Lecture 11: Applications of Grover's Search Algorithm

Scribed by: Yuan-Chung Cheng

Department of Chemistry, MIT

October 9, 2003

In this lecture we will cover several applications of Grover's search algorithm, and show that Grover's search algorithm is optimal.

Grover's search algorithm: $\exists$ function **O** such that

$$\begin{aligned} \mathbf{O}|x\rangle &= -|x\rangle \quad if\, x \in \mathbf{T}, \\ \mathbf{O}|x\rangle &= |x\rangle \quad otherwise. \end{aligned}$$

We want to find elements in the target set **T**. If there are N $|x\rangle$'s and M targets, after $c \cdot \sqrt{\frac{N}{M}}$ iterations of **G**, most amplitude will be in the target set. The following operator **G** is performed in each iteration:

$$\mathbf{G} = \mathbf{H}^{\otimes n}(2|0\rangle\langle 0| - \mathbf{I})\mathbf{H}^{\otimes n} \cdot \mathbf{O},$$

and the initial state is prepared in $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle$.

In addition to perform the search of a target in a set of elements, the Grover's search algorithm has other possible applications. Here we will demonstrate how to use Grover's search algorithm to speed up classical algorithms, and do target counting.

Speed up classical algorithms

NP problems are those for which a solution is easy to check, but not necessarily easy to find. Examples of NP problems are:

Travel Salesman Problem (TSP): Having n cities and $\begin{pmatrix} n \\ 2 \end{pmatrix}$ distances between them, we want to find a tour of the cities of length at most D.

3-SAT: Given a boolean formula of totally n variables in conjunctive normal form with at most 3 variables in each clause, ex. $(x_1 \vee x_3 \vee x_7) \wedge (x_1 \vee \bar{x_5} \vee x_9) \wedge (x_2 \vee \bar{x_3} \vee x_{11}) \ldots$, we want to know if there exist a set of $\{x_i, i = 1 \ldots n\}$ such that the whole formula is satisfied.

Using the 3-SAT problem as an example, classically we have to search exhaustively and try every set of values, therefor the algorithm takes 2^n time. Here we will show that the problem can be done using the Grover's search algorithm in $2^{n/2}$ time.

A better way to start is to find a maximal set of disjoint clauses in the formula. We can rename the variables and write the formula as

$$(x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6) \wedge \ldots \wedge (x_{m-2} \vee x_{m-1} \vee x_m) \wedge \ldots (x_i \vee x_{n-1} \vee x_n),$$

where $1 < m < n$, and $1 < i < m$. In this form, we have a disjoint clauses set (from x_1 to x_m), and the rest of the formula are 2-SAT. Since we know polynomial time algorithm for 2-SAT problems, we can easily solve the left-over part. Notice that we need to try only 7 values for each clause,and there are at most $\frac{n}{3}$ clauses. The time is $7^{n/3} \approx 2^{.93n}$. If we apply Grover's search algorithm, we can do with time in $O(2^{.93n/2})$. This is almost close to the best classical algorithm $O(2^{.43n})$. This demonstrates that applying Grover's algorithm to a relatively simple classical algorithm can gain substantial speed up (however, not exponential speed up). It is possible that the Grover's algorithm and be used on best classical algorithm can gain speed up that can not be done classically.

Approximate counting: How many solutions are there?

Grover's algorithm can be combined with the quantum phase estimation algorithm to approximately count the number of targets in the set, i.e. the value of M. Classically, suppose you need to sample A values to get $M \pm \varepsilon M$. The expected number in target set is $A \cdot \frac{M}{N}$, standard deviation is $\sqrt{A \cdot \frac{M}{N}}$. The estimate of M, M_{est}, is the number of targets you found times N/A.

$$\begin{aligned} M_{est} &= \tfrac{N}{A} \cdot number\, found \\ &= \tfrac{N}{A} A \tfrac{M}{N} \pm \tfrac{N}{A}\sqrt{\tfrac{AM}{N}} \\ &= M \pm \sqrt{\tfrac{NM}{A}} \\ &= M\left(1 \pm \sqrt{\tfrac{N}{AM}}\right) \end{aligned}$$

Therefore, $M(1+\varepsilon) = M(1 \pm \sqrt{\frac{N}{AM}})$. To get M within $\pm\varepsilon$ range, we need to sample A times, where $A \approx \frac{N}{M\varepsilon^2}$. The efficient of the estimation is $\sim 1/\varepsilon^2$.

We can use Grover's algorithm to improve the performance. Recall that in the Grover's search algorithm, if we define

$$\begin{aligned} |\alpha\rangle &= \tfrac{1}{\sqrt{N-M}} \sum_{x \notin \mathbf{T}} |x\rangle \\ |\beta\rangle &= \tfrac{1}{\sqrt{M}} \sum_{x \in \mathbf{T}} |x\rangle, \end{aligned}$$

what Grover's algorithm does in each iteration is to rotate the state vector in the $|\alpha\rangle, |\beta\rangle$ basis counter-clockwise for θ angle, where $\frac{\theta}{2} = \arcsin(\sqrt{\frac{M}{N}})$. In the $|\alpha\rangle, |\beta\rangle$ basis, we can write the $\mathbf{G}$ operator as a rotation matrix (for $N \gg M$):

$$\mathbf{G} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

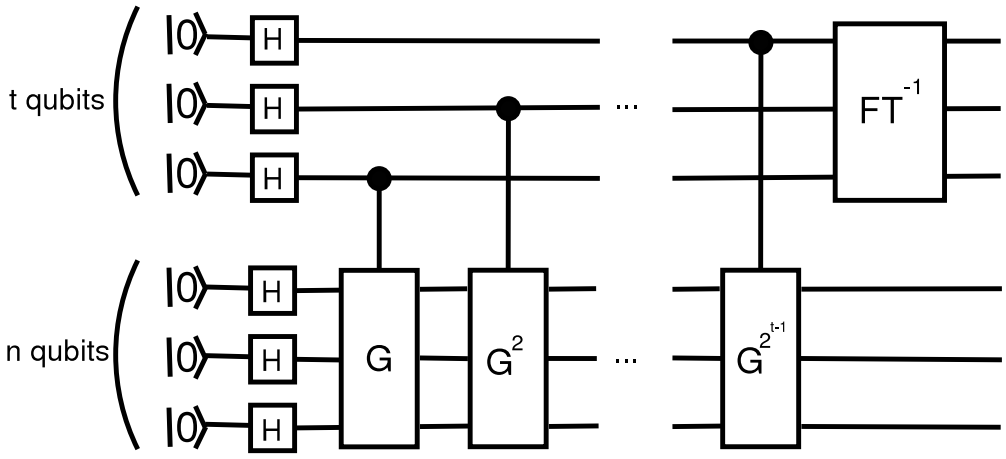

Figure 1: Circuit for quantum counting.

The eigenvalues of the matrix are $e^{\pm i\theta}$. Therefore, we can use phase estimation algorithm to obtain the phase factor θ, and compute the number of targets in the set. Figure 1. shows the circuit for approximate quantum counting to t qubits accuracy on a quantum computer.

We now analyze the efficiency of the quantum counting algorithm. Suppose we get θ to t bits accuracy, i.e. $\theta_{est} = \theta \pm \frac{1}{2^t}$. We need 2^t calls to the function. By the definition of θ, we have

$$\sin^2 \frac{\theta}{2} = \frac{M}{N}.$$

To estimate the error, we take the derivative on both side to obtain the relationship between the change of θ, $\delta\theta$, and the change of M, δM:

$$2 \sin \frac{\theta}{2} \cos \frac{\theta}{2} \cdot \delta\theta = \frac{\delta M}{N}.$$

Using $N \gg M$, and $\theta \ll 1$, we obtain

$$\delta M = 2N\delta\theta \sqrt{\frac{M}{N}} \cos \frac{\theta}{2} \leq 2\delta\theta\sqrt{MN}.$$

We define $\varepsilon M = \delta M \approx 2\delta\theta\sqrt{MN}$, so $\delta\theta = \sqrt{\frac{M}{N}} \cdot \frac{2}{\varepsilon}$. Therefore, in terms of ε, the number of calls needed to get to the accuracy of ε is $\sim \sqrt{\frac{M}{N}} \cdot \frac{2}{\varepsilon}$. This is the square-root of what we need in the classical algorithm.

Grover's search algorithm is optimal

Suppose we have a new quantum algorithm that can find state $|x\rangle$. We define $\mathbf{O}_x$ as the oracle of finding state $|x\rangle$:

$$\mathbf{O}_x = \begin{cases} |x\rangle & \rightarrow & -|x\rangle \\ |y\rangle & \rightarrow & |y\rangle \quad if\ y \neq x \end{cases}$$

Let $|\psi\rangle$ be the starting state of the algorithm. One can write any algorithm of finding state $|x\rangle$ as

$$\sqrt{1-\varepsilon}|x\rangle + \sqrt{\varepsilon}|g\rangle \approx \mathbf{U}_n\mathbf{O}_x \ldots \mathbf{U}_3\mathbf{O}_x\mathbf{U}_2\mathbf{O}_x\mathbf{U}_1\mathbf{O}_x|\psi\rangle,$$

where $|g\rangle$ is composed of states $\perp$ to $|x\rangle$. For a successful algorithm, the outcome of the operations should be very close to the target state, that is, ε has to be very small.

Let's also define

$$\begin{aligned} |\psi_k^x\rangle &= \mathbf{U}_k\mathbf{O}_x\mathbf{U}_{k-1}\mathbf{O}_x \ldots \mathbf{U}_1\mathbf{O}_x|\psi\rangle, \\ |\psi_k\rangle &= \mathbf{U}_k\mathbf{U}_{k-1} \ldots \mathbf{U}_1|\psi\rangle. \end{aligned}$$

Notice that $|\psi_k^x\rangle$ and $|\psi_k\rangle$ differ by the oracle operator $\mathbf{O}_x$ that separates the target state from other states. From this, we can define the Euclidean distance between these two states after k operations:

$$D_k = \sum_x \||\psi_k^x\rangle - |\psi_k\rangle\|^2. \tag{1}$$

This distance can serve as a measurement of how much this algorithm can made to distinguish $|x\rangle$ and $|y\rangle$ states. We will show that $D_k \leq O(k^2)$. To solve Grover's problem, one needs at least $D_{least} \approx O(N)$. Therefore, at least $k \sim \sqrt{N}$ is necessary, which is what Grover's search algorithm does.

Consider the Euclidean distance of the $k+1$ states:

$$\begin{aligned} D_{k+1} &= \sum_x \|\mathbf{U}_{k+1}\mathbf{O}_x|\psi_k^x\rangle - \mathbf{U}_{k+1}|\psi_k\rangle\|^2 \\ &= \sum_x \|\mathbf{O}_x|\psi_k^x\rangle - |\psi_k\rangle\|^2 \\ &= \sum_x \|\mathbf{O}_x(|\psi_k^x\rangle - |\psi_k\rangle) + (\mathbf{O}_x - \mathbf{I})|\psi_k\rangle\|^2, \end{aligned}$$

where we have applied the property that the unitary operator $\mathbf{U}_{k+1}$ preserves the norm. Use the inequality $\|b+c\|^2 \leq \| b \|^2 + 2 \| b \| \cdot \| c \| + \| c \|^2$, we obtain

$$\begin{aligned} D_{k+1} &\leq \sum_x \left[\|\mathbf{O}_x(|\psi_k^x\rangle - |\psi_k\rangle)\|^2 + 2\|\mathbf{O}_x(|\psi_k^x\rangle - |\psi_k\rangle)\| \cdot \|(\mathbf{O}_x - \mathbf{I})|\psi_k\rangle\| + \|(\mathbf{O}_x - \mathbf{I})|\psi_k\rangle\|^2\right] \\ &\leq \sum_x \||\psi_k^x\rangle - |\psi_k\rangle\|^2 + \sum_x 2 \cdot \||\psi_k^x\rangle - |\psi_k\rangle\| \cdot \|(\mathbf{O}_x - \mathbf{I})|\psi_k\rangle\| + \sum_x \|(\mathbf{O}_x - \mathbf{I})|\psi_k\rangle\|^2 \\ &\leq D_k + 4 \cdot \sum_x \||\psi_k^x\rangle - |\psi_k\rangle\| \cdot \|\langle x|\psi_k\rangle \cdot |x\rangle\| + 4, \end{aligned} \tag{2}$$

where we have used the property that the unitary operator $\mathbf{O}_x$ preserves the norm and $\mathbf{O}_x - \mathbf{I} = -2|x\rangle\langle x|$. Now apply the Cauchy's inequality,

$$\sum_x \||\psi_k^x\rangle - |\psi_k\rangle\| \cdot \|\langle x|\psi_k\rangle \cdot |x\rangle\| \leq \sum_x \sqrt{\||\psi_k^x\rangle - |\psi_k\rangle\|^2} \cdot \sqrt{\|\langle x|\psi_k\rangle \cdot |x\rangle\|^2},$$

in Eq. (2), we obtain the recursion relation

$$D_{k+1} \leq D_k + 4\sqrt{D_k} + 4.$$

Assume $D_k \leq 4k^2$, we get

$$D_{k+1} \leq (4k^2 + 8k + 4) = 4(k+1)^2.$$

Hence we have shown that the best you can do to distinguish the target states and other states using j operations is $D_j \leq 4 \cdot j^2$. Note that the least distance you need to distinguish target states in totally N states is $D_{least} \approx O(N)$. Therefore, at least $k \sim \sqrt{N}$ is necessary to solve the Grover's problem. This shows that you can not do better than Grover's algorithm.