Grover's Algorithm

Cody M. Grant

(Dated: February 15, 2022)

I. PROBLEM

Grover's alg is a search algorithm. Specifically, showing it's speed superiority over classical computers in searching unstructured databases. Compared to classical, it is a quadratic speed up.

In an unstructured search, you are given a large list of N items. We want to locate a specific item, and classically you would have to search $N/2$ of the items in the list (on average). With QC and Grover's amplitude amplification trick, it is sped up to about $\sqrt{N}$ items. It also doesn't use the list's internal structure as a parameter, making it generic and why the speed up is so powerful (especially at really large N).

II. CREATING AN ORACLE

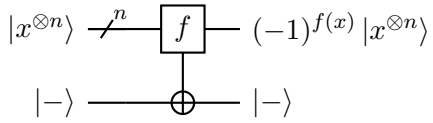
Assuming our database is comprised of all possible computational basis states. For example, with 3 qubits we'd have $|000\rangle, |001\rangle, ..., |111\rangle$ (or the states $|0\rangle \rightarrow |7\rangle$. The oracle first adds a negative phase to the state matching the solution:

$$U_\omega |x\rangle = \begin{cases} +|x\rangle, & \text{if } x \neq \omega \\ -|x\rangle, & \text{if } x = \omega \end{cases} \tag{1}$$

For many problems, it is very difficult to find a solution, but relatively easy to verify a solution, and for these we create a function f that takes the proposed solution x, and returns $f(x) = 0$ if it is not a solution and 1 if it is:

$$U_\omega |x\rangle = (-1)^{f(x)} |x\rangle, \quad \text{where } f(x = \omega) = 1 \text{ and } f(x) \neq \omega = 0 \tag{2}$$

which looks very similar to the Deutsch-Jozsa oracle as well! Even looking at the circuit diagram is similar



III. AMPLITUDE AMPLIFICATION

Here's the new part compared to the DJ alg. Since we don't know where the item of interest is in the list, any guess at it's location is equally good. The superposition of this

unknown state is

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \tag{3}$$

where it's 2^n instead of N because quantum mechanically, each list location is a qubit and a qubit has two equally likely states. Classically, on average we would need to try $N/2 = 2^{n-1}$ items in the list to find the one we're looking for.

Amplitude amp enhances the amplitude of the marked item (which already has the opposite phase of all the other ones due to the first oracle). It does this with the use of a second reflection/rotation, but what it reflects over is the important part.

If we start from the beginning, we see that there are two vectors that are important to define: the initial superposition of where the item could be $|s\rangle = H^{\otimes n} |o\rangle^{\otimes n}$, and the "winning" vector that matches the item we are looking for, $|\omega\rangle$. From these, there is an additional state that we need to define, $|s'\rangle$. It happens to sit in the same 2D plane of $|s\rangle$ and $|\omega\rangle$ and it is specifically perpendicular to $|\omega\rangle$. From here, you can see that the state $|s\rangle$ sits somewhere in between $|\omega\rangle$ and $|s'\rangle$.

$$|s\rangle \equiv \sin\theta |\omega\rangle + \cos\theta |s'\rangle \tag{4}$$

This angle, θ, can be defined as

$$\theta \equiv \arcsin\langle s|\omega\rangle = \arcsin\left(\frac{1}{\sqrt{2^n}}\right) \tag{5}$$

Afterwards, we apply the first reflection, the oracle talked about in the previous section, U_ω, which will apply a negative phase to the winner state, $|\omega\rangle$.

The second reflection is roughly about the state $|s\rangle$ and can be defined as:

$$U_s = 2 |s\rangle \langle s| - 1 \tag{6}$$

This looks weird, but all you have to do is apply a negative phase to all states orthogonal to $|s\rangle$. If we were to do it step by step, it would symbolically look like this:

$$\begin{aligned} |s\rangle &\equiv \frac{1}{\sqrt{2^n}} |\omega\rangle + \sqrt{1 - \frac{1}{2^n}} |s'\rangle \\ |s_1\rangle \equiv U_f |s\rangle &= -\frac{1}{\sqrt{2^n}} |\omega\rangle + \sqrt{1 - \frac{1}{2^n}} |s'\rangle \\ |s_G\rangle \equiv U_s U_f |s\rangle &= (2 |s\rangle \langle s|s_1\rangle - |s_1\rangle) \end{aligned} \tag{7}$$

There is one inner product we need to figure out before we continue:

$$\begin{aligned}\langle s|s_1\rangle &\equiv \left(\frac{1}{\sqrt{2^n}}\langle\omega| + \sqrt{1-\frac{1}{2^n}}\langle s'|\right)\left(-\frac{1}{\sqrt{2^n}}|\omega\rangle + \sqrt{1-\frac{1}{2^n}}|s'\rangle\right) \\ &= 1 - \frac{2}{2^n} \end{aligned} \tag{8}$$

Therefore, the final state after both rotations looks like:

$$\begin{aligned}|s_G\rangle &= 2\left(1-\frac{2}{2^n}\right)|s\rangle - |s_1\rangle \\ &= \left(2-\frac{4}{2^n}\right)\left(\frac{1}{\sqrt{2^n}}|\omega\rangle + \sqrt{1-\frac{1}{2^n}}|s'\rangle\right) - \left(-\frac{1}{\sqrt{2^n}}|\omega\rangle + \sqrt{1-\frac{1}{2^n}}|s'\rangle\right) \\ &= \left(3-\frac{4}{2^n}\right)\frac{1}{\sqrt{2^n}}|\omega\rangle + \left(1-\frac{4}{2^n}\right)\sqrt{1-\frac{1}{2^n}}|s'\rangle \end{aligned} \tag{9}$$

We can see that the normalization function of the winner state is different from all states perpendicular to it. It should also be noted that these two gate need to be repeatedly applied as the value $N = 2^n$ grows. We can see that for $N = 4$ (or $n = 2$), so we'd only need 2 qubits, the combination of $U_\omega U_s$ only need to be applied a single time. This can clearly be seen as the factor in front of $|s'\rangle$ is 0 for $n = 2$.

Whereas, for $n = 3$ the number of times isn't so obvious. After a single time, neither coefficient is zero, and they are actually pretty close, I think one is $\frac{5}{4\sqrt{8}}$ and the other is $\frac{\sqrt{7}}{4\sqrt{8}}$, which is still different, but with error the answer could skew from the correct answer. You may need to apply them 3 times as in this case $\sqrt{8} \approx 3$. We can actually solve for the number of times it'll take generically.

Say after t steps we'd have:

$$(U_s U_\omega)^t |s\rangle = \sin\theta_t |\omega\rangle + \cos\theta_t |s'\rangle$$

where

$$\theta_t = (2t+1)\theta \tag{10}$$

and remembering that $\theta = \arcsin\left(\frac{1}{\sqrt{2^n}}\right)$ and requiring $\theta_t = \frac{\pi}{2}$, we find the relation

$$\begin{aligned}(2t+1)\arcsin\left(\frac{1}{\sqrt{2^n}}\right) &= \frac{\pi}{2} \\ \rightarrow t &= \frac{\pi}{4\arcsin\left(\frac{1}{\sqrt{2^n}}\right)} - \frac{1}{2}\end{aligned} \tag{11}$$

So for $n = 2 \rightarrow t = 1$, but for $n = 3 \rightarrow t \approx \frac{\pi}{4(0.361)} - \frac{1}{2} \approx 1.67$ which means we need to run it 2 times. The number of times you'd have to run it probably get closer to $\sqrt{2^n}$ the larger n is.

IV. DIFFUSER

What I find really interesting about the diffuser is that on the Qiskit website textbook, it says the 2 qubit diffuser looks like this:

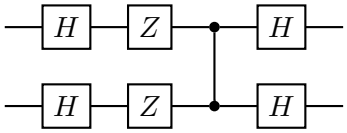


FIG. 1: 2 qubit diffuser found on qiskit's website: https://qiskit.org/textbook/ch-algorithms/grover.html

which would lead to mathematically,

$$\begin{aligned} U_{s,2A} &= (H \otimes H)(Z \otimes Z)(|0\rangle \langle 0| \otimes I + |1\rangle \langle 1| \otimes Z)(H \otimes H) \\ &= (H \otimes H)(|0\rangle \langle 0| \otimes Z - |1\rangle \langle 1| \otimes I)(H \otimes H) \\ &= (|+\rangle \langle +| \otimes X - |-\rangle \langle -| \otimes I) \end{aligned} \tag{12}$$

but for a 3 qubit circuit, the same location has diffuser has the following gates:

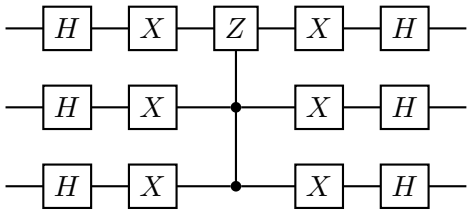


FIG. 2: generic 3 qubit diffuser found on qiskit's website: https://qiskit.org/textbook/ch-algorithms/grover.html

and if I wanted to convert the above generic diffuser to 2 qubits, it would look like this

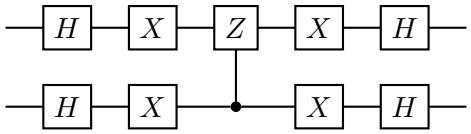


FIG. 3: generic 2 qubit diffuser converted from the generic 3 qubit diffuser found on qiskit's website: https://qiskit.org/textbook/ch-algorithms/grover.html

and mathematically, this would be the following

$$\begin{aligned} U_{s,2B} &= (H \otimes H)(X \otimes X)(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z)(X \otimes X)(H \otimes H) \\ &= (H \otimes H)(|1\rangle\langle 1| \otimes I - |0\rangle\langle 0| \otimes Z)(H \otimes H) \\ &= (|-\rangle\langle -| \otimes I - |+\rangle\langle +| \otimes X) \end{aligned} \tag{13}$$

where in the second line, we used the fact that $XZX = -Z$. Comparing $U_{s,2A}$ to $U_{s,2B}$, you see that they have a difference by an overall negative phase, $U_{s,2A} = (-1)U_{s,2B}$.

Initially, I thought a phase difference of -1 showed either some typo in either my or qiskit's work. However, while reading the section on "Creating a General Diffuser", they state that we can "ignore a global phase of -1" [for the diffuser section of the circuit]. This means that the phase difference we found was something that can be ignored.