Quantum Computing
Lecture 6

Anuj Dawar

Quantum Searching

Search Problems

One of the two most important algorithms in quantum computing is *Grover's search algorithm*—first presented by Lov Grover in 1996.

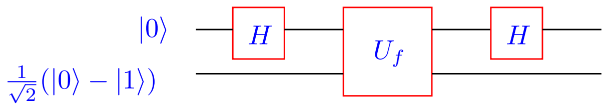
This provides a means of searching for a particular value in an *unstructured search space*.

Compare

- searching for a name in a telephone directory
- searching for a phone number in a telephone directory

Given a black box which can take any of N inputs, and for each of them gives a yes/no answer, Grover's algorithm allows us to find the unique value for which the answer is yes in $O(\sqrt{N})$ steps (with high probability).

Deutsch-Josza Algorithm revisited

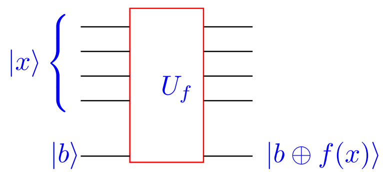


When the lower input to U_f is $|0\rangle - |1\rangle$, we can regard this as unchanged, and instead see U_f as shifting the phase of the upper qubit by $(-1)^{f(x)}$.

Oracle

Suppose we have $f : N \to \{0, 1\}$, and that $N = 2^n$, so we can think of f as operating on n bits.

We assume that we are provided a *black box* or *oracle* U_f for computing f, in the following sense:



Grover's Algorithm

Suppose further that there is exactly one n-bit value a such that

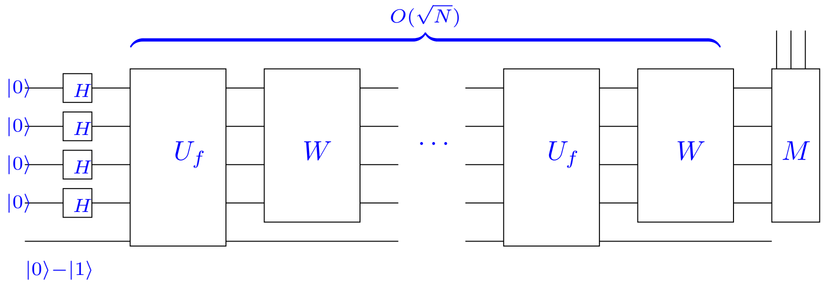
$$f(a) = 1$$

and for all other values x,

$$f(x) = 0.$$

Grover's algorithm gives us a way of using the black box U_f to determine the value a with $O(\sqrt{N}) = O(2^{n/2})$ calls to U_f.

Grover's Algorithm Schematic



The operator $G = (W \otimes I)U_f$ is known as the *Grover Iterate* (we will see soon what W is).

The input to the last bit is $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

The Action of U_f

As the "*output qubit*" is $|0\rangle - |1\rangle$, it remains unaffected by the action of U_f, which we can think of instead as a conditional phase change on the n input qubits.

$$|a\rangle \mapsto -|a\rangle$$
$$|x\rangle \mapsto |x\rangle \qquad \text{for any } x \neq a$$

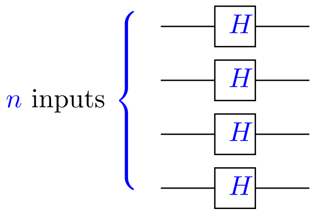
We will ignore the output bit completely and instead talk of the n-bit operator V above.

Note: $V = I - 2|a\rangle\langle a|$.

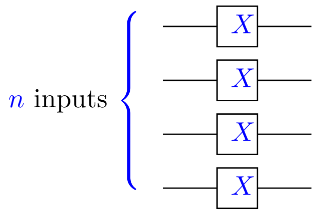
We now analyse the Grover iterate WV.

Components of W

We write $H^{\otimes n}$ for the following operation:



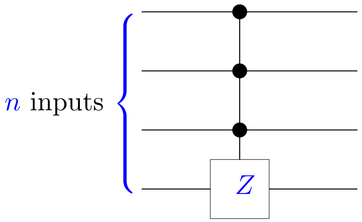
And $X^{\otimes n}$ for the following operation:



Each of these can, of course, be implemented by a series of n 1-qubit operations.

More Components of W

We write $cZ^{\otimes n}$ for the n-bit controlled-Z gate:



$$cZ^{\otimes n} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \end{bmatrix}$$

$cZ^{\otimes n}$ can be implemented using $O(n)$ cZ and *Toffoli* gates, using some workspace qubits (*Exercise*).

Defining W

Now, we can define W by:

$$\begin{aligned} W &= (-1)H^{\otimes n}(X^{\otimes n}cZ^{\otimes n}X^{\otimes n})H^{\otimes n}. \\ &= (-1)H^{\otimes n}(I - 2|0^n\rangle\langle 0^n|)H^{\otimes n} \end{aligned}$$

Write $|\Psi\rangle$ for the state

$$H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle.$$

So, $W = (-1)(I - 2|\Psi\rangle\langle\Psi|)$, i.e.

$$W = 2|\Psi\rangle\langle\Psi| - I.$$

The Grover Iterate

Since $G = WV$, we have

$$G = (2|\Psi\rangle\langle\Psi| - I)(I - 2|a\rangle\langle a|).$$

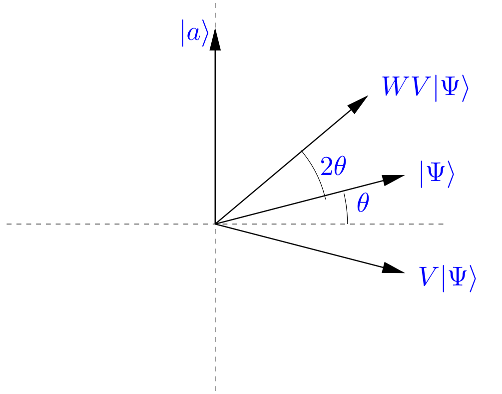
Consider the actions of W and V on the two states $|\Psi\rangle$ and $|a\rangle$.

$$\begin{aligned} W|\Psi\rangle &= |\Psi\rangle & \qquad W|a\rangle &= \tfrac{2}{\sqrt{N}}|\Psi\rangle - |a\rangle. \\ V|\Psi\rangle &= |\Psi\rangle - \tfrac{2}{\sqrt{N}}|a\rangle & \qquad V|a\rangle &= -|a\rangle \end{aligned}$$

Thus, as we start the algorithm in state $|\Psi\rangle$, the result of repeated applications of V and W will always give a *real* linear combination of $|a\rangle$ and $|\Psi\rangle$.

Geometric View

We can picture the action of W and V in the two-dimensional real plane spanned by the vectors $|a\rangle$ and $|\Psi\rangle$.



V is a *reflection* about the line perpendicular to $|a\rangle$.
W is a *reflection* about $|\Psi\rangle$.
The composition of two reflections of the plane is always a *rotation*.

The Rotation

It is clear from the picture that WV (the Grover iterate) is a rotation through an angle 2θ in the direction from $|\Psi\rangle$ to $|a\rangle$, where the angle between $|\Psi\rangle$ and $|a\rangle$ is $\frac{\pi}{2} - \theta$.

$|\Psi\rangle$ and $|a\rangle$ are *nearly orthogonal*, so θ is small (if N is large).

$$\sin\theta = \cos(\frac{\pi}{2} - \theta) = \langle a|\Psi\rangle = \frac{1}{\sqrt{N}} = \frac{1}{2^{n/2}}.$$

So,

$$\theta \sim \frac{1}{\sqrt{N}} = \frac{1}{2^{n/2}}$$

for large enough values of N.

Number of Iterations

After $t \sim \frac{\pi/2}{2\theta} \sim \frac{\pi}{4}\sqrt{N}$ iterations of the Grover iterate $G = WV$, the state of the system

$$G^t|\Psi\rangle$$

is within an angle θ of $|a\rangle$.

A measurement at this stage yields the state $|a\rangle$ with probability

$$|\langle G^t\Psi|a\rangle|^2 \geq (\cos\theta)^2 = 1 - (\sin\theta)^2 = \frac{N-1}{N}.$$

Note: Further iterations beyond t will *reduce* the probability of finding $|a\rangle$.

Multiple Solutions

Grover's algorithm works even if the solution $|a\rangle$ is not unique.

Suppose there is a set of solutions $S \subseteq \{0, \ldots, N-1\}$ and let $M = |S|$ be the number of solutions.

The Grover iterate is then a rotation in the space spanned by the two vectors

$$|\Psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \qquad |S\rangle = \frac{1}{\sqrt{M}} \sum_{j\in S} |j\rangle$$

As the angle between these is smaller, the number of iterations drops, but so does the probability of success.

Lower Bound

For classical algorithms, searching an unstructured space of solutions (such as given by a black box for f), it is easy to show a $\Omega(N)$ lower bound on the number of calls to the black box required to identify the unique solution.

Grover's algorithm demonstrates that a quantum algorithm can beat *any* classical algorithm for the problem.

It is possible to show a $\Omega(\sqrt{N})$ lower bound for the number of calls to U_f by *any* quantum algorithm that identifies a unique solution.

Grover's algorithm does not allow quantum computers to solve NP-complete problems in polynomial time.