

# Quantum Algorithms, Spring 2022: Lecture 6 Scribe

Rutvij Menavlikar, Abhyudit Mohla

January 25, 2022

## 1 Recap

### 1.1 Universality of Quantum Circuits

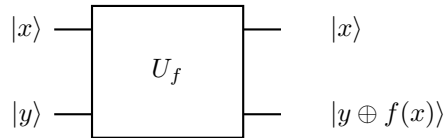
- Single qubit and  $CNOT$  gates together can be used to implement an arbitrary two-level unitary operation on the state space of  $n$  qubits.[? ]
- The set  $\{CNOT, H, R_{\frac{\pi}{4}}\}$  is universal for quantum computing, i.e. any other quantum circuit can be well approximated using quantum circuits of only these gates.  
And *Solovay-Kitaev theorem* states that any  $t$ -gate quantum circuit can be  $\epsilon$  approximated using  $\mathcal{O}(t \cdot \text{polylog}(\frac{1}{\epsilon}))$  gates from  $\{CNOT, H, R_{\frac{\pi}{4}}\}$ .

### 1.2 Quantum Parallelism

To estimate a function  $f$  s.t.

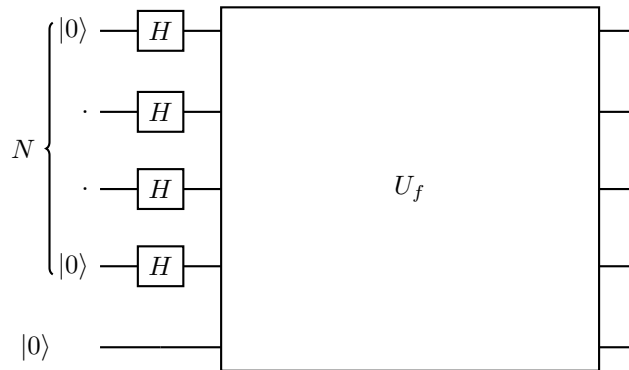
$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

we use a quantum gate  $U_f$  in the following way.



$$\begin{aligned} \text{So, if } f(x) = 0, & \quad |x\rangle |y\rangle \xrightarrow{U_f} |x\rangle |y\rangle \\ \text{if } f(x) = 1, & \quad |x\rangle |y\rangle \xrightarrow{U_f} |x\rangle |\bar{y}\rangle \end{aligned} \tag{1}$$

Now, we can parallelise this circuit, even for  $n$  qubits by adding Hadamard's gates on input qubits before applying the  $U_f$  gate in the following way.



Thus, by applying  $U_f$  only once, we are able to obtain a quantum state that contains all possible  $2^n$  values of  $f(x)$  in superposition.

$$|0\rangle^{\otimes n} |0\rangle \xrightarrow{H^{\otimes n} \otimes I} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |0\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle \quad (2)$$

Note that,

- Quantum parallelism is not enough to demonstrate the power of Quantum Computing
- Quantum parallelism needs to be combined with interference, entanglement to do something better than classical computing.

## 1.3 Deutsch Algorithm

### 1.3.1 Problem

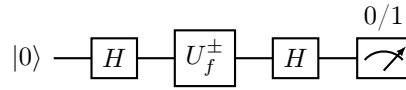
We are given a binary function  $f : \{0,1\} \rightarrow \{0,1\}$  such that either  $f(0) = f(1)$  or  $f(0) \neq f(1)$ . We have determine which using minimum number of queries.

### 1.3.2 Classical Approach

We query for value of  $f(0)$  and  $f(1)$  to determine which kind of function is  $f$ . Hence, classical approach requires 2 queries.

### 1.3.3 Quantum Algorithm

We set up the following circuit



The calculations are as follows

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \xrightarrow{U_f^\pm} \frac{1}{\sqrt{2}} \left( (-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle \right) \xrightarrow{H} \frac{1}{2} \left( \left( (-1)^{f(0)} + (-1)^{f(1)} \right) |0\rangle + \left( (-1)^{f(0)} - (-1)^{f(1)} \right) |1\rangle \right) \quad (3)$$

Thus, when  $f(0) = f(1)$  the measured state is  $|0\rangle$ , and when  $f(0) \neq f(1)$  the measured state is  $|1\rangle$ . And hence, with quantum parallelism we can solve the problem with only 1 query to  $U_f$ .

## 2 Mach-Zehnder Interferometer

In physics, the Mach-Zehnder interferometer is a device used to determine the relative phase shift variations between two collimated beams derived by splitting light from a single source.[?] Essentially, Mach-Zehnder Interferometer physically captures Deutsch's problem.

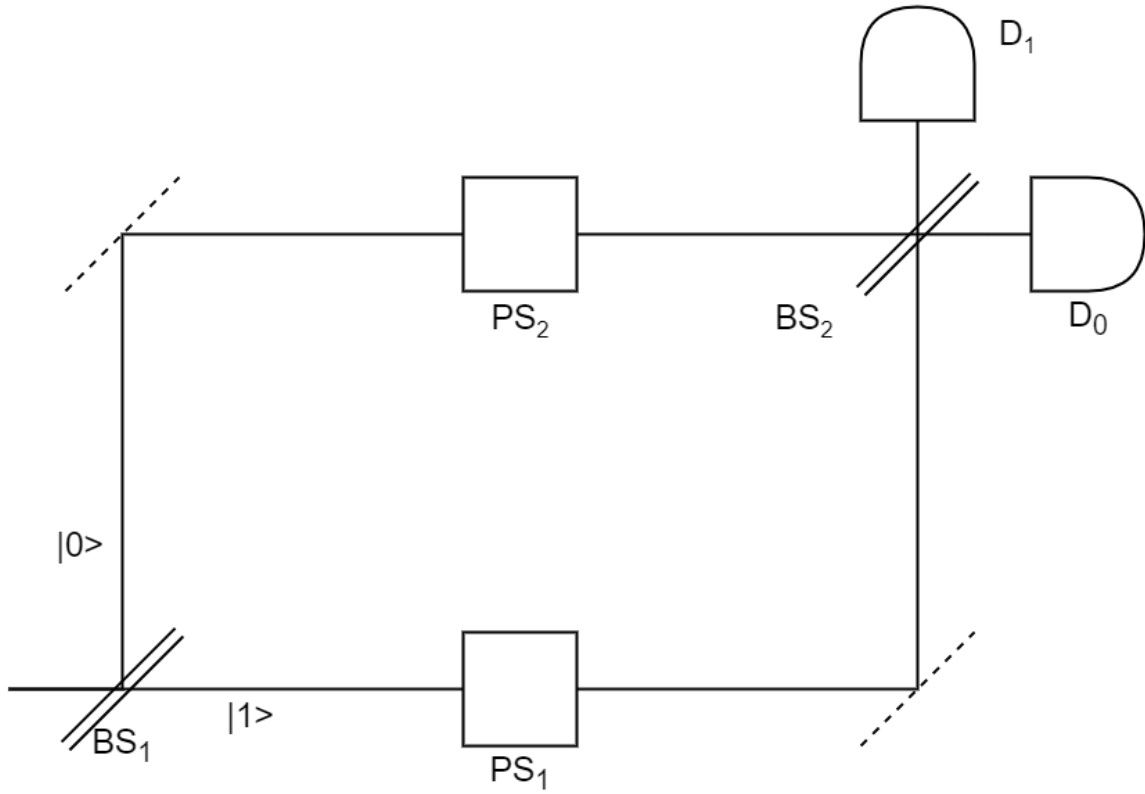


Figure 1: Mach-Zehnder Interferometer set up

The set up includes the following components

- $BS_1$  and  $BS_2$  are beam splitters. They split a photon into a superposition of *lower path* and *upper path*.
- $PS_1$  and  $PS_2$  are phase shifters.  $PS_1$  shifts the phase of the beam incident on it by  $\phi_1$  and  $PS_2$  shifts it by  $\phi_0$ .
- $D_0$  and  $D_1$  are detectors.  $D_0$  detects a photon in *lower path* and  $D_1$  detects a photon in *upper path*.

These components are arranged as shown in fig 1.

Here we assume that *lower path* is represented by  $|0\rangle$  and *upper path* is represented by  $|1\rangle$ . Thus,  $|0\rangle$  undergoes phase shift of  $\phi_0$  and  $|1\rangle$  undergoes phase shift of  $\phi_1$ . This can be represented as

$$|0\rangle \xrightarrow{PS_2} e^{i\phi_0} |0\rangle, |1\rangle \xrightarrow{PS_1} e^{i\phi_1} |1\rangle$$

Also, it is given that  $|\phi_0 - \phi_1| = 0$  or  $\pi$  and we have to determine which.

Thus, the calculations are as follows:

$$|0\rangle \xrightarrow{BS_1} \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \xrightarrow{PS} \frac{1}{\sqrt{2}} (e^{i\phi_0} |0\rangle + e^{i\phi_1} |1\rangle) \xrightarrow{BS_2} \frac{1}{2} ((e^{i\phi_0} + e^{i\phi_1}) |0\rangle + (e^{i\phi_0} - e^{i\phi_1}) |1\rangle) \quad (4)$$

But we can see that

$$\frac{1}{2} ((e^{i\phi_0} + e^{i\phi_1}) |0\rangle + (e^{i\phi_0} - e^{i\phi_1}) |1\rangle) = \frac{e^{i\phi_0}}{2} \left( (1 + e^{i(\phi_1 - \phi_0)}) |0\rangle + (1 - 1 + e^{i(\phi_1 - \phi_0)}) |1\rangle \right) \quad (5)$$

Thus,

- When  $|\phi_0 - \phi_1| = 0$ ,  $|0\rangle$  is observed, i.e.  $D_0$  detects a photon.
- When  $|\phi_0 - \phi_1| = \pi$ ,  $|1\rangle$  is observed, i.e.  $D_1$  detects a photon.

*Note that:* This is a quantum phenomena. Classically, each of the detector would detect a photon 50% of the times.

### 3 Deutsch-Jozsa Algorithm

This algorithm is a generalisation of Deutsch problem on  $n$  qubits.

#### 3.1 Problem

We are given a black box  $U_f$  for some Boolean function  $f : \{0,1\}^n \rightarrow \{0,1\}$  where it is given that  $f$  is either *constant* or *balanced*, i.e

$$\text{Either, } f(x) = \text{constant: } \forall x \in \{0,1\}^n, f(x) = e \text{ s.t. } e \in \{0,1\}$$

$$f(x) = \text{balanced: } \begin{cases} 0, & \text{for } 2^{n-1} \text{ values of } x \\ 1, & \text{for rest of the } 2^{n-1} \text{ values of } x \end{cases}$$

We have to determine which using minimum number of queries.

#### 3.2 Classical Approach

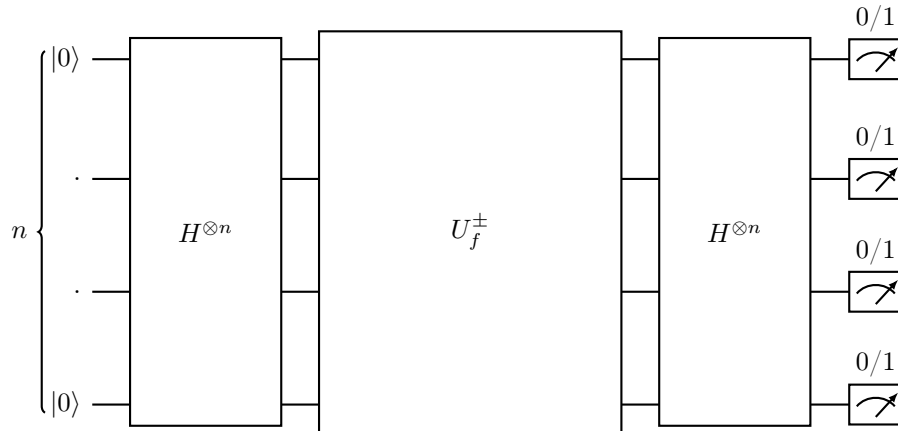
Serially, or randomly we input values of  $x \in \{0,1\}^n$  and check their outputs.

- If  $f$  is *balanced*:  
If we observe 0 and 1 in output, then  $f$  is *balanced*. And by Pigeonhole principle, we would require at-most  $2^{n-1} + 1$  queries to reach this conclusion.
- If  $f$  is *constant*:  
If we observe the same output after  $2^{n-1} + 1$  queries, we can conclude that  $f$  is *constant*.

Hence, number of queries required are  $2^{n-1} + 1$ .

#### 3.3 Quantum Algorithm

We set up the following circuit



This leads to the following changes,

$$|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \xrightarrow{U_f^\pm} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle \xrightarrow{H^{\otimes n}} \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \sum_{z \in \{0,1\}^n} (-1)^{f(x)+x \cdot z} |z\rangle \quad (6)$$

Let  $|\psi_f\rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \sum_{z \in \{0,1\}^n} (-1)^{f(x)+x \cdot z} |z\rangle$

Then, we look at the amplitude of  $|0\rangle^{\otimes n}$  in  $|\psi_f\rangle$ ,

$$\langle 00..0 | \psi_f \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \sum_{z \in \{0,1\}^n} (-1)^{f(x)+x \cdot z} \langle 00..0 | z \rangle \quad (7)$$

$$= \frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} \quad \langle 00..0 | z \rangle = \begin{cases} 1, & z = |0\rangle^{\otimes n} \\ 0, & \text{otherwise} \end{cases}, x \cdot 0 = 0 \quad (8)$$

Thus,

- If  $f$  is *balanced*:  
 $\langle 00..0 | \psi_f \rangle = 0$
- If  $f$  is *constant*:  
 $\langle 00..0 | \psi_f \rangle = 1$  or  $\langle 00..0 | \psi_f \rangle = -1$

Thus, if the result of measurement is the state  $|0\rangle^{\otimes n}$ , then  $f$  is *constant*, and for any other result of measurement  $f$  is *balanced*.

Hence, only 1 query is required to classify the function. And hence, there is an exponential optimization in query complexity.

### 3.4 Randomized Deutsch-Jozsa Algorithm

Now, we require our algorithm to classify  $f$  with a probability  $\geq 1 - \epsilon$  ( $\epsilon > 0$ )

#### 3.4.1 Classical Approach

We choose  $d$  values of  $x \in \{0,1\}^n$  to query.

$$\text{Let values of } x: S = \{x_1, x_2, \dots, x_d\} \quad x_i \in \{0,1\}^n$$

Thus, by making the  $d$  queries, we get  $f(S) = \{f(x_1), f(x_2), \dots, f(x_d)\}$  Then the following scenarios are possible,

- *Case 1*:  $\forall x_i \in S, f(x_i) = 0$  or  $\forall x_i \in S, f(x_i) = 1$   
In this case, if  $d < 2^{n-1} + 1$ , then we still cannot be sure if  $f$  is *balanced* or *constant*. And hence, there is an uncertainty in classification. But the probability of observing this case is  $\frac{1}{2^d} + \frac{1}{2^d} = \frac{1}{2^{d-1}}$
- *Case 2*:  $\exists x_i, x_j \in S$ , s.t.  $f(x_i) \neq f(x_j)$   
In this case, we can definitely conclude that  $f$  is *balanced*.

Thus, we can choose  $d$  in a way that we reduce the probability of observing the case where we are not certain about the outcome. That is,

$$\frac{1}{2^{d-1}} < 1 - (1 - \epsilon) \Rightarrow d > \log_2 \left( \frac{2}{\epsilon} \right) \quad (9)$$

Hence, the query complexity for the classical approach to the randomised Deutsch-Jozsa Problem is  $\mathcal{O} \log \left( \frac{1}{\epsilon} \right)$

## 4 Bernstein-Vazirani Algorithm

A related problem was given by Ethan Bernstein and Umesh Vazirani.

## 4.1 Problem

We are given a black box  $U_f$  for some Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  where it is given that  $\forall x \in \{0, 1\}^n f(x) = s \cdot x \pmod{2}$  for some unknown string  $s \in \{0, 1\}^n$ . i.e.,

$$\forall x \in \{0, 1\}^n f(x) = s_1 x_1 \oplus s_2 x_2 \oplus \dots \oplus s_n x_n$$

We have to determine  $s$  using minimum number of queries.

## 4.2 Classical Approach

We pass a set of inputs  $I = \{x_1, x_2, \dots, x_n\}$  such that

$$\forall x_i \in I, x_i(j) = \begin{cases} 1, & j = i \\ 0, & \text{otherwise} \end{cases}$$

Thus, the output of each input  $x_i$  looks like

$$f(x_i) = s_1 x_i(1) \oplus s_2 x_i(2) \oplus \dots \oplus s_n x_i(n) \quad (10)$$

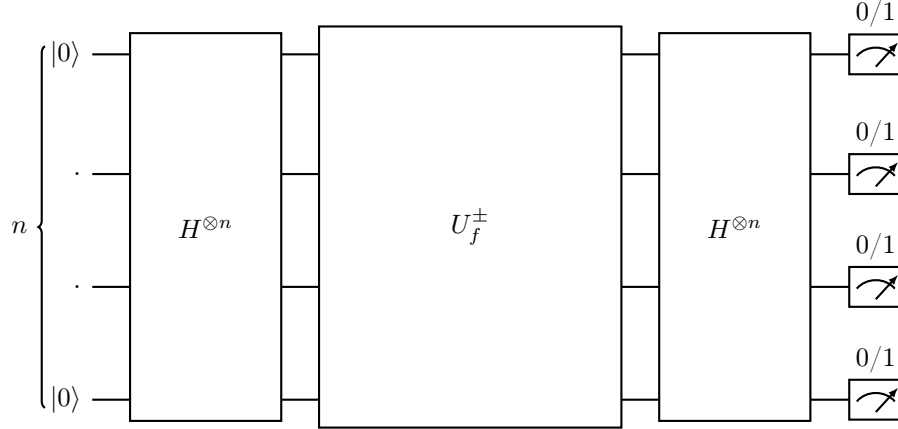
$$= 0 \oplus 0 \oplus \dots \oplus 0 \oplus s_i \oplus 0 \oplus \dots \oplus 0 \quad (11)$$

$$= s_i \quad (12)$$

Hence,  $n$  queries are required to determine  $s$ .

## 4.3 Quantum Algorithm

We set up the circuit in the same way as we did for Deutsch-Jozsa Algorithm.



The calculations are as follows,

$$|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \xrightarrow{U_f^{\pm}} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle \quad (13)$$

and

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot s \pmod{2}} |x\rangle \quad (14)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot s} |x\rangle \quad (15)$$

But note that  $|s\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot s} |x\rangle$  and  $H^{\otimes n} \times H^{\otimes n} = \mathbb{I}_n \Rightarrow (H^{\otimes n})^{-1} = H^{\otimes n}$   
Thus,

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle \xrightarrow{H^{\otimes n}} |s\rangle \quad (16)$$

Hence, the measured output is the string  $s$ .

And hence, we are able to determine  $s$  in 1 query, which gives us a linear speed up in query complexity.