G53NSC and G54NSC
Non-Standard Computation

Dr. Alexander S. Green

2nd of March 2010

Introduction

- Thank you for the feedback last week...
- 2 main points were brought to my attention:
- Lecture content:
  - I will try and slow down for the complicated bits
  - Feel free to interrupt with questions
- Portfolio exercises:
  - No exercise sheet this week (but labs as usual)
  - Final exercise sheet will be released next week...
  - Will be involved, but lots of time until deadline (1st of April)
  - Feel free to email me with queries

Introduction

- Last week we looked at some of the more simple quantum algorithms
- Superdense coding
- Quantum teleportation
- both make use of entanglement as a resource to achieve *unclassical* results
- We started to look at Deutsch's algorithm...
- and mentioned Deutsch-Jozsa
- but didn't finish covering them, so lets get back to it!
- What about today?

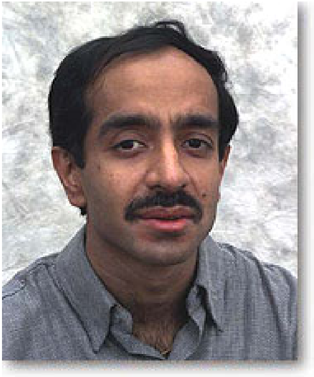
Introduction

- We're going to move on to two of the more famous quantum algorithms...
  - Grover's algorithm
  - Shor's algorithm
- We'll cover Grover's algorithm today
- and start looking at Shor's algorithm next week

Part I

Grover's Algorithm

Grover's Algorithm



Lov Grover

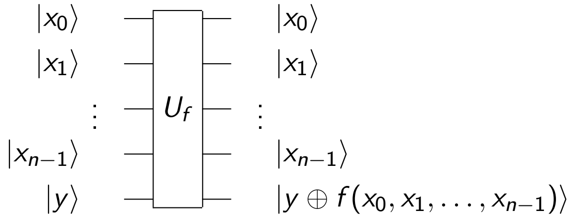
- A computer scientist working for Bell labs
- came up with his algorithm in 1996
- Often desribed as an algorithm for searching an unsorted database
- It provides a quadratic speedup over the fastest classical solution
- $O(\sqrt{N})$ compared to $O(N)$

The classical problem

- You have a large unsorted database (with N distinct elements)
- You want to find a specific element a in the database
- How can you go about finding the element a?
- The best solution classically is to look at each element in the database and see if it is a
- On average you will have to look through $\frac{N}{2} + 1$ elements
- Lets reformulate the problem slightly...
- You're given a Boolean function f with a domain of size $N = 2^n$, that only returns *True* for one element a
- On average, how many times must you call this function before finding the element a?

Grover's Algorithm

- What if we can apply this function to a quantum state?
- E.g. if we have a unitary U_f:



- How many times must we apply this unitary before finding $|a\rangle$?
- Using Grover's algorithm, we only need to apply it $\frac{\pi}{4}\sqrt{N}$ times

Searching

- Not an exponential speed up, but for large N any speed up is good!
- But, is searching an unsorted database really that useful?
- and, does the database need to be *quantum* in some way?
- Grover's algorithm has other uses...
- It can be used to find solutions to any problem that can be re-expressed as a searching problem
- So, can be used to help find solutions to NP-Complete problems
- These are problems which are believed to be unfeasible on classical computers, but whose solutions can be verified efficiently

Searching

- E.g. the travelling salesman problem
- Given a list of cities, the pairwise distances between them, and a tour around them, does a tour exist that is shorter than the given one?
- A brute force solution would be to search every permutation for a shorter one
- So, this can be treated as a searching problem
- and Grover's Algorithm could give us a speed-up over the fastest classical solution

Grover's Algorithm

- Lets look at how Grover's algorithm works
- It is nice to think of what it is doing geometrically...
- and is often presented in this manner
- The first thing we should look at, is what happens if the last input to the unitary U_f is in the state $|-\rangle$
- with an arbitrary state $|x\rangle = |x_0, x_1, \ldots, x_{n-1}\rangle$ as the rest of the input
- The entire input state can be thought of as $|x\rangle \otimes |-\rangle$
- and the output state will be...
- $(-1)^{f(x)} |x\rangle \otimes |-\rangle$

Grover's Algorithm

- The last qubit is unchanged, and the component of $|x\rangle$ that we're looking for has had a negative phase added to it
- As the last qubit is unchanged, we can ignore it...
- defining the unitary operator V as having the behaviour described above
- $V|x\rangle = (-1)^{f(x)}|x\rangle = \begin{cases} |x\rangle, & x \neq a \\ -|a\rangle, & x = a \end{cases}$
- Grover's Algorithm only requires one other unitary, which we shall denote W
- In fact, W is quite similar to V, but doesn't depend on the search function f
- $W|x\rangle = \begin{cases} |\phi\rangle, & x = \phi \\ -|x\rangle, & x \neq \phi \end{cases}$
- where $|\phi\rangle$ is an equal super-position of n qubits
- $|\phi\rangle = H^{\otimes n}|0\rangle = \frac{1}{2^{\frac{n}{2}}} \sum_{x=0}^{2^n-1} |x\rangle$

Grover's Algorithm

- It may be easier to think of W in terms of $-W$, which would only effect the outcome by a possible negative phase, and hence not the measurement
- $-W$ can be defined more easily using the computational basis
- $-W = H^{\otimes n} W' H^{\otimes n}$ where
  $$W' \left|x\right\rangle = (-1)^{x \equiv 0} \left|x\right\rangle = \begin{cases} \left|x\right\rangle, & x \neq 0 \\ -\left|0\right\rangle, & x = 0 \end{cases}$$
- giving the unitary operator W' which is even more closely related to V
- We now have all the unitary operations that we require for Grover's algorithm
- We shall refer to the application of V followed by an application of W as a Grover iteration
- Each iteration only calls the *search* function once

Grover's Algorithm

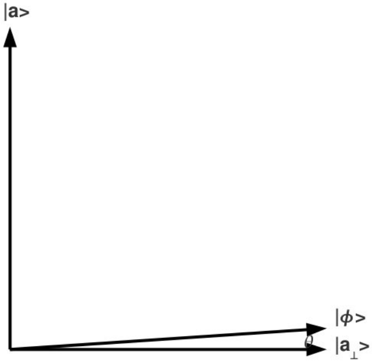
- We can now use a geometric interpretation to show that after only $\frac{\pi}{4}\sqrt{N}$ Grover iterations, we can measure (with high probability) to get the state $|a\rangle$
- In order to do this, we can notice that both V and W acting on the states $|a\rangle$ and $|\phi\rangle$ will return linear combinations of those two states (with Real coefficients)

$$V|a\rangle = -|a\rangle \qquad V|\phi\rangle = |\phi\rangle - \frac{2}{2^{\frac{n}{2}}}|a\rangle$$

- Remembering that $\langle\alpha|\beta\rangle = \langle\beta|\alpha\rangle^*$, we have $\langle\phi|a\rangle = \langle a|\phi\rangle = \frac{1}{2^{\frac{n}{2}}}$

$$W|\phi\rangle = |\phi\rangle \qquad W|a\rangle = \frac{2}{2^{\frac{n}{2}}}|\phi\rangle - |a\rangle$$

Grover's Algorithm

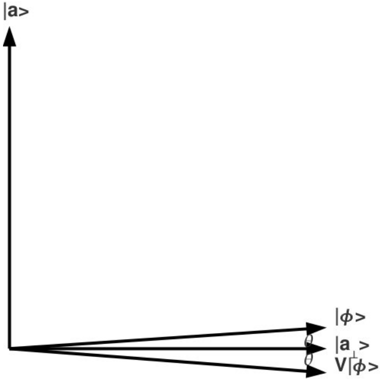


- If we start with the state $|\phi\rangle$, and only perform combinations of V and W, then we can visualise this on a plane spanned by the states $|a\rangle$ and $|\phi\rangle$
- The state $|a_\perp\rangle$ contains all the states orthogonal to $|a\rangle$

Grover's Algorithm

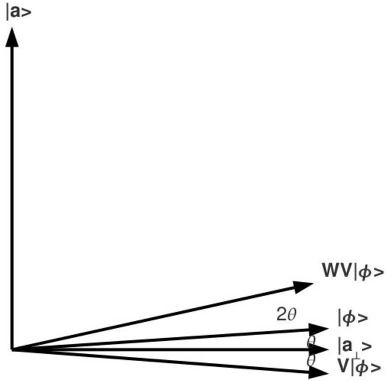
- For large N, $|\phi\rangle$ is close to $|a_\perp\rangle$
- We can calculate the angle θ using $sin\theta = 2^{\frac{-n}{2}} = \frac{1}{\sqrt{N}}$
- Which for large N can be approximated to $\theta \approx 2^{\frac{-n}{2}}$
- We can now look at the behaviour of the unitary operations V and W on this plane
- W leaves $|\phi\rangle$ invariant, and reverses the direction of any vector orthogonal to $|\phi\rangle$
- V reverses the direction of $|a\rangle$ and leaves any vector orthogonal to $|a\rangle$ unchanged

Grover's Algorithm



- V represents a reflection about the $|a_{\perp}\rangle$ vector

Grover's Algorithm



- W represents a reflection about the $|\phi\rangle$ vector
- Two reflections combine to form a rotation

Grover's Algorithm

- So, each Grover iteration rotates the state by an angle of 2θ
- Applying a Grover iteration to the state $|\phi\rangle$ gives us a state that sits 3θ from $|a_\perp\rangle$
- Applying a Grover iteration again gives us a state that sits 5θ from $|a_\perp\rangle$
- and so on...
- We know that $|a\rangle$ is orthogonal to $|a_\perp\rangle$, so we just need to work out how many Grover iterations are required to get us a close to $|a\rangle$ as possible
- Iterations required $= \frac{\pi}{2}.\frac{1}{2\theta} = \frac{\pi}{4\theta}$
- Since $\theta \approx 2^{\frac{-n}{2}}$, this simplfies to $\frac{\pi}{4}2^{\frac{n}{2}} = \frac{\pi}{4}\sqrt{N}$
- We can check if we've measured the correct result with one last call to the searching *oracle*

Next week...

- Next week, we shall look at an example of Grover's algorithm over a search space of size $N = 8$
- and start to look at the most famous quantum algorithm...
- Shor's algorithm
- It's quite complicated, so we shall be spending the next two weeks looking at it
- Remember, labs on Thursday!
- I hope to see you there
- Thank you