Grover's Search Algorithm

Name: BOKKA UDAYA PEDDIRAJU
ID Number: Ó1-02-04-10-51-21-1-19424

1 Summary

In computer science database is a collection of data, the data could be structured or unstructured. In structured database the data is organised in a predefined manner, example: In a sorted array the data is arranged in a sequential manner either ascending or descending order, on the other-hand In an unstructured database the data is not organised.

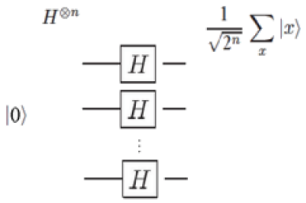
If we have a unstructured list of N items let's call the list as A, the task is to find out a particular item location (index) from the list A let's call it as target. If we use the classical searching algorithms to find the target, then in the best case on an average we have to check N/2 items and in the worst case we have to check the entire array (imagine we have started searching sequentially from index-0 and the target item is at the last index of the list). however we can find the item with highest probability by checking approximately $\sqrt{N}$ items from the list if we use the Grover's algorithm which is a quantum algorithm. In this paper I'll demonstrate how to implement the Grover's search algorithm with a database which consists of all the possible computational basis states of our qubits. The experimental results from the IBM's quantum computer are also included for reference (please check in my github link provided).

2 Steps for Implementing the Grover's algorithm

Grover's algorithm utilizes the superposition and phase interference to improve the speed of the search, there are three steps to implement Grover's algorithm among which amplitude amplification is the crucial step which plays a vital role. Let us look at the details of steps involved in the Grover's search algorithm.

2.1 Initialization

First we have to create a database with all the input states ranging from 0 to 2^n where n is an integer which indicates the number of qubits, and N = 2^n (size of the database), usually in real world database is already available for us and we have to search an item from it by applying Grover's algorithm. Now, for the chosen n qubits create a superposition state which represents all possible states, passing each qubit (with initial value 0) through a H gate or simply operating a n fold Hadamard gate on n qubits will give us the superposition state, of course the qubits are initialized with 0 or null vector. Let's call the resulting superposition state as $|s>$, this state represents the database with all the possible computational basis states. The mathematical and visual representations of $|s>$ are given below for better understanding.



$$|s> \;=\; \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x>$$

2.2 Phase inversion by the Oracle (U_f)

For example let's take n=2 qubits, then our database is comprised of all the possible computational basis states that our qubits can be i.e,. from $|00>$ to $|11>$. Let's call our target state as $|w>$ whose location we would like to find in the database (location in the sense imagine it as the index of an unsorted array), then when the Oracle matrix U_f is operated on the state $|s>$, then it shifts the phase of the target state $|w>$ that in the superposition state $|s>$ i.e,. only for x=w, it leaves the phase as it is for the remaining states.

$$U_f|s> = \begin{cases} -|x> \text{ if } x = w \\ |x> \qquad\qquad\quad \text{otherwise} \end{cases}$$

One way to realize the Oracle matrix U_f is create function a $f(x)$ such that it returns one only when the input target state $|w>$ matches with any of the states in the database state $|s>$ i.e,. x=w, then construct a circuit which defines the Oracle U_f (it's just a unitary operator) matrix which operates on the database state $|s>$ then shifts the corresponding state phase i.e,. $|w>$ in $|s>$ if and only if the target state $|w>$ is in the database. Anyway we apply Grover's algorithm by assuming the target item is in the database. Here I am giving you some pictorial representation of the initial state $|s>$ and how the oracle producing reflection on $|s>$ about the modified state $|s^{'}>$ which is the perpendicular state to $|w>$, generally U_f called as the reflection operator.

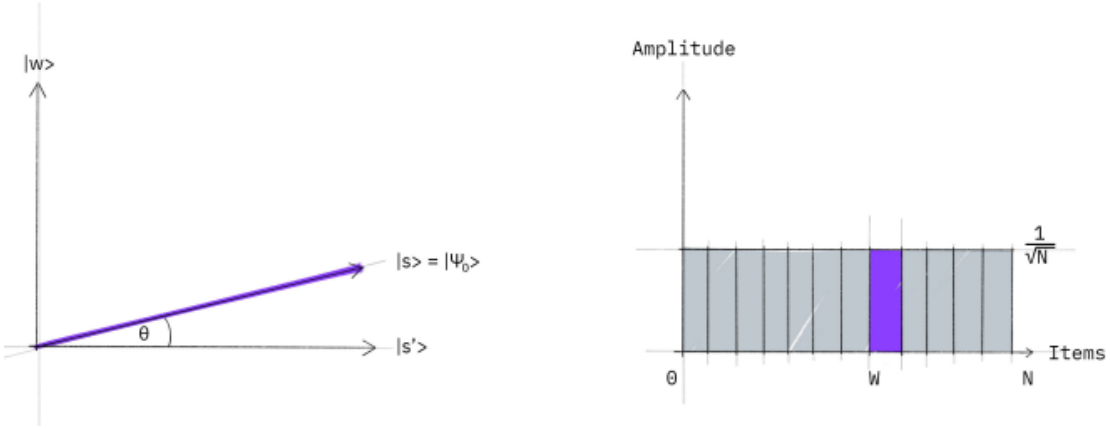


Figure 1: State Vector $|s>$ representation and the initial amplitude of the target vector $|w>$ in database, of course every state has equal probability initially which is represented interms of amplitude.

The above graph represents the two dimensional plane spanned by perpendicular vectors $|w>$ and $|s^{'}>$ and the initial state can be expressed interms of these as $|s>=\sin(\theta)|w>+\cos(\theta)|s^{'}>$, So θ can be expressed as $\theta = \arcsin < w|w> = \arcsin \frac{1}{\sqrt{N}}$. Now, let's see how the reflection or phase shift happens after the oracle matrix U_f operated on $|s>$, mathematically U_f= I-2$|w><w|$ (I is the identity operator).

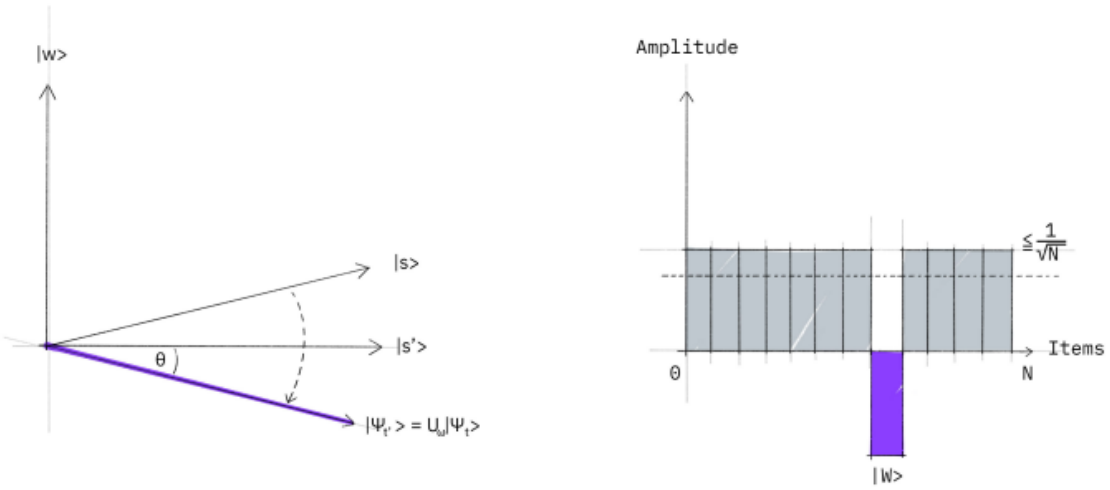


Figure 2: The state $|s>$ reflection with an angle θ after one oracle matrix U_f operation on $|s>$, This transformation means that the amplitude in front of the $|w>$ state that is in the database becomes negative, which in turn means that the average amplitude (indicated by a dashed line) has been lowered.

2.3 Inversion About the Mean by Diffuser Operator (U_s)

This step is called amplitude amplification stage because here we apply apply diffuser operator on the resultant state vector of the step2 where we applied reflection operator on the initial state. This operator reflects or inverts the phase of the input state around the mean i.e,. $|s>$, mathematically it can be expressed as U_s=2$|s><s|$-I, pictorially it can be viewed as.

We completed one iteration of the Grover's algorithm which increased the probability of finding our target state by some value. Now, we go to step two which is the phase inversion, then step3 to apply 2nd iteration, like this we have to apply these two steps t times to get the maximum probability of finding our target state in the measurement of the output state of the diffusion operator. After t iterations the state looks like $|\Psi_t>=(U_sU_f)^t|s>$. It is easy to find the value of t with the help of $\theta = \arcsin \frac{1}{\sqrt{N}}$, since we know that after t iterations our state aligns with the target state $|w>$

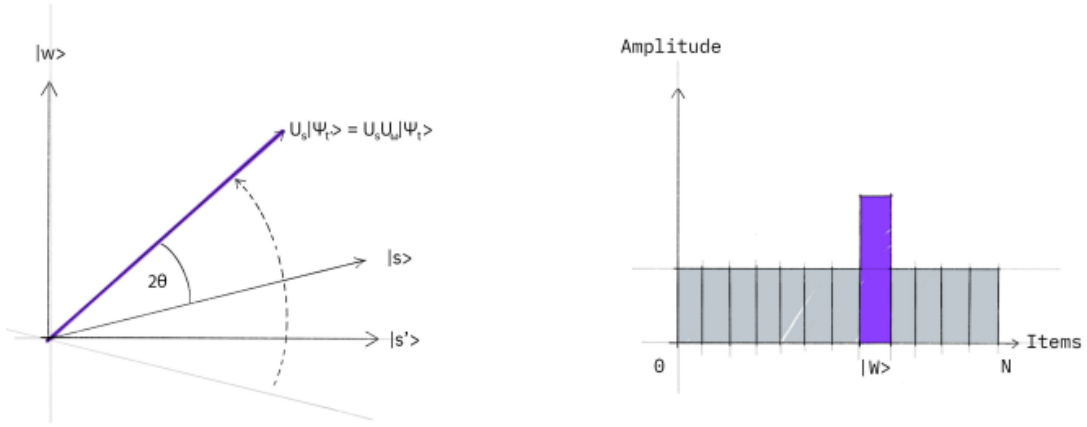


Figure 3: Representation of $(U_s)(U_f)|s>$, the rotation of the initial state towards the target state, also increase in the amplitude of the target state and decrease in the amplitude of other states.

which makes an angle of 90^0 with the state $|s^{'}>$ so we can relate this to the reflection angle θ by the following formula $(2t+1)\theta = \frac{\pi}{2}$, since our final state looks like $(U_s U_f)^t|s>=\sin(\frac{\pi}{2})|w>=|w>$. Okay, now here comes the question we said we can perform database search in $\sqrt{N}$ but we are using t many iterations?. Yes, but the value of t turns out to roughly $\sqrt{N}$ because the amplitude of the target grows linearly with the number of iterations approximately by $tN^{-\frac{1}{2}}$. However, since we are dealing with amplitudes and not probabilities, the vector space's dimension enters as a square root. Therefore it is the amplitude, and not just the probability, that is being amplified in this procedure. For multiple target states T it takes approximately $\sqrt{\frac{N}{T}}$ iterations.

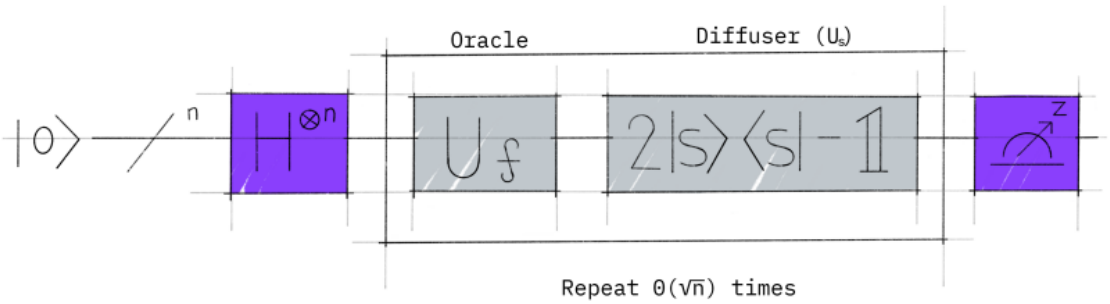


Figure 4: The complete picture of the Grover's Algorithm

Note that here the target state $|w>$ need not to be a single state of computational basis, it can be in any superposition state, means that our target can be in superposition or a single state and until and unless it is measured the resultant state won't collapse to particular state it stays in the same form as our target state. If you need to find the application of this kind, you can refer to the paper Grover Meets Simon– Quantumly Attacking the FX-construction, where Grover's algorithm applied to get the secret keys associated with the various cipher texts. of course without measurement only we can have these many keys (i.e,. more than one).

Conceptually everything seems perfect but how to implement the Oracle, diffuser and how to query the Grover's algorithm to get the index value of a randomly queried item in the database. Practically we need n qubits to store the database and another n qubits to store the index values, then one more qubit is required to introduce the phase kickback to the target state otherwise the diffuser considers phase shift introduced by oracle as global phase.

Explore the animation of Grover's algorithm at http://davidbkemp.github.io/animated-qubits/grover.html

I have implemented the Grover's algorithm with N=8 as database size and given link to my git hub page where you can find my code and example results. Most of the resources on internet like Qiskit text book implemented Grover's algorithm which is applicable to a particular target state only, like for example if you go to Qiskit textbook you can see the target state for two qubit circuit is taken as $|11>$ and they hard coded the Oracle and diffuser to work with this particular input target state only. So what if you want to apply Grover's algorithm to find the target state $|00>$ or some other, For that I have Implemented Grover's algorithm by considering all the possible cases i.e,. you can query any random state you wish to search from the database.

2.4 References

1)https://qiskit.org/textbook/ch-algorithms/grover.html

2)https://towardsdatascience.com/grovers-search-algorithm-simplified-4d4266bae29e

3)https://medium.com/swlh/grovers-algorithm-quantum-computing-1171e826bcfb

4)https://medium.com/visionary-hub/what-exactly-is-grovers-algorithm-a8f5dce1e1b3

5)http://davidbkemp.github.io/animated-qubits/grover.html

6)https://aapt.scitation.org/doi/pdf/10.1119/10.0004835

7)https://eprint.iacr.org/2017/427.pdf

8)Ashley Montanaro and Ronald de Wolf.A Survey of Quantum Property Testing Book.

9)Here's my github link for Grover's implementation is https://github.com/udayapeddirajub/Grover-s-Algorithm-Implementation-on-3-qubit-database.git