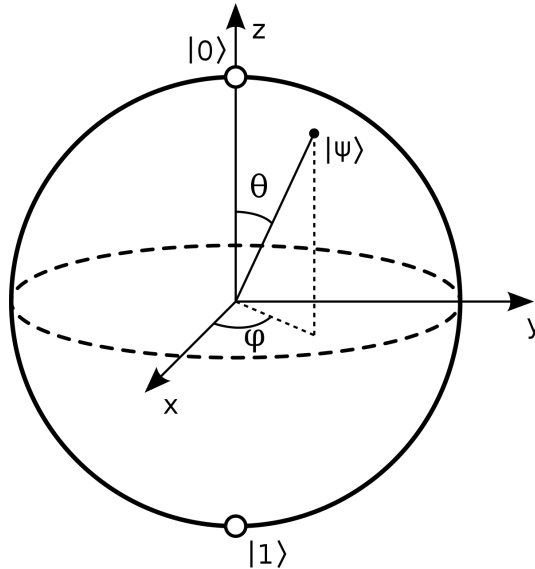


Figure 2: Bloch Sphere

Another operation useful for understanding Grover's algorithm is the tensor product denoted with the \otimes operator. The tensor product of two qubits is usually represented in shorthand by $|0\rangle \otimes |0\rangle = |00\rangle$ and the four possible outcomes for the $|0\rangle$ and $|1\rangle$ are shown in Eq. 3.

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, |01\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, |10\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, |11\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (3)$$

The dual vector of $|\psi\rangle$ is denoted by $\langle\psi|$ and corresponds to the transposed complex-conjugated vector. The product of these two vectors results in the *outer product* which is an $n \times n$ matrix, like below:

$$|\psi\rangle \langle\psi| = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} \begin{bmatrix} a_1^* & \dots & a_n^* \end{bmatrix} = \begin{bmatrix} a_1 a_1^* & \dots & a_1 a_n^* \\ \dots & \ddots & \dots \\ a_n a_1^* & \dots & a_n a_n^* \end{bmatrix} \quad (4)$$

3.2 Quantum Circuits

The Hadamard gate is an important one-qubit gate [4]. It's constructed from the Hadamard transform; a $2^m \times 2^m$ matrix scaled by a normalisation factor. If the input is $|0\rangle$ the Hadamard gate creates a superposition of states with equal weights, which is a general feature for two or more qubits.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (5)$$

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle \quad (6)$$

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle \quad (7)$$

The Oracle gate \hat{O} is a unitary gate that flips the amplitude of a searched component in a superposition and leaves everything else unchanged [5]. It uses a similar Boolean function to the one mentioned previously whereby if the input to the function is the searched element x^* it outputs a 1 and a 0 otherwise.

$$f : X \rightarrow \{0, 1\},$$

$$f(x) = \begin{cases} 1, & \text{if } x = x^*. \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

This function is used in the Oracle gate in the following way. It shows how the amplitude of the searched qubit gets flipped but the others stay the same.

$$\begin{aligned} |\psi_{\hat{O}}\rangle &= \hat{O}(|\psi_i\rangle, |x^*\rangle) \\ &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \hat{O}(|x\rangle, |x^*\rangle) \\ &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle |x^*\rangle \\ &= -\frac{1}{\sqrt{2^n}} |x^*\rangle + \sum_{x \in \{0,1\}} \frac{1}{\sqrt{2^n}} |x\rangle \end{aligned} \quad (9)$$

Finally, the diffusion operation raises the amplitude of a searched element [6]. It results in a net transfer of equal amplitude from each that is not the searched state, to the searched state. The operator $2|\psi\rangle \langle\psi| - I$ is called inversion about the mean.

$$\begin{aligned} |\psi_D\rangle &= (2|\psi\rangle \langle\psi| - I) |\psi_{\hat{O}}\rangle \\ &= \frac{2^{n-2} - 1}{2^{n-2}} |\psi\rangle + \frac{2}{\sqrt{2^n}} |x^*\rangle \end{aligned} \quad (10)$$

4 Grover's Algorithm

Definition. Given an unstructured database of N elements in $X = \{x_0, x_1, \dots, x_{N-1}\}$ and a Boolean function $f : X \rightarrow \{0, 1\}$ find x^* in X such that $f(x^*) = 1$ [7].

Algorithm 1 Grover's Search

- 1: **Input:** $f(x)$, n
 - 2: **Output:** x^*
 - 3: Calculate the optimal $k = \text{round}(\frac{\pi}{4}\sqrt{2^n})$ iterations
 - 4: Create equally weighted superposition of states in *register1*: $|\psi_0\rangle \leftarrow \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$ ▷ Eq. 11
 - 5: Pass x^* through Hadamard gate: $|-\rangle \leftarrow H|x^*\rangle$ ▷ Eq. 7
 - 6: **for** i **in** k **Grover Iterations** **do**
 - 7: Pass $|\psi_i\rangle$ and $|-\rangle$ through the Quantum Oracle gate \hat{O} ▷ Eq. 9
 - 8: Pass $|\psi_i\rangle$ through the Diffusion Operator D ▷ Eq. 10
 - 9: **end for**
 - 10: **Measure** $|\psi_k\rangle$
-

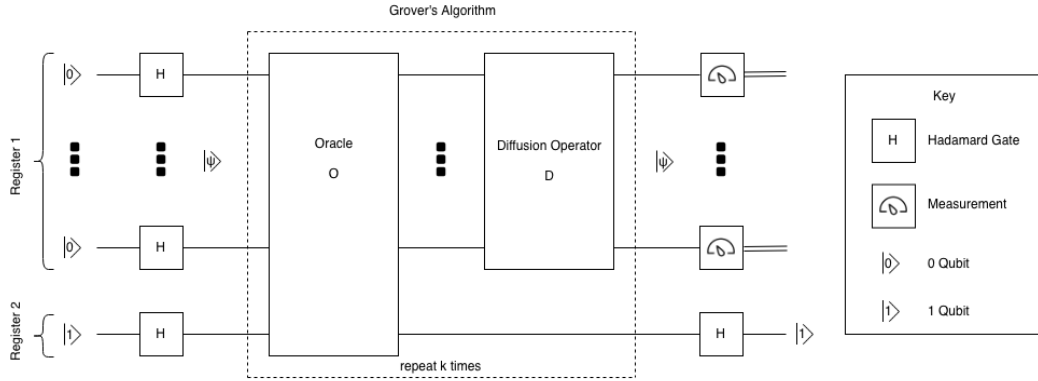


Figure 3: Grover Circuit Diagram

The proof for the optimal k value is in the following section. The next step uses the Hadamard gate from Eq. 3 to create a superposition $|\psi_0\rangle$ of all the basis states in *register1* with equal amplitude $\frac{1}{\sqrt{N}}$. Firstly, n qubits are initialised to $|0\rangle$ and the Hadamard gate produces a superposition of all to create the computational basis states in the following way, where $A^{\otimes n}$ tensors A with itself n times [8]:

$$\begin{aligned}
 |\psi_0\rangle &= H|0\rangle^{\otimes n} \\
 &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n} \\
 &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle
 \end{aligned} \tag{11}$$

We also apply a Hadamard gate to *register2*. Let us assume that $|x^*\rangle = |1\rangle$ then applying the H to this state gives $|-\rangle$ from Eq. 7. Now much like in the classical search algorithm we need a function $f : X \rightarrow \{0, 1\}$ that returns 1 if the $x = x^*$ and 0 otherwise. From here we use the Oracle operator to flip the amplitude of the searched bit in $|\psi\rangle$ [9]. The final gate in the Grover iteration is the diffusion operator D . It magnifies the amplitude of the searched qubit by flipping it about the absolute mean of all the amplitudes in the superposition, while leaving everything else relatively unchanged [10]. The Grover iteration is repeated k times with $|\psi\rangle$ approaching $|x^*\rangle$ with each iteration. Hence, with each iteration the probability of $|\psi\rangle$ collapsing to $|x^*\rangle$ upon measurement increases.

4.1 Geometric Proof

Take $|x^*\rangle$ and $|0\rangle$ as vectors within the computational basis, we know then that $|\psi\rangle$ is non-orthogonal. Grover's algorithm is essentially multiple applications of \hat{O} and D which rotates $|\psi\rangle$ in the plane spanned by $|x^*\rangle$ and $|\psi\rangle$, holding the unit norm. The angle of rotation is given by:

$$\cos\theta = 1 - \frac{1}{2^n - 1} \quad (12)$$

\hat{O} rotates $|\psi\rangle$ by θ clockwise and then D rotates $|\psi\rangle$ by 2θ degrees counter-clockwise, on the Bloch sphere. Since θ depends on n , the number of iterations until completion also depends on n . The number of iterations needed for completion is given by the following equation:

$$k = \text{round}\left(\frac{\pi}{4}\sqrt{N}\right) \quad (13)$$

Thus, the algorithm is correct and optimal [11].

4.2 Complexity

Each Grover Iteration first flips the amplitude of the searched element then flips it back around the mean of all amplitudes. Intuitively, as shown by Eq. 9, we see that each Grover Iteration will bring the amplitude of $|x^*\rangle$ up by around $\frac{1}{\sqrt{2^n}}$ each iteration. After $\sqrt{2^n}$ iterations the amplitude of $|x^*\rangle$ will be close to 1 which means that upon measurement the superposition will collapse to $|x^*\rangle$ with probability close to 1. This gives it a complexity of $O(\sqrt{N})$. Thus, it produces a quadratic speed up over classical universal searches.

5 Conclusion

A limitation of the classical simulation of Grover's algorithm attached with this project (besides space complexity) is that our amplitude for the searched state can actually exceed 1, which of course is impossible physically as it breaks Eq. 2. Aside from this the limitations of Grover's Algorithm typically come from hardware related problems such as the decoherence effect. Unlike the Deutsch-Jozsa and Simon's algorithm, Grover's algorithm has applications in a broad range of quantum software of the future. Anything that requires the simple task of searching a list could make use of Grover's algorithm, making its applications extensive. More generally, quantum computing appears to be showing anywhere from quadratic to exponential speed up for the same tasks on a classical computer. This provides an exciting prospect for the future as the full potential for this technology is likely not yet known. What is known is that Grover's algorithm will be instrumental in changing the world of the future.

References

- [1] D. Deutsch and R. Jozsa. Rapid Solution of Problems by Quantum Computation. *Proceedings of the Royal Society of London Series A*, 439:553–558, 1992.
- [2] D. R. Simon. On the power of quantum computation. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, SFCS '94, pages 116–123, Washington, DC, USA, 1994. IEEE Computer Society.
- [3] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.
- [4] A. N. Akansu and R. Poluri. Walsh-like nonlinear phase orthogonal codes for direct sequence cdma communications. *IEEE Transactions on Signal Processing*, 55(7):3800–3806, July 2007.
- [5] Eva Borbely. Grover search algorithm. *CoRR*, abs/0705.4171, 2007.
- [6] John Wright. Lecture 4: Grovers algorithm, September 2015.
- [7] D. K. Ningtyas and A. B. Mutiara. Simulating grover’s quantum search in a classical computer. *CoRR*, abs/1003.1930, 2010.
- [8] Carlile Lavor, LRU Manssur, and Renato Portugal. Grover’s algorithm: quantum database search. *arXiv preprint quant-ph/0301079*, 2003.
- [9] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. ACM.
- [10] Lov K. Grover. From schroinger’s equation to the quantum search algorithm. *American Journal of Physics*, 69(7):769–777, 2001.
- [11] Christof Zalka. Grovers quantum searching algorithm is optimal. *Physical Review A*, 60(4):2746, 1999.