

Quantum information and computing

Lecture 9: Quantum search algorithms

Jani-Petri Martikainen

Jani-Petri.Martikainen@helsinki.fi

<http://www.helsinki.fi/~jamartik>

Department of Physical Sciences

University of Helsinki

Quantum search algorithms

- Suppose you are given a map containing many cities and you want to find the shortest route passing through all cities.
- A simple algorithm is to search through all possible routes and keep track of the shortest one.
- On a classical computer, if there are N routes, it takes $\mathcal{O}(N)$ operations to find the shortest route.
- With a quantum search algorithm (**Grover's algorithm**) this task can be accomplished with $\mathcal{O}(\sqrt{N})$ operations.

Quantum search algorithms: Oracle

- Suppose we wish to search through a space of N elements
- Rather than search the elements directly, we concentrate on **index** $(0 \dots N - 1)$ of those elements
- Choose $N = 2^n$ for convenience so the index can be stored in n bits
- Each search problem has exactly M solutions with $1 \leq M \leq N$
- For instance represent the problem by a function $f(x)$ ($x = 0 \dots N - 1$). $f(x) = 1$ if x is a solution to the search problem and 0 otherwise

Quantum search algorithms: Oracle

- We are supplied with a quantum **oracle**: a black box, with an ability to recognize solutions to the search problem.
- Recognition is signaled by using an oracle qubit. Oracle acts as

$$|x\rangle|q\rangle \rightarrow |x\rangle|q \oplus f(x)\rangle \quad (1)$$

and the oracle qubit $|q\rangle$ is a single qubit which is flipped if $f(x) = 1$ and unchanged otherwise.

- We can check if x is a solution by preparing $|x\rangle|0\rangle$, applying the oracle and checking to see if the oracle qubit has been flipped to $|1\rangle$
- In quantum search it is useful to prepare the oracle qubit in $(|0\rangle - |1\rangle)/\sqrt{2}$

Quantum search algorithms: Oracle

- If x is not a solution then the oracle won't change the initial state $|x\rangle(|0\rangle - |1\rangle)/\sqrt{2}$, otherwise we get a state $-|x\rangle(|0\rangle - |1\rangle)/\sqrt{2}$...

$$|x\rangle(|0\rangle - |1\rangle)/\sqrt{2} \rightarrow (-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle)/\sqrt{2} \quad (2)$$

- Notice that the state of the oracle qubit did not change! Therefore, it can be omitted from further discussion of the algorithm. This makes our description simpler and we express the action of the oracle as

$$|x\rangle \rightarrow (-1)^{f(x)}|x\rangle \quad (3)$$

- Oracle marks the solutions by shifting the phase of the solution

Quantum search algorithms: Oracle

- It turns out that we need only apply the search oracle $\mathcal{O}(\sqrt{N/M})$ times in order to find a solution on a quantum computer.
- Why discuss oracle without describing how it works? It seems as though the oracle already knows the answer.
- Answer: There is a difference in **knowing** the solution and being able to **recognize** the solution. Crucially it is possible to do the latter without necessarily being able to do the former!
- Example: factoring, some large $m = pq$ is a product of two primes. Obvious method on a classical computer is to search through all numbers from 2 to $m^{1/2}$ and find smaller of the two primes...requires roughly $m^{1/2}$ trial divisions.

Quantum search algorithms: Oracle

- Quantum search can speed things up: By definition with input $|x\rangle$ the oracle divides m by x and checks if the division is exact, flipping the oracle qubit if so.
- But we still need an efficient circuit to implement the oracle. (exercise in reversible computing)
- Begin by defining $f(x) = 1$ if x divides m and $f(x) = 0$ otherwise
- Then construct a classical reversible circuit which takes (x, q) (input and a single bit output register) into $(x, q \oplus f(x))$ (modify the usual irreversible classical circuit doing the trial division)
- Irreversible circuit does not require essentially more resources

Quantum search algorithms: Oracle

- Classical reversible circuit can be immediately translated into a quantum circuit that takes $|x\rangle|q\rangle \rightarrow |x\rangle|q \oplus f(x)\rangle$ as required of the oracle.
- **Key: even without knowing the prime factors m we can construct an oracle which recognizes a solution to the search problem when it sees one.**
- Using this oracle and a quantum search algorithm we can search the range from $2 \dots m^{1/2}$ using $\mathcal{O}(m^{1/4})$ oracle consultations instead of the $\mathcal{O}(m^{1/2})$ with a classical algorithm
- (Factoring example was just a conceptual example, not a practical one. There are much faster classical algorithms for factoring.)

Quantum search algorithm

- Schematic circuit for the quantum search algorithm...SEE DIAGRAM
- algorithm makes use of a single n qubit register.
- Internal workings of the oracle are not important for the description of the search algorithm
- The goal is to find a solution to the search problem using as few calls to the oracle as possible.
- Computer starts in $|0\rangle^{\otimes n}$ and then Hadamard transforms put the computer into state

$$|\psi\rangle = \frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle \quad (4)$$

Quantum search algorithm

- Quantum search algorithm uses a repeated application of the quantum subroutine known as the **Grover operator** or **Grover iteration** denoted by G
- Grover iteration in four steps:
 1. Apply the oracle O
 2. Apply the Hadamard transform $H^{\otimes n}$
 3. Perform the conditional phase shift, with every computational basis state except $|0\rangle$ receiving a phase shift of -1 :

$$|x\rangle \rightarrow (-1)^{\delta_{x,0}} |x\rangle \quad (5)$$

4. Apply the Hadamard transform $H^{\otimes n}$

Quantum search algorithm

- Each of these operations may be efficiently implemented on a quantum computer...
- steps 2 and 4 for the Hadamards require $n = \log N$ operations each...
- step 3 for the conditional phase shifts may be implemented using $\mathcal{O}(n)$ gates
- Cost of the oracle call depends on application, but the Grover iteration requires just one oracle call
- steps from 2...4 can be combined into

$$H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2|\psi\rangle\langle\psi| - I \quad (6)$$

where $|\psi\rangle$ is the equally weighted superposition state in Eq. (4). Grover iteration $G = (2|\psi\rangle\langle\psi| - I)O$

Geometric picture

- G can be viewed as a rotation in a 2D space.
- SHOW THIS ON A BLACK BOARD

Quantum search algorithm: Performance

- How many times must the Grover iteration be repeated in order to rotate $|\psi\rangle$ close to $|\beta\rangle = 1/\sqrt{M} \sum' |x\rangle$, where \sum' denotes sum over the solutions of the search problem.
- If $|\alpha\rangle = 1/\sqrt{N-M} \sum'' |x\rangle$ is where \sum'' is a sum over states that are NOT solutions then

$$|\psi\rangle = \sqrt{(N-M)/N} |\alpha\rangle + \sqrt{M/N} |\beta\rangle \quad (7)$$

- Rotating through $\arccos \sqrt{M/N}$ takes the system to $|\beta\rangle$
- Let $CI(x)$ denote the integer closest to the real number x (round halves down).

Quantum search algorithm: Performance

- Then repeating Grover iteration

$$R = CI \left(\frac{\arccos \sqrt{M/N}}{\theta} \right) \quad (8)$$

times rotates $|\psi\rangle$ within angle $\theta/2 \leq \pi/4$ of $|\beta\rangle$

- Observation of the state in the computational basis then yields a solution to the search problem with a probability bigger than $1/2$.
- For some specific N and M it is possible to achieve a much higher probability.
- For example, with $M \ll N$, $\theta \approx \sin \theta \approx 2\sqrt{M/N}$ and the angular error in the final state is at most $\theta/2 \approx \sqrt{M/N}$. This gives a probability of an error of at most M/N

Quantum search algorithm: Performance

- R depends on the number of solutions, but not on the identity of those solutions.
- So provided we know M we can apply the quantum search algorithm as described here.
- Even this requirement can be removed as we will explain later.
- Note that $R \leq \pi/2\theta$ so if $M \leq N/2$ we have $\theta/2 \geq \sin \theta/2 = \sqrt{M/N}$...upperbound

$$R \leq \frac{\pi}{4} \sqrt{\frac{N}{M}} \quad (9)$$

Quantum search algorithm

Input: a black box oracle, $n + 1$ qubits in $|0\rangle$

Output: x_0

1. Initial state: $|0\rangle^{\otimes n} |0\rangle$

2. $H^{\otimes n}$ to first n qubits and HX to the last qubit:

$$\rightarrow \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]$$

3. Apply Grover iteration $R \approx \pi\sqrt{2^n}/4$ times:

$$\rightarrow [(2|\psi\rangle\langle\psi| - I)O]^{\otimes R} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]$$

$$\approx |x_0\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]$$

(10)

4. Measure the first n qubits: $\rightarrow x_0$

Quantum search algorithm

- What if more than half the items are solutions i.e. $M > N/2$?
- Then we see that the angle θ gets smaller as M varies from $N/2$ to N . Therefore, the number of required iterations would increase with M . This is silly property for a search algorithm... search should be easier if there are more solutions.
- If we know that $M \geq N/2$ then we could just guess a solution and use the oracle to check it.
- If we do not know the number of solutions, we can use an alternative which is useful also in counting the number of solutions.
- Idea: double the number of elements in the search space by adding elements which are not solutions.

Quantum search algorithm

- This is effected by adding a single qubit $|q\rangle$ to the search index, doubling the number of items to be searched.
- The new augmented oracle O' is also constructed which marks the solutions and the extra bit is set to zero. We can make this construction using the earlier oracle O .
- The new problem only has M solutions out of $2N$ so running the algorithm with the new oracle O' we see that at most

$$R = \pi/4\sqrt{2N/M} \quad (11)$$

calls to O' are required.

- It follows that $\mathcal{O}(\sqrt{N/M})$ calls to O are required.

Quantum search as a quantum simulation

- Now we “derive” the quantum search algorithm from a more physical stand point.
- Assume for the sake of simplicity that the search problem has exactly one solution, which we label x
- **First** we guess a Hamiltonian which solves the search problem: We write down a Hamiltonian H which depends on the solution x and the initial state $|\psi\rangle$ (specify this also) such that the system will evolve from $|\psi\rangle$ to $|x\rangle$ after some prescribed time.
- **Second:** simulate the action of the Hamiltonian using a quantum circuit.
- Amazingly, this leads very quickly to the quantum search algorithm.

Quantum search as a quantum simulation

- Algorithm starts with the quantum computer in $|\psi\rangle$ (we will tie it down later)
- For simplicity we try to construct the Hamiltonian using only states $|\psi\rangle$ and $|x\rangle$. H is a sum of terms like $|\psi\rangle\langle\psi|$, $|x\rangle\langle x|$, $|\psi\rangle\langle x|$, and $|x\rangle\langle\psi|$
- Guess $H = |\psi\rangle\langle\psi| + |x\rangle\langle x|$ or $H = |x\rangle\langle\psi| + |\psi\rangle\langle x|$...both work!
- Let us choose the first one. After some time t the state evolves to

$$\exp(-iHt)|\psi\rangle \quad (12)$$

- Intuitively promising: for small t $|\psi\rangle$ goes to $(1 - it)|\psi\rangle - it\langle x|\psi\rangle|x\rangle$

Quantum search as a quantum simulation

- I.e. the state is rotated slightly towards the solution.
- We can restrict our focus into a space spanned by $|x\rangle$ and $|\psi\rangle$ and using Gram-Schmidt procedure find $|y\rangle$ so that $|x\rangle$ and $|y\rangle$ form an orthonormal basis and $|\psi\rangle = \alpha|x\rangle + \beta|y\rangle$ ($\alpha^2 + \beta^2 = 1$)
- In this orthonormal basis we have

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \alpha^2 & \alpha\beta \\ \alpha\beta & \beta^2 \end{bmatrix} = I + \alpha(\beta X + \alpha Z) \quad (13)$$

- Therefore the state evolves according to

$$\exp(-it) [\cos \alpha t |\psi\rangle - i \sin \alpha t (\beta X + \alpha Z) |\psi\rangle] \quad (14)$$

Quantum search as a quantum simulation

- Global phase factor can be ignored and simple algebra shows that $(\beta X + \alpha Z)|\psi\rangle = |x\rangle$.
- Therefore, observation of the system at a time $t = \pi/2\alpha$ yields the result $|x\rangle$ with the probability one. We have found a solution to the search problem!
- Unfortunately, the time of the observation depends on α and thus on $|x\rangle$ which we were trying to determine.
- Try to arrange things so that α is the same for all $|x\rangle$. This is achieved by using

$$|\psi\rangle = \sum_x |x\rangle / \sqrt{N} \quad (15)$$

in which case $\alpha = 1/\sqrt{N}$ and $t = \pi\sqrt{N}/2$. This state we can prepare using Hadamard transformations.

Quantum search as a quantum simulation

- Can we find a quantum circuit to simulate this Hamiltonian?
- Natural way is to alternately simulate Hamiltonians $H_1 = |x\rangle\langle x|$ and $H_2 = |\psi\rangle\langle\psi|$ for short time increments Δt .
- These can be done with circuits...SEE DIAGRAMS
- The number of oracle calls required by the quantum simulations is determined by how small time-step is required to obtain reasonably accurate results.
- Suppose we use a step Δt which is accurate to $\mathcal{O}(\Delta t^2)$
- The total number of steps is $t/\Delta t = \mathcal{O}(\sqrt{N}/\Delta t)$ and the cumulative error is $\mathcal{O}(\Delta t^2 \times \sqrt{N}/\Delta t) = \mathcal{O}(\Delta t\sqrt{N})$

Quantum search as a quantum simulation

- To obtain reasonably high success probability we need the error to be $\mathcal{O}(1)$ which means we must choose $\Delta t = \mathcal{O}(1/\sqrt{N})$ which results in $\mathcal{O}(N)$ oracle calls- no better than the classical solution.!
- What if we use a more accurate method of simulation lets say $\mathcal{O}(\Delta t^3)$?
- The cumulative error in this case is $\mathcal{O}(\Delta t^2 \sqrt{N})$ and thus we need to choose $\Delta t = \mathcal{O}(N^{-1/4})$ to achieve reasonable success rate.
- This gives the total number of oracle calls $\mathcal{O}(N^{3/4})$, which is an improvement over the classical situation although not as good as the quantum search algorithm

Quantum search as a quantum simulation

- Moving to higher accuracy simulation brings us ever closer to the quantum search algorithm scaling. (exercise)
- Lets calculate the evolution using the low order $U(\Delta t) = \exp(-i|\psi\rangle\langle\psi|\Delta t) \exp(-i|x\rangle\langle x|\Delta t)$ approximation for the exact evolution operator.
- In the orthonormal $|x\rangle |y\rangle$ basis $|x\rangle\langle x| = (I + Z)/2 = (I + \hat{z} \cdot \bar{\sigma})$ where $\hat{z} = (0, 0, 1)$. Also $|\psi\rangle\langle\psi| = (I + \bar{\psi} \cdot \bar{\sigma})$ where $\bar{\psi} = (2\alpha\beta, 0, (\alpha^2 - \beta^2))$
- It then follows that

$$U(\Delta t) = (\cos^2(\Delta t/2) - \sin^2(\Delta t/2)\bar{\psi} \cdot \hat{z}) I - 2i \sin(\Delta t/2) \left(\cos(\Delta t/2) \frac{\bar{\psi} + \hat{z}}{2} + \sin(\Delta t/2) \frac{\bar{\psi} \times \hat{z}}{2} \right) \cdot \bar{\sigma}$$

Quantum search as a quantum simulation

- This is a rotation on the Bloch sphere about an axis of rotation

$$\bar{r} = \cos(\Delta t/2) \frac{\bar{\psi} + \hat{z}}{2} + \sin(\Delta t/2) \frac{\bar{\psi} \times \hat{z}}{2} \quad (16)$$

and through an angle defined by

$$\cos(\theta/2) = \cos^2(\Delta t/2) - \sin^2(\Delta t/2) \bar{\psi} \cdot \hat{z} \quad (17)$$

- using $\bar{\psi} \cdot \hat{z} = \alpha^2 - \beta^2 = (2/N - 1)$ we get

$$\cos(\theta/2) = 1 - \frac{2}{N} \sin^2(\Delta t/2) \quad (18)$$

Quantum search as a quantum simulation

- Since $\bar{\psi} \cdot \bar{r} = \hat{z} \cdot \bar{r}$, both $|\psi\rangle\langle\psi|$ and $|x\rangle\langle x|$ lie on the same circle of revolution about the \bar{r} axis of the Bloch sphere.
- So... $U(\Delta t)$ rotates $|\psi\rangle\langle\psi|$ about an \bar{r} axis.
- We terminate the rotations once we have done enough of them to rotate near to the solution $|x\rangle\langle x|$

Quantum search as a quantum simulation

- At first we assumed that Δt is small but looking at the exact solution in Eq.(18) suggests that we should instead choose $\Delta t = \pi$ in order to maximize the rotation angle.
- If we do this then $\cos \theta/2 = 1 - 2/N$ which for large N indicates $\theta \approx 4/\sqrt{N}$.
- The the number of oracle calls is $\mathcal{O}(\sqrt{N})$ just as for the quantum search algorithm!
- With this choice of the time-step $\exp(-i\pi|\psi\rangle\langle\psi|) = I - 2|\psi\rangle\langle\psi|$ and $\exp(-i\pi|x\rangle\langle x|) = I - 2|x\rangle\langle x|$. Up to the global phase shift these are **identical to steps in the Grover iteration!**
- We have rederived the quantum search algorithm!

Quantum search as a quantum simulation

Some hints for quantum algorithm design

1. Specify the problem to be solved, including desired input and output from the quantum algorithm.
2. Guess a Hamiltonian to solve the problem and verify that it actually works.
3. Find a procedure to simulate the Hamiltonian.
4. Analyze the resource costs of the simulation.

Quantum counting

- How do we determine the number of solutions M to the search problem?
- On a classical computer this takes $\mathcal{O}(N)$ consultations with an oracle, but on a quantum computer this task can be accomplished faster.
- I will not explain this in detail...see the text book for more information.
- The idea is to combine Grover iteration with phase estimation.
- This allows us to find θ in $|\psi\rangle = \cos \theta/2|\alpha\rangle + \sin \theta/2|\beta\rangle$ (see earlier for the definition of $|\alpha\rangle$ and $|\beta\rangle$)

Quantum counting

- Once we have sufficiently accurate estimate for the phase we can use the relation

$$\sin^2 \theta/2 = M/2N \quad (19)$$

to find M .

- Note: the number of elements above was doubled to $2N$ in order to ensure that the number of solutions is guaranteed to be less than half the number of elements.
- In this way the quantum search algorithm is guaranteed to work and can be used after the quantum counting step to find a solution.
- Quantum counting can be necessary, since in order to know how many (R) Grover iterations we must use in the search, we need to know M

Quantum search algorithm

- It can be shown that the quantum search algorithm we gave here is optimal in a sense that $\mathcal{O}(\sqrt{N})$ calls to the oracle is the best we can do.
- See the text book for proof.
- A bit disappointing since we might have dreamed of an algorithm finding a solution in $\mathcal{O}(\log N)$ calls.
- If such algorithm existed we would be able to solve NP-complete problems efficiently on a quantum computer.
- Naive search-based method for attacking NP-complete problems is guaranteed to fail!