NP-complete problems and Quantum Search

0.1 NP-Completeness

Consider SAT, the prototypical example of an NP-complete problem. An instance of this problem consists of a Boolean function $f(x_1,\ldots,x_n) = c_1 \wedge \ldots \wedge c_m$; the SAT problem asks you to determine whether there exists a satisfying assignment—that is, an input $(a_1,\ldots,a_n)$ such that $f(a_1,\ldots,a_n) = 1$. UNIQUE-SAT is a variant of SAT that poses the same problem with the restriction that f must have zero or one satisfying assignments, but no more. As it turns out, there is a randomized reduction from SAT to UNIQUE-SAT; thus, the two problems are equally hard.

We'll use the black box model when considering this problem. In this model, we know that either $f \equiv 0$ or there exists exactly one a such that $f(a) = 1$, where a is chosen uniformly at random. That is, f is treated as a black box; we can make queries to f, but we have no access to the Boolean formula itself. Equivalently we can represent f by a table of $N = 2^n$ entries where either none or exactly one entry is 1. Ideally we want a quantum algorithm that solves this problem in time $O(\text{poly}(n)) = O(poly - log(N))$.

Can a quantum computer solve this problem by going into a superposition of all exponentially many possible truth assignments? To answer this question precisely, let us define the black box query model:

0.2 The quantum black box model

Here's the problem: You are given a boolean function $f : \{1,\ldots,N\} \to \{0,1\}$, and are promised that for exactly one $a \in \{1,\ldots,N\}$, $f(a) = 1$. Think of this as a table of size N, where exactly one element has value 1, and all the others are 0. Since we assume f can be computed classically in polynomial time, we can also compute it in superposition:

$$\sum_x \alpha_x |x\rangle |0\rangle \to \sum_x \alpha_x |x\rangle |f(x)\rangle$$

Now, we might as well assume f is a black box or oracle. All we need to do is design an algorithm that finds $a : f(a) = 1$.

0.3 The Hybrid Argument

For the purposes of this discussion, we want to separate the quantum algorithm itself from the function f. We assume that the quantum algorithm is infinitely powerful (i.e., it can do any computation in one step) and focus instead on the number of queries it must make to f. All queries to f occur in superposition; that is, a single query on $\sum_x \alpha_x |x\rangle |0\rangle$ yields the output $\sum_x \alpha_x |x\rangle |f(x)\rangle$.

Theorem 0.1: *In the black box model, any quantum algorithm for determining whether there exist $x_1,\ldots,x_n$ such that $f(x_1,\ldots,x_n) = 1$ must make $\Omega(\sqrt{N})$ queries to f.*

Proof:

Consider any quantum algorithm A for solving this search problem. First do a test run of A on the function $f \equiv 0$. Let T be the number of queries that A makes to f, and let $\alpha_{x,t}$ be the amplitude with which A queries x at time t (that is, the query at time t is $\sum_x \alpha_{x,t} |x\rangle$). Now, define the query magnitude of x to be $\sum_t |\alpha_{x,t}|^2$. The expectation value of the query magnitude of x is $E_x\left(\sum_t |\alpha_{x,t}|^2\right) = T/N$. Thus $\min_x \left(\sum_t |\alpha_{x,t}|^2\right) \leq T/N$. Let z be the input at which this minimum occurs; then by the Cauchy-Schwarz inequality, $\sum_t |\alpha_{z,t}| \leq T/\sqrt{N}$.

1

Let $|\phi_t\rangle$ be the states of A_f after the t-th step. Now run the algorithm A on the function g such that $g(z) = 1$ and for all $y \neq z$, $g(y) = 0$. Suppose the final state of A_g is $|\psi_T\rangle$. By the claim that follows, $|\phi_T\rangle - |\psi_T\rangle = |E_0\rangle + \ldots + |E_{T-1}\rangle$ where $\||E_t\rangle\| \leq \sqrt{2}|\alpha_{z,t}|$. Using the triangle inequality and the inequality proved above, we have $\||\phi_T\rangle - |\psi_T\rangle\| \leq \sum_t \||E_t\rangle\| \leq \sqrt{2}\sum_t |\alpha_{z,t}| \leq T\sqrt{2/N}$. This implies that the two states can be distinguished with probability at most $O(T/\sqrt{N})$ by any measurement. Thus any quantum algorithm that distinguishes f from g with constant probability of success must make $\Omega(\sqrt{N})$ queries.

□

Claim: $|\psi_T\rangle = |\phi_T\rangle + |E_0\rangle + |E_1\rangle + \ldots + |E_{T-1}\rangle$, where $\||E_t\rangle\| \leq \sqrt{2}|\alpha_{z,t}|$.

Proof:

Consider two runs of the algorithm A, which differ only on the t-th step. The first run queries the function f on the first t steps and queries g for the remaining $T-t$ steps; the second run queries f on the first $t-1$ steps and g for the remaining $T-t+1$ steps. After the first $t-1$ steps, both runs are in state $|\phi_t\rangle$. On the t-th step, one run queries f and the other queries g. The outputs of these queries differ only on the amplitude of the two basis vectors $|z\rangle|0\rangle$ and $|z\rangle|1\rangle$, so overall the output vectors differ by at most $\sqrt{2}|\alpha_{z,t}|$. Thus, at the end of the t-th step, the state of the first run is $|\phi_t\rangle$, whereas the state of the second run is $|\phi_t\rangle + |F_t\rangle$, where $\||F_t\rangle\| \leq \sqrt{2}|\alpha_{z,t}|$. Now if U is the unitary transform describing the remaining $T-t$ steps (of both runs), then the final state after T steps for the two runs are $U|\phi_t\rangle$ and $U(|\phi_t\rangle + |F_t\rangle)$, respectively. The latter state can be written as $U|\phi_t\rangle + |E_t\rangle$, where $|E_t\rangle = U|F_t\rangle$. Since unitary transformations preserve length, we know that $\||E_t\rangle\| \leq \sqrt{2}|\alpha_{z,t}|$. Thus, the effect of switching the queried function only on the t-th step can be described by an "error" $|E_t\rangle$ in the final state of the algorithm, where $\||E_t\rangle\| \leq \sqrt{2}|\alpha_{z,t}|$.

We can transform the run A_f to A_g by a succession of T changes of the kind described above. Overall, the difference between the final states of A_f and A_g is $|E_0\rangle + |E_1\rangle + \ldots + |E_{T-1}\rangle$, where $\big\||E_t\rangle\big\| \leq \sqrt{2}|\alpha_{z,t}|$.

□

Finally, it is useful to consider where this factor of $\sqrt{N}$ comes from. In the worst case, we query z with amplitude $1/\sqrt{N}$ at each time step. The vectors that indicate the differences at each step could all be orthogonal, in which case the total distance is the sum of the squares of each vector's length, which is about N. However, if all vectors are in the same direction, the total distance is the sum of the length of each vector, which is approximately $\sqrt{N}$. Grover's algorithm, which we will describe next, demonstrates that it is possible to align all of these vectors and achieve the factor of $\sqrt{N}$.

Grover's Algorithm:

Recall that you are given a program for computing $f : \{1,\ldots,N\} \to \{0,1\}$, such that there is a unique a : $f(a) = 1$. Think of this as a table of size N, where exactly one element has value 1, and all the others are 0. Since we assume f can be computed classically in polynomial time, we can also compute it in superposition:

$$\sum_x \alpha_x |x\rangle|0\rangle \to \sum_x \alpha_x |x\rangle|f(x)\rangle$$

[1]The Cauchy-Schwarz inequality says that for two vectors a and b of length T, $(\sum_t a_t b_t)^2 \leq \left(\sum_t a_t^2\right)\left(\sum_t b_t^2\right)$. If we let $b_t = 1$ for all t, then we have $(\sum_t a_t)^2 \leq T\sum_t a_t^2$. Thus, if $\sum_t |\alpha_{z,t}|^2 \leq T/N$, then $(\sum_t |\alpha_{z,t}|)^2 \leq T^2/N$.

As we saw before, we can use circuit for f to put information about $f(x)$ in the phase by effecting the transformation:

$$\sum_x \alpha_x |x\rangle \rightarrow \sum_x \alpha_x (-1)^{f(x)} |x\rangle$$

Here is another way of creating this phase state:

$$\begin{aligned}
\sum_x \alpha_x |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) &\mapsto \sum_x \alpha_x \left(\frac{|x\rangle |f(x)\rangle - |x\rangle |\overline{f(x)}\rangle}{\sqrt{2}} \right) \\
&= \sum_x \alpha_x |x\rangle \left(\frac{|f(x)\rangle - |\overline{f(x)}\rangle}{\sqrt{2}} \right) \\
&= \sum_x \alpha_x |x\rangle (-1)^{f(x)} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)
\end{aligned}$$

Now, we might as well assume f is a black box or oracle. All we need to do is design an algorithm that finds $a : f(a) = 1$.

0.4 Grover's algorithm

Grover's algorithm finds a in $O(\sqrt{N})$ steps. Consider the N dimensional Hilbert space spanned by $|1\rangle, \ldots, |N\rangle$. We wish to find $|a\rangle$. There is a state that we can create: $|u\rangle = \sum_x \frac{1}{\sqrt{N}} |x\rangle$. Consider the two dimensional subspace spanned by $|a\rangle$ and $|u\rangle$. Let $|e\rangle$ be the state orthogonal to $|a\rangle$ in this subspace. Let θ be the angle between $|u\rangle$ and $|e\rangle$. Then $sin\ \theta = 1/\sqrt{N}$ and therefore $\theta \approx 1/\sqrt{N}$. See Figure 1 for an illustration of these vectors.

$|a\rangle$ is the target, so we want to increase θ. But how?

One way to rotate a vector is to make two reflections. In particular, we can rotate a vector $|v\rangle$ by 2θ by reflecting about $|e\rangle$ and then reflecting about $|u\rangle$. This transformation is also illustrated in Figure 1.

Each step of our algorithm is a rotation by 2θ (we discuss the implementation below). This means that we need $\frac{\pi/2}{2\theta}$ iterations for the algorithm to complete. Now, what's θ?

$$\langle \psi_0 | a \rangle = \cos(\pi/2 - \theta) = \sin(\theta) = \frac{1}{\sqrt{N}}$$

Since $\sin\theta \approx \theta$, we know that $\theta \approx \frac{1}{\sqrt{N}}$. Thus, we need $O(\sqrt{N})$ iterations for the algorithm to complete. In the end, we get very close to $|a\rangle$, and then with high probability, a measurement of the state yields a.

How do you implement the two reflections?

1. Reflection about $|e\rangle$ is easy. We can reflect about the hyperplane orthogonal to $|a\rangle$ by flipping the phase of the component in the direction of $|a\rangle$; i.e. carry out the transformation

$$\sum_x \alpha_x |x\rangle \rightarrow \sum_x \alpha_x (-1)^{f(x)} |x\rangle$$

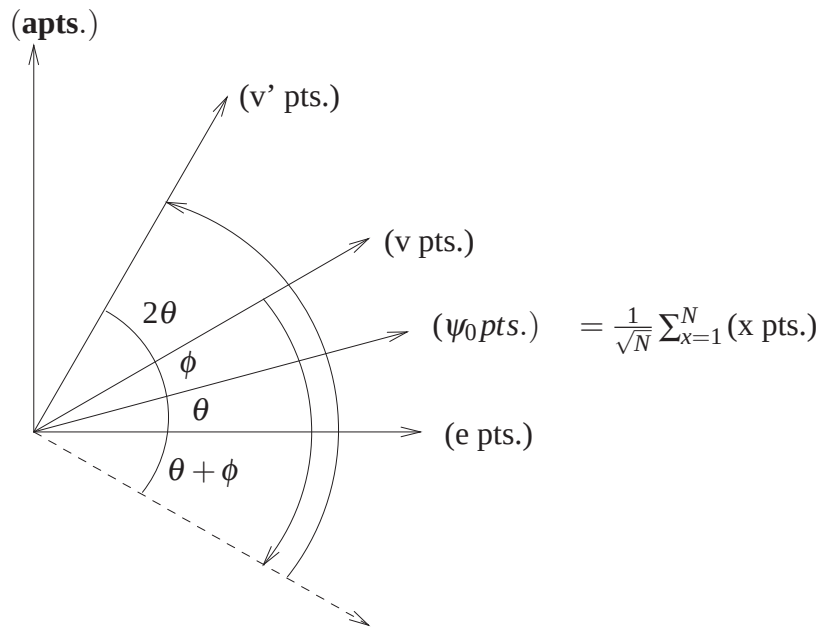

Figure 1: To rotate $|v\rangle$ by 2θ to $|v'\rangle$, we reflect around $|e\rangle$ and then reflect around $|\psi_0\rangle$.

2. For the reflection about $|u\rangle$, we will actually reflect about $|u\rangle$ in the N dimensional space as follows: apply the Hadamard transform $H^{\otimes n}$ to transform $|u\rangle$ to $|0^n\rangle$. Now apply a phase flip if the register contents are anything other than $|0^n\rangle$. And apply the Hadamard transform to switch back from the Hadamard basis.

0.5 Another approach

Let's look at the search algorithm differently, with all superpositions. The rotation about $|u\rangle$, D, is an "inversion about the mean":

(a) For $N = 2^n$, D can be decomposed and rewritten as:

$$
\begin{aligned}
D &= H_N \begin{pmatrix} -1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} H_N \\
&= H_N \left(\begin{pmatrix} -2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} + I \right) H_N \\
&= H_N \begin{pmatrix} -2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} H_N + I \\
&= \begin{pmatrix} -2/N & -2/N & \cdots & -2/N \\ -2/N & -2/N & \cdots & -2/N \\ \vdots & \vdots & \ddots & \vdots \\ -2/N & -2/N & \cdots & -2/N \end{pmatrix} + I \\
&= \begin{pmatrix} -2/N+1 & -2/N & \cdots & -2/N \\ -2/N & -2/N+1 & \cdots & -2/N \\ \vdots & \vdots & \ddots & \vdots \\ -2/N & -2/N & \cdots & -2/N+1 \end{pmatrix}
\end{aligned}
$$

Observe that D is expressed as the product of three unitary matrices (two Hadamard matrices separated by a conditional phase shift matrix). Therefore, D is also unitary. Regarding the implementation, both the Hadamard and the conditional phase shift transforms can be efficiently realized within $O(n)$ gates.

(b) Consider D operating on a vector $\left|\alpha\right\rangle$ to generate another vector $\left|\beta\right\rangle$:

$$
D \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_i \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_i \\ \vdots \\ \beta_N \end{pmatrix}
$$

If we let μ be the mean amplitude, then the expression $2\mu - \alpha_i$ describes a reflection of α_i about the mean. Thus, the amplitude of $\beta_i = -\frac{2}{N}\sum_j \alpha_j + \alpha_i = -2\mu + \alpha_i$ can be considered an "inversion about the mean" with respect to α_i.

The quantum search algorithm iteratively improves the probability of measuring a solution. Here's how:

(a) Start state is $\left|\psi_0\right\rangle = \sum_x \frac{1}{\sqrt{N}}\left|x\right\rangle$

(b) Invert the phase of $\left|a\right\rangle$ using f

(c) Then invert about the mean using D

(d) Repeat steps 2 and 3 $O(\sqrt{N})$ times, so in each iteration α_a increases by $\frac{2}{\sqrt{N}}$

This process is illustrated in Figure 2.

Suppose we just want to find a with probability $\frac{1}{2}$. Until this point, the rest of the basis vectors will have amplitude at least $\frac{1}{\sqrt{2N}}$. In each iteration of the algorithm, α_a increases by at least $\frac{2}{\sqrt{2N}} = \sqrt{\frac{2}{N}}$. Eventually, $\alpha_a = \frac{1}{\sqrt{2}}$. The number of iterations to get to this α_a is $\leq \sqrt{N}$.

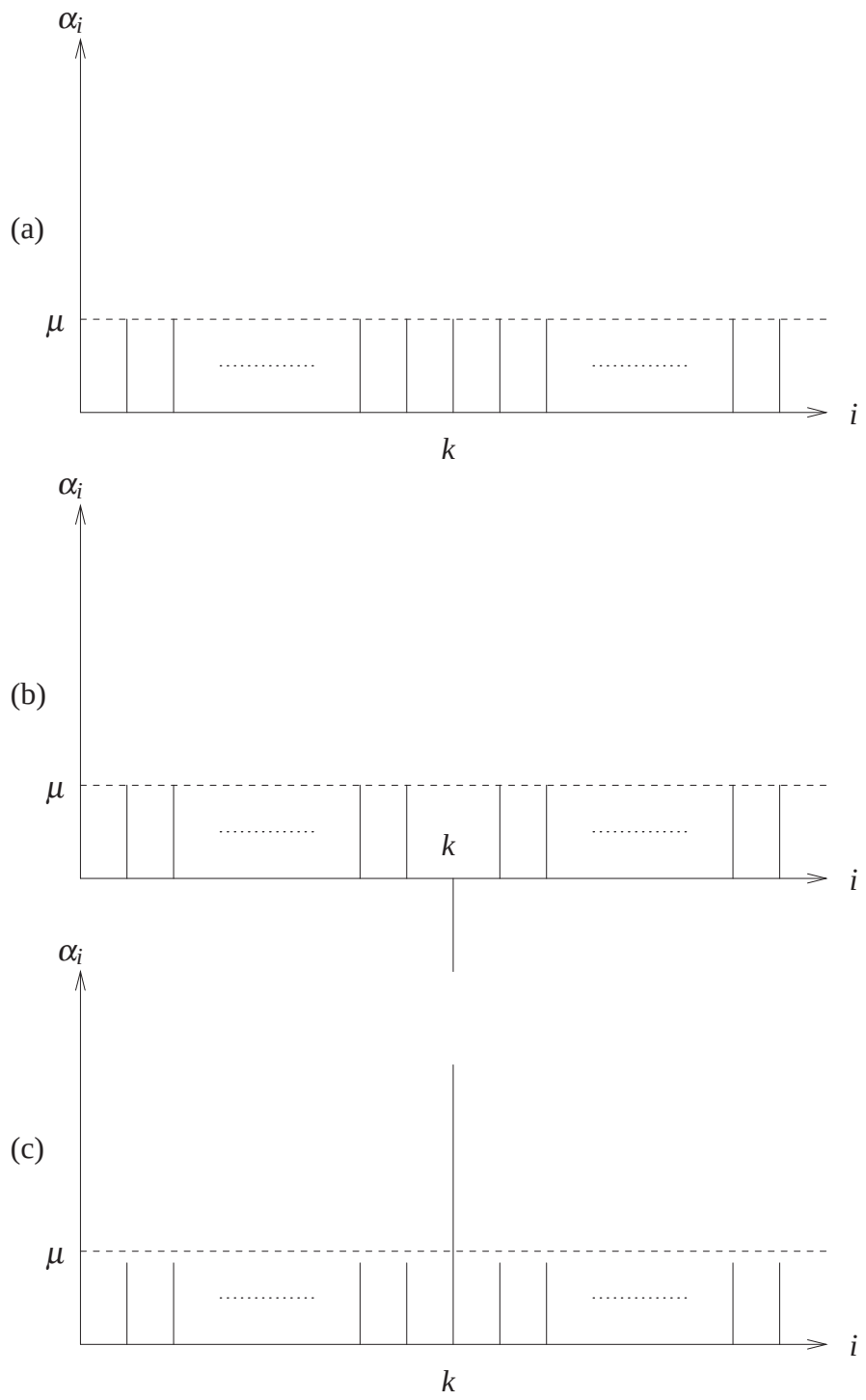

Figure 2: The first three steps of Grover's algorithm. We start with a uniform superposition of all basis vectors in (a). In (b), we have used the function f to invert the phase of α_k. After running the diffusion operator D, we amplify α_k while decreasing all other amplitudes.