Lecture 8: Unstructured search and query complexity

Rajat Mittal
IIT Kanpur

We move on to the other famous quantum algorithm, Grover’s algorithm. It does not provide exponential speedup but it deals with the very general search problem. This makes the algorithm useful in variety of applications. Subsequently, many variations and other algorithms have been discovered on the basis of Grover’s algorithm.

The query complexity framework is really useful to look at the search problem and it complexity. We will briefly discuss the framework in the end and see some results.

1 Search Problem

The search problem is a very general problem of searching through an unstructured database. For example, consider that you want to find the roll number of a student through her name but the list is sorted through roll numbers (or unsorted, sorted on the basis of roll number does not help).

Specifically, you are given \( n \) elements in a list and some of them are marked. You have an oracle, given an index of the list, it tells you if the corresponding element is marked or not. The task is to find a marked element.

For simplification we will assume that there is only one marked element. Also we will assume that \( n = 2^k \), so that the index is a \( k \) bit string. Then \( f : \{0,1\}^k \rightarrow \{0,1\} \) is function which tell us whether the index is marked or not. So, the action of the oracle is,

\[
|x, b\rangle \rightarrow |x, b \oplus f(x)\rangle.
\]

Where \( x \) is a \( k \)-bit string and \( b \) is a bit. This, like before, can be changed into an oracle,

\[
O_f|x\rangle = (-1)^f(x)|x\rangle.
\]

Exercise 1. Do you remember how we did it? If not, go back and check.

Given such an oracle \( O_f \), we need to find an \( x \), such that, \( f(x) = 1 \). At this point, let us just say that we are interested in the number of queries to oracle \( O_f \).

Clearly, a classical algorithm needs to look through all the indices and hence has to query oracle \( \Theta(n) \) times.

Exercise 2. What about the randomized algorithm?

Grover’s algorithm finds the marked element in \( O(\sqrt{n}) \) queries. This might not seem like a big improvement (it is polynomial as compared to exponential), but is very useful because of the usefulness of search. Many different problems in diverse areas have a brute force search algorithm. If we have the subroutine (oracle) to check the solution, we can improve the running time quadratically for all these problems on a quantum computer.

One example would be NP-complete problems. Their solutions are easy to verify but difficult to find. The search for their solutions can be improved by a quadratic factor.

Exercise 3. Can it help in the brute force algorithm for factoring?

Before we see the solution, let us take a look at the important parts of the search problem.

Since we assumed that there is only one marked element, say \( x^0 \), we are looking for the state,

\[
|M\rangle = |x^0\rangle.
\]
The remaining states (bad states, where we don’t want to end) has an equal superposition,

$$|U\rangle = \frac{1}{\sqrt{n-1}} \sum_{x \neq x^0} |x\rangle.$$ 

The states $|M\rangle, |U\rangle$ are orthogonal to each other. The plane spanned by them contains the equal superposition,

$$|\psi\rangle = \frac{1}{\sqrt{n}} \sum_{x} |x\rangle.$$

**Exercise 4.** Find the coefficients $\alpha, \beta$, so that,

$$|\psi\rangle = \alpha |M\rangle + \beta |U\rangle.$$

We will just worry about the plane spanned by $|U\rangle$ and $|M\rangle$. Using Hadamard transformation, we can create the state $|\psi\rangle$. The objective is to go closer to state $|M\rangle$ and away from $|U\rangle$.

What does the oracle do? The oracle puts a phase of $-1$ for the marked state and leaves others unchanged. Hence the action of the oracle is $I - 2|x^0\rangle\langle x^0|$. This is equivalent to $2|U\rangle\langle U| - I$.

The matrices of the form $2|\phi\rangle\langle \phi| - I$ are called the reflection matrices. That is because they reflect around the vector $|\phi\rangle$.

**Exercise 5.** Convince yourself that a reflection matrix performs a reflection in a plane.

So we have state $|\psi\rangle$ in the plane spanned by $|U\rangle$ and $|M\rangle$. We can reflect about the state $|U\rangle$ (the oracle), and we need to get closer to $|M\rangle$.

This will be achieved by using another reflection. Why, for that we need to understand the properties of product of two reflections.

---

**Fig. 1.** Search problem
1.1 Product of two reflections

Let us say that we are given two reflections, one about vector $|a\rangle$ and $|b\rangle$. We will consider the plane spanned by $|a\rangle$ and $|b\rangle$. Without loss of generality, assume that $|a\rangle = |0\rangle$ and $|b\rangle = \cos \theta |0\rangle + \sin \theta |1\rangle$.

The reflection around $|a\rangle$ is $2|0\rangle\langle 0| - I$, and hence equal to the matrix,

$$
\begin{pmatrix}
1 & 0 \\
0 & -1
\end{pmatrix}
$$

**Exercise 6.** Show that the reflection around $|b\rangle$ is,

$$
2|b\rangle\langle b| - I = \begin{pmatrix}
2 \cos^2 \theta - 1 & 2 \cos \theta \sin \theta \\
2 \cos \theta \sin \theta & 2 \sin^2 \theta - 1
\end{pmatrix}
$$

Then the product of these two reflections is

$$R_{2\theta} = \begin{pmatrix}
\cos 2\theta & -\sin 2\theta \\
\sin 2\theta & \cos 2\theta
\end{pmatrix}
$$

**Exercise 7.** Find the eigenvalues and eigenvectors of this matrix.

This is the rotation matrix in the plane which rotates by angle $2\theta$. So we conclude, the product of two reflections is a rotation by angle $2\theta$, where $\theta$ is the angle between the reflection axes.

**Exercise 8.** Can you guess the Grover’s algorithm now?

2 Grover Search

As hinted above, the idea is to rotate the equal superposition vector multiple times and bring it as close to the marked state as possible. The rotation will be obtained by the product of two reflections. One reflection is around the unmarked states (performed by the oracle), and the other will be around the equal superposition $|\psi\rangle$.

One rotation is known as the Grover iteration and by previous section it rotates any vector in the plane of $|U\rangle$ and $|M\rangle$ by an angle $2\theta$. We need to figure out two things.

– How to perform reflection around $|\psi\rangle$?
– How many times should we rotate?

The first question is easier to answer.

**Exercise 9.** How can we perform reflection around $|0\rangle$ state?

Since $|0\rangle$ is a constant state, in the sense that it does not depend on the input, we can easily recognize it and hence reflect around it. Put a phase of $-1$ if the state is not $|0\rangle$, otherwise keep it unchanged (using controlled operations).

Now notice that $|\psi\rangle = H^{\otimes k}|0\rangle$. So the reflection around $|\psi\rangle$ will be performed by,

– Apply Hadamard transform to switching to $|0\rangle$ basis.
– Perform the reflection around $|0\rangle$.
– Apply Hadamard to come back to $|\psi\rangle$ basis.

So the reflection around $|\psi\rangle$ will be written as,

$$H^{\otimes k}(2|0\rangle\langle 0| - I)H^{\otimes k} = 2|\psi\rangle\langle \psi| - I.$$

Now we turn to the second problem, how many times should we rotate the initial vector $|\psi\rangle$.

The grover iteration is $G = H^{\otimes k}(2|0\rangle\langle 0| - I)H^{\otimes k}O_f$. This will rotate any vector by an angle $2\theta$ in the plane of $|U\rangle$ and $|M\rangle$. 


Exercise 10. What is $\theta$ here?

From the Fig. 2, it is clear that $\theta = \cos^{-1} \sqrt{\frac{n-1}{n}}$.

After $l$ rotations the state $|\psi\rangle$ is at an angle $\theta + 2l\theta = (2l + 1)\theta$ from the unmarked state $|U\rangle$. We want this angle to be as close to $\pi/2$ as possible.

So we need to apply grover iteration,

$$l = \left( \frac{\pi}{2\cos^{-1} \sqrt{\frac{n-1}{n}} - 1} \right)/2,$$

times. Since we can only apply Grover iteration integer number of times, we will apply it closest integer to $l$ times.

To analyze the complexity of the algorithm, notice that $l = \Theta\left(\frac{1}{\theta}\right)$. Since for small angles $\sin \theta \approx \theta$. The number of oracle calls required are $O(\sqrt{n})$.

Exercise 11. Since we can do the reflection around $|\psi\rangle$ in polylog($n$) operations. Show that the time complexity of Grover’s algorithm is also $O(\sqrt{n})$.

To conclude, the Grover’s algorithm is stated below.

- Apply Hadamard on $|0\rangle$ state to obtain $|\psi\rangle$.
- Apply Grover iteration $G$ to the state $|\psi\rangle$ for $l'$ iterations. Where $l'$ is the closest integer to $l = \left( \frac{\pi}{2\cos^{-1} \sqrt{\frac{n-1}{n}} - 1} \right)/2$.
- Measure in the standard basis.

Exercise 12. Since we can only apply Grover iteration integer number of times, bound the the maximum possible error.
The same algorithm can be extended to the case when number of marked items are less than \( \frac{n}{2} \). The details are given as an assignment question. If the number of marked items are more than \( \frac{n}{2} \), there are easy ways to find a marked item (assignment).

2.1 Number of marked items are not known

Notice that if we rotate more than the exact amount, then we get away from the marked state. So it seems that we need to know the number of marked items.

How can we estimate the number of marked items?

The answer is hidden in the Grover’s algorithm and its properties. Remember that we calculated the eigenvalues and eigenvectors of the rotation matrix,

\[
R_{2\theta} = \begin{pmatrix}
\cos 2\theta - \sin 2\theta \\
\sin 2\theta & \cos 2\theta
\end{pmatrix}
\]

The Grover iteration \( G \) is a matrix of this kind with \( \cos \theta = \sqrt{\frac{n-m}{n}} \), where \( m \) is the number of marked elements.

Exercise 13. Prove the above statement.

In the exercise above, you were asked to find the eigenvalue and eigenvectors of this matrix \( G \).

The matrix \( G \) has an eigenvector \( \frac{1}{\sqrt{2}} |U\rangle + \frac{1}{\sqrt{2}} |M\rangle \) with eigenvalue \( e^{2i\theta} \). The other eigenvector is \( \frac{1}{\sqrt{2}} |U\rangle + \frac{i}{\sqrt{2}} |M\rangle \) with eigenvalue \( e^{-2i\theta} \).

So finding the number of marked solutions is essentially a phase estimation on the Grover operator \( G \).

Exercise 14. What is the relation between \( \theta \) and number of marked elements?

The only thing is, we need to start with an eigenvector of \( G \).

Exercise 15. Show that \( |\psi\rangle \) is a linear combination of the two eigenvectors mentioned above.

Hence, we can start with state \( |\psi\rangle \) and apply phase estimation on \( G \). We will either obtain \( 2\theta \) or \( -2\theta \). In either case, it is easy to determine the number of marked elements. It is enough to find \( m \) with \( n/2 + f(\epsilon) \) bits of precision, where \( \epsilon \) is the error probability of phase estimation. Hence only \( O(\sqrt{n}) \) calls to the oracle are enough to estimate \( m \). For a proof of this, please refer to [1].

Exercise 16. Write the circuit for finding the number of marked elements.

This algorithm is also called quantum counting.

2.2 Amplitude amplification

Given a set \( X \), say there is a good subset \( X^1 \) and there is a bad subset \( X^0 \). We are interested in finding out the elements in the good subset. Again, given a good element, we can recognize/verify it efficiently.

Exercise 17. Why is this problem useful?

Suppose we are given an algorithm which finds a good element with probability \( p \). How can we increase the probability of success. The classical approach would be to apply the algorithm and check if the obtained element is good or not. Repeat this procedure \( l \) times. Then the probability of success is,

\[
(1 - p)^l.
\]

Exercise 18. What \( l \) will make the above procedure succeed with constant probability?
The above exercise shows that we need to implement $A$ around $1/p$ times. What will be the quantum analog of this? The Hilbert space $H$ is the space spanned by $|x\rangle$ for all $x \in X$. The good subspace is $S_1$ spanned by $|x\rangle$ for all $x \in X^1$ and similarly we have $S_0$. We are given the ability to recognize good elements, in other words, there is an oracle $O_G$ which puts the phase $-1$ if and only if the element is good.

A quantum algorithm $A$, will move $|0\rangle$ to state $A|0\rangle$ which will have $1/\sqrt{p}$ overlap with the good subspace. This ensures that the algorithm succeeds with the probability $1/p$.

The quantum analog of the problem considered above is, how many iteration of $A$ are needed to boost the probability of success to a constant. It turns out that we can do it using $O(1/\sqrt{p})$ iterations of $A$ and its inverse $A^{-1}$. This is quadratically faster than the classical approach.

Note 1. The inverse of algorithm $A$ exists if it does not do any measurement. This is done by amplifying the amplitude of the state on good subspace and hence is called amplitude amplification. The strategy is very similar to Grover search and can be seen as a generalization. Suppose the state $A|0\rangle$ is,

$$|\psi\rangle := A|0\rangle = \cos \theta |B\rangle + \sin \theta |G\rangle.$$  

Where $|G\rangle$ is the closest state to $|\psi\rangle$ in the good subspace and similarly $|B\rangle$.

Exercise 19. Show that we can always write $|\psi\rangle$ as such a state. what do we know about $\theta$?

The strategy for amplitude amplification is similar. Consider the plane spanned by $|B\rangle$ and $|G\rangle$. We will apply $G$ which will be the product of two reflections. First one will be around $|\psi\rangle$ and the other one will be around $|B\rangle$.

The reflection around $|B\rangle$ is given by the oracle $O_G$. The reflection around $|\psi\rangle$ is just $A(2|0\rangle\langle 0| - I)A^{-1}$.

Exercise 20. Show that $O(1/\sqrt{p})$ iteration of $G$ will suffice to create a state with at least $2/3$ overlap with good subspace.

Note 2. The amplitude amplification assumes that $A$ does not perform any measurements. Otherwise we will not have $A^{-1}$.

Why is amplitude amplification useful? Grover search makes the brute force search algorithm faster. Amplitude amplification can make other classical algorithms quadratically better if they have a certain form.

Exercise 21. On what kind of classical algorithms can we apply amplitude amplification?

3 Optimality of Grover search

We have shown that the search problem can be solved with $O(\sqrt{n})$ oracle queries. Can we do better? In this section, we will show that Grover search is optimal. That means, we will prove that any algorithm with less queries will not be able to give the correct answer with probability $\geq 1/2$. Again we will assume that there is only one marked element.

The first approach could be to say that if we start with a state close to $|U\rangle$, a query can only move the vector a small distance. Ultimately we should end up in state $|M\rangle$, which is very far away from $|U\rangle$. So lot of queries are required.

The problem with this argument is, we don’t need to start with something close to $|U\rangle$. But, a priori, we don’t know which element is marked. Since the algorithm should work for all oracles $O_x$, we can pick $O_x$ where $x$ is far from the starting state of the algorithm.

Let us see how a generic algorithm progresses in this framework of query.
Any generic algorithm will start with state $|\psi\rangle$ and apply unitaries and oracle one after another. If there are $l$ oracle queries, the final state can be written as,

$$|\psi_f\rangle = U_lO_xU_{l-1}O_x \cdots U_1O_x|\psi\rangle.$$ 

If there was no oracle then the state would have been,

$$|\psi_l\rangle = U_lU_{l-1} \cdots U_1|\psi\rangle.$$ 

The idea of the lower bound is (assuming we do small number of queries),

– The state $|\psi_l\rangle$ cannot be close to all $|x\rangle$.
– If the algorithm succeeds with high probability, then for all x, the state $|\psi_f^x\rangle$ should be close to $|x\rangle$. From the first point, the state $|\psi_l\rangle$ and $|\psi_f^x\rangle$, for almost all x’s, should be far.
– The state $|\psi_l\rangle$ and $|\psi_f^x\rangle$, for almost all x’s, should be close as we have done only small number of queries.

Hence we arrive at the contradiction. Notice that it is important that we consider all x’s, otherwise the argument is not valid.

To prove it formally, consider the potential function,

$$\Phi_l = \sum_x \|\psi_f^x - \psi_l\|^2.$$ 

The lower bound will follow from two parts,

– $\Phi_l \geq O(n)$, since algorithm succeeds with probability $\geq 1/2$.
– $\Phi_l \leq 4l^2$, since one query can’t change $\Phi_l$ by a large amount.

**Exercise 22.** Show that the lower bound follows from the two parts.

**3.1 $\Phi_l \geq O(n)$**

We know that,

$$\Phi_l = \sum_x \|\psi_f^x - \psi_l\|^2 = \sum_x \|\psi_f^x - x\| + (x - \psi_l)\|^2.$$ 

Expanding,

$$\Phi_l \geq \sum_x \|\psi_f^x - x\|^2 + \sum_x \|x - \psi_l\|^2 - 2\sum_x \|\psi_f^x - x\||x - \psi_l||.$$ 

Consider the first two terms,

$$s_l = \sum_x \|\psi_f^x\| - x\|, \quad b_l = \sum_x \|x - \psi_l\|.$$ 

**Exercise 23.** Using the Cauchy-Schwarz inequality, prove,

$$\Phi_l \geq (\sqrt{b_l} - \sqrt{s_l})^2.$$ 

Read about Cauchy-Schwarz inequality.

If the algorithm works correctly then $s_l$ should be small. For any constant unit vector $\psi$, $b_l$ should be big. This will prove the lower bound.

**Lemma 1.** If the algorithm outputs the correct answer with at least half the probability, then

$$s_l \leq (2 - \sqrt{2})n.$$
Proof. Notice that $|\langle x|\psi_l^r \rangle|^2 \geq 1/2$, because the algorithm works. Then the rest of the proof is given as an exercise. This exercise shows that for two unit vectors $a$ and $b$, if they are close ($\|a - b\|$ will be small) then $|\langle a|b \rangle|$ will big. If $a$ and $b$ are far then $|\langle a|b \rangle|$ will be small.

Lemma 2. For any unit vector $|\psi\rangle$,

$$\sum_x \|x - \psi\|^2 \geq 2n - 2\sqrt{n}.$$ 

Proof. First we will write $|\psi\rangle$ in the basis of $x$,

$$|\psi\rangle = \sum_x a_x |x\rangle.$$ 

Here $\sum_x |a_x|^2 = 1$. Then looking at the concerned quantity,

$$\sum_x \|x - \psi\|^2 = \sum_x \left( \sum_{y \neq x} |a_y|^2 + |1 - a_x|^2 \right).$$

Simplifying,

$$\sum_x \|x - \psi\|^2 = (n - 1) + \sum_x |1 - a_x|^2 = 2n - 2 \sum_x |a_x|. $$

Applying Cauchy-Schwarz on the last term, we get the required result.

Exercise 24. Prove that, $\sum_x |a_x| \leq \sqrt{n}$.

From the previous two lemmas and the Ex. 23

$$\Phi_l \geq (\sqrt{b_l} - \sqrt{s_l})^2 \geq O(n).$$

Exercise 25. Make sure that you can derive the above bound from lemma and the Ex. 23.

3.2 $\Phi_l \leq 4l^2$

Now we need to prove that $\Phi_l$ does not increase much after the application of a query. Notice that $\Phi_0 = 0$.

We will express $\Phi_{l+1}$ in terms of $\Phi_l$,

$$\Phi_{l+1} = \sum_x \|O_x \psi_l^r - \psi_l\|^2$$

$$= \sum_x \|O_x \psi_l^r - O_x \psi_l + O_x \psi_l - \psi_l\|^2$$

$$\leq \left( \sum_x \|O_x (\psi_l^r - \psi_l)\|^2 + \sum_x 2\|O_x (\psi_l^r - \psi_l)\| \|(O_x - I) \psi_l\| + \sum_x \|\psi_l\| \|(O_x - I) \psi_l\|^2 \right)$$

$$\leq \Phi_l + 4\sqrt{\Phi_l} + \sum_x \|O_x \psi_l^r - \psi_l\|^2 \quad \text{(using Cauchy-Schwarz)}$$

$$\leq \Phi_l + 4\sqrt{\Phi_l} + 4 \sum_x |\langle x|\psi_l \rangle|^2$$

$$\leq \Phi_l + 4\sqrt{\Phi_l} + 4$$

Exercise 26. Make sure that you can prove every part of the above line of equations.

So we get that,

$$\Phi_{l+1} \leq \Phi_l + 4\sqrt{\Phi_l} + 4.$$ 

Exercise 27. Using induction, prove that $\Phi_l \leq 4l^2$. 

8
4 Quantum query complexity

The decision problem of search, whether there is a marked element or not, can be thought of as computing boolean OR function.

Exercise 28. Convince yourself that this is the case. Notice that name of elements is not important, only if they are marked or not.

Then the search problem can be posed as, given an input \( x \in \{0,1\}^n \), find if there is a 1 or not. We are given an oracle which gives \( x_i \) on input \( i \).

The number of queries needed to compute OR of \( x \), is called the query complexity of OR. From Grover search and the optimality of Grover search, the query complexity of OR is \( \Theta(\sqrt{n}) \).

This question can be posed for different functions too, like parity or majority of bits. The query complexity is important because it is a good substitute of time complexity in many cases.

We will look at one of the techniques to lower bound the query complexity of a general \( f \), inspired from the proof in previous section.

4.1 Adversary method

The proof for optimality of Grover search can be thought of as an argument between the algorithm and an adversary. The algorithm claims that it can solve OR with less than \( \sqrt{n} \) queries.

The adversary picks pairs of input which have different output and says that the algorithm should produce lot of difference (the potential function) between these pairs of input. But one query can’t produce much difference. So summing up, \( o(\sqrt{n}) \) queries won’t be able to produce required difference.

For the case of OR function the input pairs were 0, \( x \), where 0 is all 0 string and \( x \) is a string with exactly one 1.

We will see a generic form of this Adversary method. It is very important to come up with these pairs of inputs where function value differs.

Suppose we want to give a lower bound on the query complexity of a function \( f : \{0,1\}^n \rightarrow \{0,1\} \). To encode the input pairs, we will keep a matrix \( W \) with rows and columns indexed by inputs. \( W(x,y) \) gives a non-negative weight to the input pair we are considering, with \( W(x,y) = 0 \) if \( f(x) = f(y) \).

Suppose the state of the algorithm after \( l \) queries to input \( x \) is \( |\psi_x^l\rangle \). We will define the potential function in a very similar way as before,

\[
\Phi_l = \sum_{x,y} W_{x,y} \beta_x \beta_y \langle \psi_x^l | \psi_y^l \rangle,
\]

where \( \beta \) is the eigenvector corresponding to the maximum eigenvalue (absolute value).

**Total change in \( \Phi_l \)** Again we know that \( \Phi_0 = \sum_{x,y} W_{x,y} \beta_x \beta_y \).

Exercise 29. Prove that \( \Phi_0 = \|W\| \).

Say the algorithm takes \( t \) queries. If the algorithm succeeds with probability \( \epsilon \), then \( |\psi_t^x\rangle \) will be close to \( |f(x)\rangle \), maximum angle \( \theta \) such that \( \sin^2 \theta = \epsilon \). Similarly \|\psi_t^y\rangle \) will be close to \( |f(y)\rangle \). If \( f(x) \neq f(y) \) then \|\psi_t^y\rangle \) will be at least at an angle of \( \pi/2 - 2\theta \) with \|\psi_t^x\rangle \). Hence,

\[
\langle \psi_t^x | \psi_t^y \rangle \leq 2\sqrt{\epsilon(1-\epsilon)}.
\]

Exercise 30. Formally prove the argument in the paragraph above.

So the total change in \( \Phi_l \) is \( \Omega(\|W\|) \). We will upper bound the change because of every query and hence prove a lower bound on the number of queries.
Change through one query

After \( l + 1 \) queries the state on input \( x \) is \( U_{l+1}O_x|\psi_f^l\rangle \). Where \( U_{l+1} \) is independent of input \( x \). So,

\[
\Phi_{l+1} - \Phi_l \leq \sum_{x,y} W_{x,y}\beta_x \beta_y |\langle \psi_f^l | \psi_f^l \rangle - \langle O_x \psi_f^l | O_y \psi_f^l \rangle|.
\]

The term in the summation can be simplified to,

\[
|\langle \psi_f^l | O_y O_x - I | \psi_f^l \rangle| \leq \sum_{i : x_i \neq y_i} 2|\alpha_{x,i}|\alpha_{y,i}|.
\]

Here \( \alpha_{x,i} \) is the amplitude of \( |i\rangle \) in \( |\psi_f^l\rangle \), so \( \sum_i |\alpha_{x,i}|^2 = 1 \).

From Eq. 2

\[
\Phi_{l+1} - \Phi_l \leq \sum_{x,y} 2W_{x,y}\beta_x \beta_y \sum_{i : x_i \neq y_i} |\alpha_{x,i}\alpha_{y,i}|.
\]

For convenience, define a matrix \( \Delta_i \) of the same dimensions as \( W \), s.t., \( \Delta_i(x,y) = 1 \) if \( x_i \neq y_i \) and 0 otherwise. So,

\[
\Phi_{l+1} - \Phi_l \leq 2 \sum_{x,y} \sum_i (W \circ \Delta_i)_{x,y}\beta_x \beta_y |\alpha_{x,i}\alpha_{y,i}|.
\]

Here, \( W \circ \Delta_i \) denotes the matrix obtained by taking entry-wise multiplication of \( W \) and \( \Delta_i \).

Again, say \( v_i \) is a vector indexed by \( x \), s.t., \( v_i(x) = \beta_x |\alpha_{x,i}| \).

The change in potential can be bounded as,

\[
\Phi_{l+1} - \Phi_l \leq 2 \sum_{x,y} \sum_i (W \circ \Delta_i)_{x,y}v_i(x)v_i(y)
\]

\[
= 2 \sum_i \sum_{x,y} (W \circ \Delta_i)_{x,y}v_i(x)v_i(y)
\]

\[
\leq 2 \sum_i \|W \circ \Delta_i \| \|v_i\|^2
\]

\[
\leq 2 \max_i \|W \circ \Delta_i \|
\]

The last equation is true because \( \sum_i \|v_i\|^2 = 1 \).

Exercise 31. Prove that \( \sum_i \|v_i\|^2 = 1 \).

From Sec. 4.1 and the bound on change through every query, we get the following theorem.

Theorem 1. Given a function \( f : \{0,1\}^n \rightarrow \{0,1\} \) and a \( 2^n \times 2^n \) matrix \( W \) (positive entry-wise), such that \( W_{x,y} = 0 \) if \( f(x) = f(y) \). The query complexity of \( f \) is,

\[
\Omega \left( \frac{\|W\|}{\max_i \|W \circ \Delta_i\|} \right).
\]

To give a lower bound, we need to come up with a good \( W \). Generally the idea is to put more weight on pairs which differ on small number of bits but there function value is different. You can prove a lower bound on \( OR \) using this theorem, the weight on input pairs can be figured out by the proof of optimality of Grover search. This is given as an exercise in the assignment.
Exercise 32. Draw the circuit for the Grover iteration.

Exercise 33. Run the Grover’s algorithm for $m \leq \frac{n}{2}$ marked elements. How many queries do we need to find a marked element?

Exercise 34. What if we have more than $\frac{n}{2}$ marked elements?

Exercise 35. In how many queries can we find all the marked elements?

Exercise 36. Why is amplitude amplification a generalization of Grover search? Carry out the details of amplitude amplification.

Exercise 37. Read section 6.2 from the book of Nielsen and Chuang [1].

Exercise 38. Prove that the query complexity of OR is $\Omega(\sqrt{n})$ using Thm. [1]

References