Quantum algorithms (CO 781, Winter 2008)
Prof. Andrew Childs, University of Waterloo
LECTURE 19: Adiabatic optimization

Having established the quantum adiabatic theorem, we will now see how it can be applied to solve optimization problems. After describing the general framework, we will see how this approach gives an alternative $O(\sqrt{N})$-time algorithm for unstructured search.

An adiabatic optimization algorithm Many computational problems can be cast as the minimization of some cost function. For concreteness, suppose we are given a function

$$h : \{0,1\}^n \to \mathbb{R} \tag{1}$$

from strings of n bits to real numbers. A natural question is, does there exist a string $z \in \{0,1\}^n$ such that $h(z) = 0$? Or alternatively, can you find a string z that globally minimizes $h(z)$? In general, such questions can be very difficult to answer—the first problem is NP-complete, and the second is NP-hard. (For example, $h(z)$ could be the number of clauses violated by some CNF formula.) But of course, specific instances of such a problem can be more tractable than the general case.

The quantum adiabatic theorem suggests a natural approach to minimizing functions such as (1). The basic idea is to encode the solutions of the minimization problem in the ground state of a Hamiltonian and to adiabatically evolve into this ground state, starting from a known ground state. According to the adiabatic theorem, the probability of finding a solution will be high provided the evolution is sufficiently slow.

To cast the problem of minimizing (1) in quantum mechanical terms, consider a Hamiltonian that is diagonal in the computational basis, with eigenvalues $h(z)$:

$$H_P := \sum_{z\in\{0,1\}^n} h(z)\,|z\rangle\langle z| \tag{2}$$

We refer to H_P as the *problem Hamiltonian*, since it corresponds to the problem of minimizing h. Clearly, its ground state consists of strings z such that $h(z)$ is minimized. Therefore, if we could prepare the ground state of H_P, we could solve the minimization problem.

To prepare the ground state of H_P, we will adiabatically evolve from the ground state of a simpler Hamiltonian. Let the the *beginning Hamiltonian* H_B be some Hamiltonian whose ground state is easy to prepare. Then let $H_T(t)$ be a smoothly varying time-dependent Hamiltonian with $H_T(0) = H_B$ and $H_T(T) = H_P$, where T is the total run time of the evolution. Assuming the evolution is sufficiently close to adiabatic, the initial ground state will evolve into a state close to the final ground state, thereby solving the problem.

For any given H_B and H_P, there are many possible choices for the interpolation $H_T(t)$. One simple choice is a time-dependent Hamiltonian of the form

$$H_T(t) = H(t/T) := \big(1 - f(t/T)\big)H_B + f(t/T)H_P \tag{3}$$

where $f(s)$ is a smooth, monotonic function of $s \in [0,1]$ satisfying $f(0) = 0$ and $f(1) = 1$, so that $H(0) = H_B$ and $H(1) = H_P$. In other words, the interpolating function $f(t/T)$ should vary smoothly from 0 to 1 as the time t varies from 0 to T. If $f(s)$ is twice differentiable, and if the

ground state of $H(s)$ is nondegenerate for all $s \in [0,1]$, then the adiabatic theorem guarantees that the evolution will become arbitrarily close to adiabatic in the limit $T \to \infty$. An especially simple choice for this interpolation schedule is the linear interpolation $f(s) = s$, but many other choices are possible.

Finally, how should we choose the beginning Hamiltonian? If we choose an interpolation of the form (3), then H_B clearly should not commute with H_P, or else no evolution will occur. One natural choice for H_B is

$$H_B = -\sum_{j=1}^{n} \sigma_x^{(j)} \tag{4}$$

where $\sigma_x^{(j)}$ is the Pauli x operator on the jth qubit. This beginning Hamiltonian has the ground state

$$|S\rangle := \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} |z\rangle, \tag{5}$$

a uniform superposition of all possible solutions $S = \{0,1\}^n$. But as for the method of interpolation, many other choices for H_B are possible.

To summarize, a quantum adiabatic optimization algorithm works as follows:

1. Prepare the quantum computer in the ground state of the beginning Hamiltonian H_B.
2. Evolve the state with the Hamiltonian $H(t)$ for a total time T, ending with the problem Hamiltonian H_P.
3. Measure in the computational basis.

Step 1 can be performed efficiently if H_B has a sufficiently simple ground state—for example, if it is the state (5). Step 2 can be simulated efficiently on a universal quantum computer, assuming the Hamiltonian is of a suitable form (say, if it is sparse) and the run time T is not too large. Step 3 is straightforward to implement, and will yield a state close to the ground state assuming the simulation of the evolution is sufficiently good and the evolution being simulated meets the conditions of the adiabatic theorem.

The running time and the gap The quantum adiabatic optimization algorithm described above is guaranteed to produce the correct answer with high probability provided the run time is sufficiently large. But how long is long enough? Unfortunately, this question is difficult to answer for almost all interesting problems. However, using the adiabatic theorem, it can at least be rephrased as a statement about spectral properties of the Hamiltonian.

From the adiabatic theorem, we see that the run time depends crucially on the gap $\Delta(s)$ between the ground and first excited states of $H(s)$. Suppose for simplicity that we use linear interpolation between H_B and H_P, i.e., (3) with $f(s) = s$. Clearly

$$\dot{H} = H_P - H_B \tag{6}$$

$$\ddot{H} = 0. \tag{7}$$

Now let

$$\Delta_{\min} := \min_{s \in [0,1]} \Delta(s) \tag{8}$$

be the *minimum gap* between the ground and first excited states. Then we have

Theorem. *Suppose the evolution time satisfies*

$$T \geq \frac{2}{\epsilon}\left[2c_1\frac{\|H_P - H_B\|}{\Delta_{\min}^2} + (3c_1^2 + c_1 + c_3)\frac{\|H_P - H_B\|^2}{\Delta_{\min}^3}\right]. \tag{9}$$

Then $\||\psi(1)\rangle - |\phi(1)\rangle\| \leq \epsilon$.

Recall that to be efficiently simulable, H_B and H_P should not have very large norm. Thus we see that if the minimum gap $\Delta_{\min}$ is not too small, the run time need not be too large. In particular, to show that the adiabatic algorithm runs in polynomial time, it sufficies to show that the minimum gap is only polynomially small, i.e., that $1/\Delta_{\min}$ is upper bounded by a polynomial in n.

Of course, this does not answer the question of whether the adiabatic algorithm runs in polynomial time unless the minimum gap can be estimated. In general, calculating the gap for a particular Hamiltonian is a difficult problem, which makes the adiabatic algorithm difficult to analyze. Nevertheless, there are a few examples of interest for which the gap can indeed be estimated.

Adabatic optimization algorithm for unstructured search Recall that the unstructured search problem for N items can be solved in time $O(\sqrt{N})$ (by Grover's algorithm, or by a quantum walk search algorithm), but has classical query complexity $\Omega(N)$. Let's show that adiabatic optimization can also solve this problem in $O(\sqrt{N})$, demonstrating that it can indeed provide quantum speedup.

Unstructured search is equivalent to minimizing the black box function $h : \{0, 1, \ldots, N-1\} \to \{0, 1\}$ defined as

$$h(z) := \begin{cases} 0 & z \text{ is marked} \\ 1 & z \text{ is unmarked.} \end{cases} \tag{10}$$

For simplicity, let's focus on the case of a single marked item m. Then the problem Hamiltonian (2) corresponding to (10) is

$$H_P = 1 - |m\rangle\langle m|. \tag{11}$$

A natural starting point for adiabatic evolution is the uniform superposition state $|S\rangle$, where now $S = \{0, 1, \ldots, N-1\}$. Then a particularly simple choice of the beginning Hamiltonian is the projector onto states orthogonal to $|S\rangle$,

$$H_B = 1 - |S\rangle\langle S|. \tag{12}$$

(Note that such a beginning Hamiltonian is not a good choice in general, since one can show that it allows at most a quadratic speedup over brute force search. We could instead use the initial Hamiltonian (4) for unstructured search, corresponding to adiabiatic optimization over the hypercube, but the analysis would be more complicated.) For this beginning Hamiltonian, $|S\rangle$ is the ground state, with energy 0, and all orthogonal states have energy 1. Finally, suppose we use the interpolation Hamiltonian (3), so that

$$H(s) = 1 - \big[\big(1 - f(s)\big)|S\rangle\langle S| + f(s)|m\rangle\langle m|\big] \tag{13}$$

for some as yet undetermined function $f : [0, 1] \to [0, 1]$.

Just as in Grover's algorithm, $H_T(t)$ acts nontrivially only on the subspace spanned by $|m\rangle$ and $|S\rangle$, making its spectrum straightforward to calculate. Working in the $\{|m\rangle, |m^\perp\rangle\}$ basis we

defined when analyzing the continuous-time quantum walk algorithm for unstructured search on the complete graph (with $|m^\perp\rangle = (|S\rangle - a|m\rangle)/\sqrt{1-a^2}$, where $a := \langle S|m\rangle = 1/\sqrt{N}$),

$$H = \begin{pmatrix} (1-f)(1-a^2) & -(1-f)a\sqrt{1-a^2} \\ -(1-f)a\sqrt{1-a^2} & 1-(1-f)(1-a^2) \end{pmatrix} \tag{14}$$

$$= \frac{1}{2} - (1-f)a\sqrt{1-a^2}\,\sigma_x + \Big[(1-f)(1-a^2) - \frac{1}{2}\Big]\sigma_z. \tag{15}$$

Then it is straightforward to compute the eigenvalues

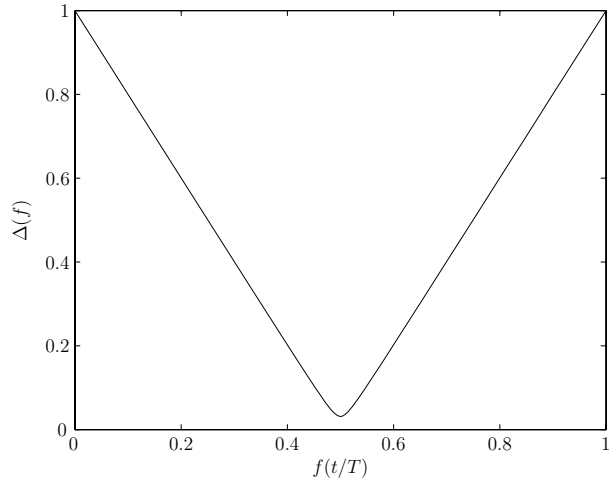
$$E_0 = \frac{1}{2}\Big(1 - \sqrt{1-4f(1-f)(1-a^2)}\Big) \tag{16}$$

$$E_1 = \frac{1}{2}\Big(1 + \sqrt{1-4f(1-f)(1-a^2)}\Big). \tag{17}$$

In other words, the gap between the ground and first excited states is

$$\Delta = \sqrt{1-4f(1-f)(1-a^2)}. \tag{18}$$

For example, with $N = 1000$, we have the following:



In general, the minimum value occurs at $f = 1/2$, where we have $\Delta_{\min} = a = 1/\sqrt{N}$.

To finish specifying the algorithm, we must choose a particular interpolation function (or *schedule*) $f(s)$. The simplest choice is to use the linear interpolation $f(s) = s$, but it turns out that this simple choice does not work. Applying (9), which pessimistically depends solely on the minimum value of the gap, only shows it is sufficient to take $T = O(1/\Delta^3) = O(N^{3/2})$. But even if we use the full adiabatic theorem, we only find that it is sufficient to take the run time to be large compared to

$$\int_0^1 \frac{\mathrm{d}f}{\Delta^3} = \int_0^1 \frac{\mathrm{d}f}{[1-4f(1-f)(1-a^2)]^{3/2}} = \frac{1}{a^2} = N. \tag{19}$$

While the adiabatic theorem only gives an upper bound on the running time, it turns out that the bound is essentially tight in this case: with linear interpolation, the run time must be $\Omega(N)$ for the evolution to remain approximately adiabatic.

However, we can do better by choosing a different interpolation schedule $f(s)$. Intuitively, since the gap is smallest when s is close to $1/2$, we should evolve more slowly for such values. The fact

that the gap is only of order $1/\sqrt{N}$ for values of $|f - 1/2|$ of order $1/\sqrt{N}$ ultimately means that it is possible to choose a schedule for which a total run time of $O(\sqrt{N})$ suffices. Since we should evolve most slowly when the gap is smallest, it is reasonable to let $\dot{f} \propto \Delta^p$ for some power p. For concreteness, we will use $p = 3/2$, although any $p \in (1, 2)$ would work.

If we let

$$\dot{f} = \alpha \Delta^{3/2}, \tag{20}$$

then the coefficient α is fixed by the equation $\int_0^1 \mathrm{d}s = \int_0^1 \mathrm{d}f/\dot{f} = 1$, i.e.,

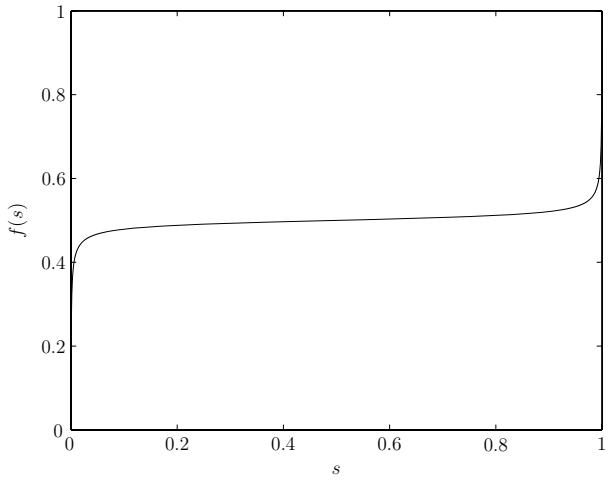
$$\alpha = \int_0^1 \frac{\mathrm{d}f}{\Delta^{3/2}} \tag{21}$$

$$= \int_0^1 \frac{\mathrm{d}f}{[1 - 4f(1-f)(1-a^2)]^{3/4}} \tag{22}$$

$$= \operatorname{Im} B_{(1+i\sqrt{N-1})/2}(1/4, 1/4) \frac{N^{3/4}}{\sqrt{2(N-1)}} \tag{23}$$

$$= 2\sqrt{\pi} \frac{\Gamma(5/4)}{\Gamma(3/4)} N^{1/4} + O(1) \tag{24}$$

where $B_z(a, b)$ denotes the incomplete beta function, and $\Gamma(z)$ denotes the gamma function. Then for example, with $N = 1000$, the schedule obtained by integrating (20) looks as follows:



Now we want to evaluate the terms appearing in the adiabatic theorem. For the first three terms, we need to calculate

$$\|\dot{H}(s)\| = \|\dot{f}(s)(H_P - H_B)\| \tag{25}$$

$$= |\dot{f}(s)| \sqrt{1 - a^2} \tag{26}$$

$$= \alpha \sqrt{1 - a^2} \Delta^{3/2}. \tag{27}$$

The first and second terms are

$$\frac{\|\dot{H}(0)\|}{\Delta(0)^2} = \frac{\|\dot{H}(1)\|}{\Delta(1)^2} = \alpha \sqrt{1 - a^2} \tag{28}$$

$$= O(N^{1/4}), \tag{29}$$

so they will turn out to be negligible. The third term is

$$\int_0^1 \frac{\|\dot{H}\|^2}{\Delta^3}\mathrm{d}s = (1-a^2)\int_0^1 \frac{\dot{f}^2}{\Delta^3}\frac{\mathrm{d}f}{\dot{f}} \tag{30}$$

$$= \alpha(1-a^2)\int_0^1 \frac{\mathrm{d}f}{\Delta^{3/2}} \tag{31}$$

$$= \alpha^2(1-a^2) \tag{32}$$

$$= O(\sqrt{N}). \tag{33}$$

To calculate the final term, we need to compute

$$\|\ddot{H}\| = \|\ddot{f}(s)(H_P - H_B)\| \tag{34}$$

$$= |\ddot{f}(s)|\sqrt{1-a^2} \tag{35}$$

$$= \frac{3}{2}\alpha\Delta^{1/2}|\dot{\Delta}|\sqrt{1-a^2} \tag{36}$$

$$= \frac{3}{2}\alpha\sqrt{1-a^2}\Delta^{1/2}\dot{f}\Big|\frac{\mathrm{d}\Delta}{\mathrm{d}f}\Big|, \tag{37}$$

and

$$\frac{\mathrm{d}\Delta}{\mathrm{d}f} = \frac{2(2f-1)(1-a^2)}{\Delta}. \tag{38}$$

Then we have

$$\int_0^1 \frac{\|\ddot{H}\|}{\Delta^2}\mathrm{d}s = \int_0^1 \frac{\|\ddot{H}\|}{\Delta^2}\frac{\mathrm{d}f}{\dot{f}} \tag{39}$$

$$= \frac{3}{2}\alpha\sqrt{1-a^2}\int_0^1 \frac{1}{\Delta^{3/2}}\Big|\frac{\mathrm{d}\Delta}{\mathrm{d}f}\Big|\,\mathrm{d}f \tag{40}$$

$$= 3\alpha(1-a^2)^{3/2}\int_0^1 \mathrm{d}f\,\frac{|2f-1|}{[1-4f(1-f)(1-a^2)]^{5/4}} \tag{41}$$

$$= \frac{6\alpha(1-a^2)^{3/2}}{\sqrt{a}(1+\sqrt{a})(1+a)} \tag{42}$$

$$= O(\sqrt{N}). \tag{43}$$

Overall, we find a total run time of $T = O(\sqrt{N})$ suffices to make the evolution arbitrarily close to adiabatic.

In the above analysis, it was essential to understand the behavior of the gap as a function of f. In particular, since the spectrum of the Hamiltonian (13) does not depend on which item m is marked, we can choose a schedule that is simultaneously good for all possible marked items. For general instances of adiabatic optimization, this may not be the case.

To implement this adiabatic optimization algorithm for unstructured search in the conventional quantum query model, we must simulate evolution according to this Hamiltonian. Using the fact that $[H_B, H_P] = O(1/\sqrt{N})$, it is possible to perform this simulation using $O(\sqrt{N})$ queries to a black box for $h(z)$.