Chapter 4: Quantum Algorithms

Jessica John Britto
Indian Institute of Technology Kharagpur

June 20, 2022

1 Introduction

Circuit complexity is defined as the least number of quantum gates that are needed for implementing a quantum algorithm w.r.t universal quantum gates. Query complexity is the number of times a function is being called to solve a particular problem, and we can define bounds for the same. Quantum oracle is similar to a black box which takes in the inputs and gives the corresponding outputs. Upon measurement of the output, we will get a linear combination of states corresponding to the function *f(x)*.

In a **Phase oracle**, the inputs acquire a phase change of $(-1)^{f(x)}$ while the ancilla qubits remain the same. This is called *phase kickback*. For example, after applying a phase oracle (U_f) on the input qubit - $\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$, we will get $(-1)^{f(0)}\frac{\sqrt{3}}{2}|0\rangle + (-1)^{f(1)}\frac{1}{2}|1\rangle$. For design of quantum algorithms, we need a quantum register, oracle, measurement.

2 Simple Aglorithms

In **Deutsch's Algorithm**, using quantum algorithm, only one query is required to determine if a univariate function *f(x)* is constant or balanced unlike its classical counterpart where two queries are needed. The function $f : \{0,1\} \rightarrow \{0,1\}$ can be one of the following possibilities:

1. $f(0) = f(1) = 1$ which is a constant function.
2. $f(0) = f(1) = 0$ which is a constant function.
3. $f(0) = 0, f(1) = 1$ which is a balanced function.
4. $f(0) = 1, f(1) = 0$ which is a balanced function.

Using classical algorithm, it can be done in the following way using *if-else* statements.

```
if f(0) = 0:
   if f(1) = 0:
      print("Constant")
   else:
      print("Balanced")
else:
   if f(1) = 0:
      print("Balanced")
   else:
      print("Constant")
```

Using quantum algorithm, we can tackle this issue for a univariate function in one query although we cannot determine the value of the given function. Let the input register be initialised to $|0\rangle$ while ancilla register be initialised to $|1\rangle$ This can be done in the following steps:

1. Apply Hadamard gate to both input and output registers each of a single qubit. This transforms to -
$|x\rangle\,|y\rangle \rightarrow \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle - |10\rangle - |11\rangle)$

2. Upon applying U_f on both the registers, we get $|y\rangle$ as $|y \oplus f(x)\rangle$ and the input register qubits remain the same. If y takes the value of 0, $|y \oplus f(x)\rangle$ transforms to $|f(x)\rangle$, else we get complement of $f(x)$.

   Therefore, $|y\rangle$ transforms to
   $\frac{1}{\sqrt{2}}(|0, f(0)\rangle + |0, \neg f(0)\rangle - |1, f(1)\rangle - |1, \neg f(1)\rangle)$

3. If $f(0) = f(1)$, then we will get $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)(|f(0) - \neg f(0)\rangle)$, else we will get $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)(|f(0) - \neg f(0)\rangle)$. Then apply the hadamard gate only to the input register.

4. Upon measurement of the first register, if we get $|0\rangle$, then f is a constant function, else $|1\rangle$, then f is a balanced function. Measuring of the ancilla register will not give us the value of f.

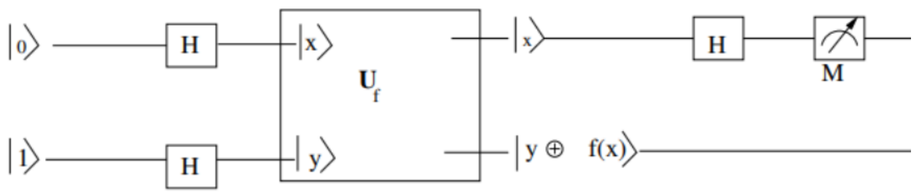


Figure 1: Circuit Diagram of Deutsch's Algorithm

Deutsch's Algorithm is useful for univariate functions, let us tackle the same issue for a mutlivariate function $f : \{0,1\}^n \rightarrow \{0,1\}$. An algorithm for this is known as **Deutsch-Jozsa Algorithm**.

In **Deutsch-Jozsa Algorithm**, the input register consist of n-qubits initialised to $|0\rangle^{\otimes n}$ and ancilla register as $|1\rangle$. We follow the same procedure as that of the Deutsch's Algorithm. On doing so, we write a general form for the input register n-qubits after applying Hadamard gate on them, as -

$$\frac{1}{2^{\frac{n}{2}}} \sum_{j=0}^{2^n-1} |j\rangle$$

where $|j\rangle = |j_{n-1} .....j_0\rangle$.

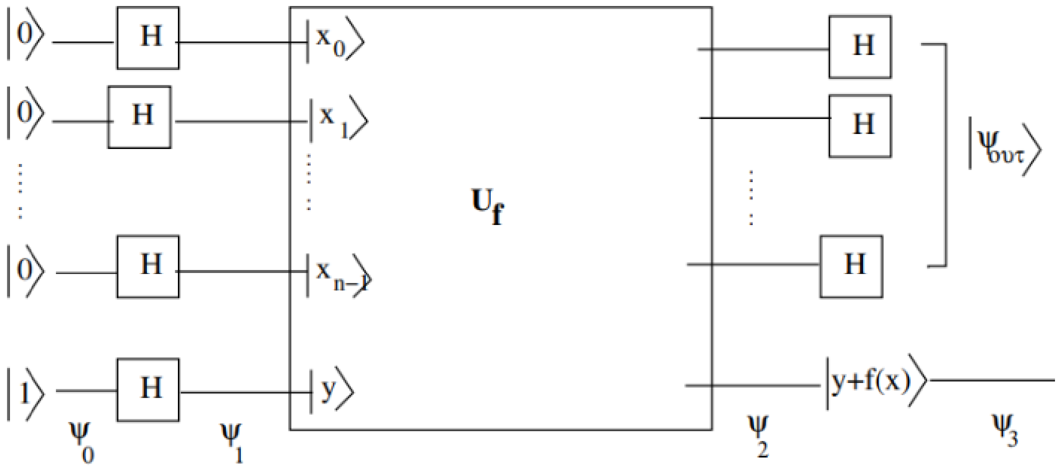


Figure 2: Circuit Diagram of Deutsch-Joza's Algorithm

Now, applying the oracle, we get the state to be transformed as

$$\frac{1}{2^{\frac{n}{2}}} \sum_{j=0}^{2^n-1} |j\rangle \, |y \oplus f(j)\rangle$$

Upon simplifying this further, we get it as

$$\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j, f(j) - \neg f(j)\rangle$$

$$\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)} \, |j\rangle \, \frac{1}{\sqrt{2}} \, |0\rangle - |1\rangle$$

On applying the Hadamard gate on the input n-qubit registers, we will get -

$$H^{\otimes n} \, |j\rangle = \frac{1}{2^{\frac{n}{2}}} \sum_{k=\{0,1\}^n} (-1)^{k.j} \, |k\rangle$$

where $k \cdot j$ is $k_0 j_0 + ...k_{n-1} j_{n-1} \ (mod\ 2)$

The state becomes

$$\frac{1}{2^n} \sum_{j=0}^{2^n-1} \sum_{k=\{0,1\}^n}^{2^n-1} (-1)^{k.j+f(j)} \, |k\rangle \, |-\rangle$$

In case, the function is balanced, then the amplitude for the case $|k\rangle = |0\rangle^{\otimes n}$ will be zero, then it means that when we measure the input registers we will get any state other than $|00...0\rangle$. If the function is constant, then the amplitude for the state $|00..0\rangle$ will be one.

In case of **Bernstein-Vazirani Problem**, we need to determine the value of a for a function $f = a \cdot x$. The algorithm is same as that of Deutsch-Jozsa Algorithm. The final state before measurement is the following -

$$\frac{1}{2^n} \sum_{j=0}^{2^n-1} \sum_{k=\{0,1\}^n}^{2^n-1} (-1)^{(k+a)\cdot j} |k\rangle |-\rangle$$

Only for the case when $a = k$, the probability amplitude of the state will be one, which means for all other states, the probability amplitude will be zero. Hence, we can find the value of the unknown string a. To solve this problem using classical algorithm, we will need to call the function n times where n is the length of the unknown string a using a given input x such that $f(100..00) = a_{n-1}$ and so on.

3 Simon's Algorithm

In **Simon's problem**, we need to find the value of a non-zero unknown string s. Let a function f be $f : \{0,1\}^n \to \{0,1\}^n$ such that $f(x) = f(y)$ if $x = y \oplus s$. If the function is one-to-one, we will get s as $|00..0\rangle$, and if the function is two-to-one, then we will get a non-zero value for s. Let there be two registers consist of n-qubits each. We apply Hadamard gate on the first register and then apply the query function, then we will get the state as

$$\frac{1}{2^{\frac{n}{2}}} \sum_{x=\{0,1\}^n} |x\rangle |f(x)\rangle$$

On measuring the second register, we will get a certain value of $f(x)$, then the first register is reduced to the linear combination of $|x\rangle$ and $|x \oplus s\rangle$, i.e, $\frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle)$ and this happens if f is two-to-one. The measurement of the second register causes the two registers to be entangled with each other. We then pass the first register qubits to the hadamard gate. Then we obtain the state as -

$$\frac{1}{2^{\frac{n+1}{2}}} \sum_{y=\{0,1\}^n} [(-1)^{x\cdot y} + (-1)^{(x\oplus s)\cdot y}] |y\rangle$$

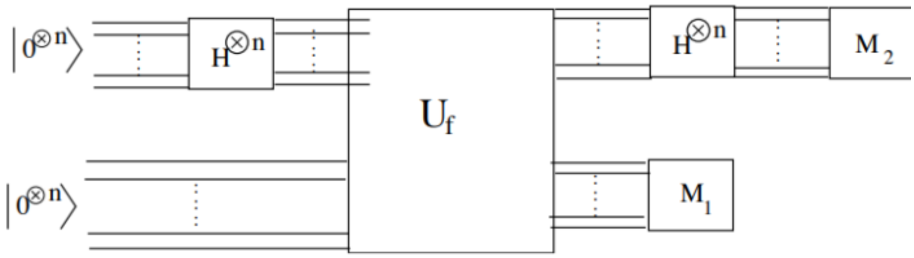


Figure 3: Circuit Diagram of Simon's Algorithm

If $(-1)^{x \cdot y} = (-1)^{(x \oplus s) \cdot y}$, we get $s \cdot y = 0$, i.e the inner product of s and y must be zero, they must be orthogonal to each other, and on measurement of the first register, we will get a value of y which satisfies this condition. In this case, the state will become -

$$\frac{1}{2^{\frac{n-1}{2}}} \sum_{y=\{0,1\}^n} [(-1)^{x \cdot y}] |y\rangle$$

and its probability amplitude is $\frac{1}{2^{n-1}}$. If $(-1)^{x \cdot y}$ is not equal to $(-1)^{(x \oplus s) \cdot y}$, then the coefficient of $|y\rangle$ will be zero and hence its probability amplitude.

$$\begin{cases} b \cdot z_1 = 0 \\ b \cdot z_2 = 0 \\ \quad \vdots \\ b \cdot z_n = 0 \end{cases}$$

To find the unknown non-zero string s, we have to run this algorithm n times to obtain linearly independent vectors such that they are orthonormal to s and also we may consider that y is not equal to zero. Upon solving each of those linearly independent vectors by taking inner product with s will give us its value as shown in the figure above.

4 Grover's Search Algorithm

This algorithm is used for finding M elements in an unstructured database in $O\sqrt{N/M}$ where M is the number of elements that are to be found in that database. The quantum oracle finds a function f for n qubit inputs and returns 1 when the input matches a particular string w, called the marked string, in all other cases it returns zero. Therefore, the function is $f : \{0,1\}^{\otimes n} \rightarrow \{0,1\}$. Let M be 1. Then, after applying the oracle, the input register will be altered whose sign depends on the function f_w such that is zero when x is not equal to w, otherwise, it is one. This can be achieved using an unitary operator - $U_w = I - 2\left|w\right\rangle\left\langle w\right|$, where $\left|w\right\rangle$ is the marked state which is orthogonal to the states associated with the remaining items in the database. Firstly, we have to create a standard state which is a superposition of all the basis states of a n-qubits system. This is obtained by applying Hadamard gate to the input register of n-qubits, which are initialised to $\left|0\right\rangle^{\otimes n}$. Clearly, the standard state has the marked state in it, so $\langle w|s\rangle = \frac{1}{\sqrt{N}}$ where $N = 2^n$. Then we need $U_s = 2\left|s\right\rangle\left\langle s\right| - I$. If any arbitrary state acts on U_s, then component of the state along s is unaltered but flips the sign of the component perpendicular to $\left|s\right\rangle$. Therefore, U_s is the reflection operator. Grover rotation operator R_G is product of the sign flip operator U_w and the reflection operator U_s, i.e, $R_G = U_s U_w$. R_g is defined in the plane of $\left|s\right\rangle$ and $\left|w\right\rangle$. The angle between these vectors is $\frac{\pi}{2} - \theta$. After applying R_g on $\left|\chi\right\rangle$, then the angle between the initial and final states is 2θ, where θ is the angle between $\left|w_\perp\right\rangle$ and $\left|s\right\rangle$. This is clearly shown in figure 4.

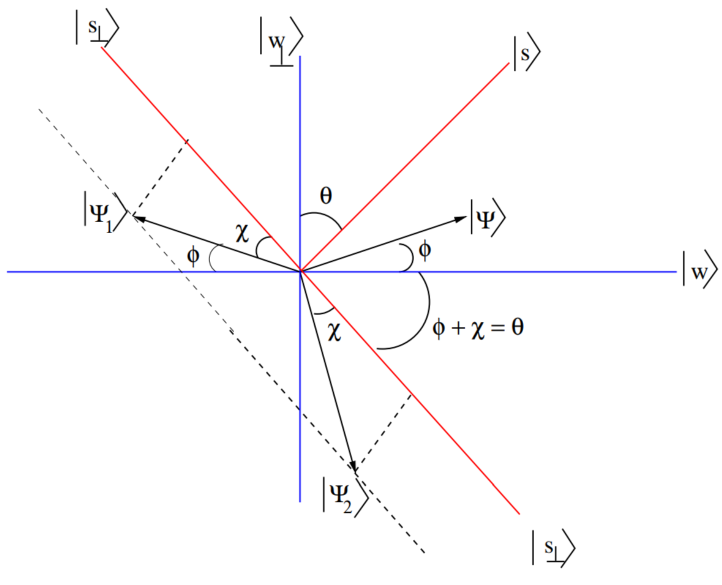
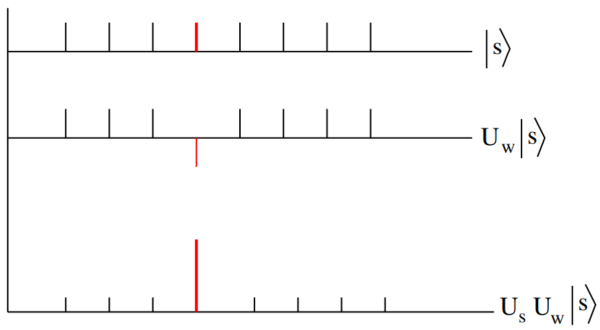


Figure 4: Grover Rotation

Clearly, with every application of R_g, the arbitrary state comes close to the marked state, however after a certain, it will go away from the marked state. Hence, it becomes important to know the number of times the grover rotation operator has to be applied to get the desired result. For example, when $N = 4$, $\theta = 30^{\circ}$, therefore, we get the marked state in one iteration, since in one iteration, the angle between initial and final states become 60°. So, if we continue iterating anymore, the arbitrary state will not be aligned with the marked state. Effect of U_s on $\chi = \sum_x a_x |x\rangle$.

$$\begin{aligned} \langle s|\chi\rangle &= \frac{1}{\sqrt{N}} \sum_x a_x = \sqrt{N}\bar{a} \\ U_s |\chi\rangle &= [2|s\rangle\langle s| - I]|\chi\rangle = 2|s\rangle\langle s|\chi\rangle - |\chi\rangle \\ U_s |\chi\rangle &= 2\sqrt{N}\bar{a}|s\rangle - |\chi\rangle = \sum_x (2\bar{a} - a_x)|x\rangle \end{aligned} \tag{1}$$

Clearly, application of the reflection operator increases the amplitude of the marked state quite higher than the remaining states. This is possible due to selective amplification as shown in the figure below.



We can find the maximum iteration needed to get the marked state using - $\sin\theta \approx \theta = \frac{1}{\sqrt{N}}$, where N is quite large and m is the number of iterations needed. For this, $m \cdot 2\theta \approx \frac{\pi}{2} - \theta$. This implies that - $m = \frac{\pi}{4\theta} - 2$, for large N, we have - $m \approx \frac{\pi}{4}\sqrt{N}$. Therefore, after m iterations, the angle between $|s\rangle$ and $|\chi\rangle$ is $\frac{\pi}{2} - (2 \cdot m + 1)\theta$ and the probability amplitude of $|w\rangle$ in $|s\rangle$ becomes the absolute value of $\sin 2 \cdot m + 1)\theta = \cos\frac{1}{\sqrt{N}} \approx 1 - \frac{1}{2 \cdot N}$. Clearly, this amplitude is close to 1. Matrix representation of Grover operator is defined as $D = -I + 2\frac{J}{N}$ where J is a N X N matrix where each element is 1, therefore D is unitary matrix since $(\frac{J}{N}) = (\frac{J}{N})^2$, i.e, its a projection operator. This can be obtained from $D = HRH$, where H is the hadamard gate and $R = (-1)^{1-\delta_{i,0}}\delta_{i,j}$.

5 References

1. Chapter 7 from "Introduction to Classical and Quantum Computing"

2. Simon's Algorithm, Simple Algorithms, Grover's Search Algorithm, Shor's Factorization Algorithm, QFT - Lecture Notes by Professor D K Ghosh
3. Lecture Videos by Professor D K Ghosh
4. Simon's Algorithm from QuIC Seminar
5. Simon's Algorithm - from Stackexchange Discussion
6. Simon's Algorithm and Simple Algorithms
7. Deutsch-Jozsa Algorithm - Nielson and Chuang
8. Simon's Algorithm - Lecture Videos by Advanced Maths
9. Grover's Search Algorithm - from Lecture: 22 by Scott Aaronson
10. Twisted Oak - Grover's Search Algorithm by Craig Gidney
11. daytonellwanger's Grover's Algorithm
12. CMU15-859BB, Fall 2015 Lecture-4 by Lecturer: John Wright
13. Quantum Computing Lecture Notes by Ronald de Wolf