CS 880: Quantum Information Processing 10/21/2010

Lecture 19: Adiabatic Quantum Computing

Instructor: Dieter van Melkebeek Scribe: Hesam Dashti

Today we will talk about Adiabatic Quantum Computing which is an alternate model for Quantum computing other than the circuit model which we have been working with. The Adiabatic Quantum model is closly relted to the continuous time Quantum walk which we have discussed in the previous lecture. In this lecture we show how one can simulate the Adiabatic model by using the Circuit model and vice versa with a polynomial overhead in time.

1 Adiabatic Evolution

"Adiabatic" is a term from thermodynamics referring to a process in which there is no heat transfer. In the quantum setting the term refers to a process in which the system changes gradually such that it always is close to its equilibrium state. For the Adiabatic Evolution, we are looking at a Quantum system that is described by a Hamiltonian $H(t)$. The evolution is prescribed by Schrödinger's equation:

$$i\frac{d\left|\Psi\right\rangle}{dt} = H(t)\left|\Psi(t)\right\rangle.$$

When the Hamiltonian is constant the evolution of the system is simple: $\left|\Psi(t)\right\rangle = e^{-iHt}\left|\Psi(0)\right\rangle$. But in general, the Hamiltonian could depend on time, in which case the evolution becomes more complex.

Here, we consider a case in which the Hamiltonian depends on time but only changes gradually. In this case the evolution is again relatively simple in the following sense. If the initial state of the system is close to one of the eigenstates of the initial Hamiltonian, provided there is no degeneracy, the state of system follows the eigenstate and at every point in time it will be close to the corresponding eigenstate of the Hamiltonian. If there is degenerecy, then there is no guarantee. The closer to degeneracy we are, the slower the change in the Hamiltonian needs to be.

The egenvalues of the Hamiltonian correspond to the energy level of the system, which are always real numbers. A *ground state* is a state of minimum energy. Starting in the ground state of the system, we remain close to the ground state provided that there is no degeneracy and we move slowly whenever the gap between lowest energy level and the next one becomes small.

We consider the evolution process in an interval $[0, T]$ and rescale it by $s = \frac{t}{T}, t \in [0, T]$. So $s = 0$ would be the initial state and $s = 1$ the end of the process. Let $\left|\Phi(s)\right\rangle$ denote the ground state of $H(s)$ and $\Delta(s)$ the difference of the lowest energy level and the next one.

For a given process profile $H(s), s \in [0, 1]$, the Adiabatic theorem tells us how much time T we need to run the process in order to end up in a state that is no more than ϵ away from the ground state of $H(1)$.

Theorem 1. ***Adiabatic Theorem*** *If we set up our initial state $\left|\Psi(0)\right\rangle$ equal to the ground state of the Hamiltonian $\left|\Phi(0)\right\rangle$ then*

$$\|\left|\Psi(1)\right\rangle - \left|\Phi(1)\right\rangle\| \leq \epsilon$$

provided:

$$T \geq \Omega\left(\frac{1}{\epsilon}\left[\frac{\|\dot{H}(0)\|}{\Delta^2(0)} + \frac{\|\dot{H}(1)\|}{\Delta^2(1)} + \int_0^1 \frac{\|\dot{H}(s)\|}{\Delta^3(s)} + \frac{\|\ddot{H}(s)\|}{\Delta^2(s)} ds\right]\right), \tag{1}$$

where $\dot{H}(s) = \frac{dH(s)}{ds}$ *and* $\ddot{H}(s) = \frac{d^2H(s)}{ds^2}$.

When the Hamiltonian changes linearly as a function of s, the second derivative of it ($\|\ddot{H}(s)\|$) would be zero and the first derivative ($\|\dot{H}(s)\|$) would be a constant, so the value of T would be $\Omega\left(\frac{1}{\epsilon} \times \frac{1}{\Delta^3_{min}}\right)$.

Next, we are going to use the Adiabatic Theorem in computations, where a natural usage of it would be optimization as follows.

2 Adiabatic Optimization

We are given a function f, as a black box $f : \{0,1\}^n \rightarrow \mathbb{R}$ and the *Goal* is to find

$$x^* \in \{0,1\}^n \text{ such that } f(x^*) = \min(f).$$

In order to solve this problem using the Adiabatic evolution, we assume that the function f has a unique minimum -to avoid degeneracy-. We start with a Hamiltonian for which we can easily construct its ground state, and let the system evolve Adiabaticly to a Hamiltonian whose ground state is $|x^*\rangle$.

Algorithm
Setup: In order to setup we need to define our Hamiltonian as well as its ground state. In order to define our initial Hamiltonian, there are several possibilities, but one is to define it such that it is a small sum of Hamiltonians that act locally, i.e., act on a constant number of qubits – in this case is a single qubit:

$$H(0) = -\sum_{j=1}^{n} I \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes I, \tag{2}$$

where for every j, the middle matrix acts on the j^{th} qubit. This middle matrix has the eigenstate $|+\rangle$ for eigenvalue $+1$ and $|-\rangle$ for the eigenvalue -1. By considering the minus sign before the sum, the eigenstate $|+\rangle^n$ gives us the lowest energy state, namely $-n$, and all other eigenvalues are at least $-n+1$. So, our initial ground state is

$$\Phi(0) = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle .$$

We want the ground state at the end to be the x that minimize the f. We can set

$$H(1) = \sum_{x \in \{0,1\}^n} f(x)\, \Pi_x.$$

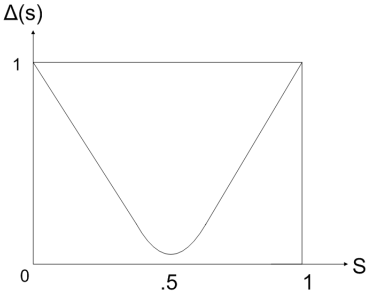
where Π_x is a projection on x.

Process of evolution: After setting our system up, we need to clarify how it evolves by defining an interpolation function between $H(0)$ and $H(1)$:

$$H(s) = (1 - g(s))H(0) + g(s)H(1),$$

where g is a monotone and smooth function with $g(0) = 0, g(1) = 1$.

As an example, consider searching for a unique marked item. We can cast the problem as an Adiabatic optimization problem by chosing f to be the indicator for being non-marked. To determine the time T needed to guarantee success, need to compute $\Delta(s)$.



If we choose a linear function for g, then calculations show that the integral in Equation 1 is $\Theta(N)$, which is no good. We are not going to do calculations to find a better g, but intuitively from Equation 1, $\dot{H}$ should be small whenever Δ is small, and $\dot{H}$ can be larger when Δ is larger. Thus, in the above figure, g can grow quickly at the beginning and the end of the interval. But in the middle, where Δ is close to zero, g can only grow slowly. By adapting g optimally, it turns out we only need time $T = \Theta(\sqrt{N})$.

Thus, by using the Adiabatic optimization we can get the same type of results as Quantum circuit model, namely a quadratic speedup for unstructured search. In general, we do not know whether the Adiabatic optimization is a universal model or not, but the Adiabatic evolution is more general and is a universal model of Quantum computation.
It is easy to see that we can simulate our Adiabatic evolution on our Quantum circuit model; the Hamiltonian evolution process could be divided into small pieces and assumed that in every piece the Hamiltonian is constant. Then $|\Psi((\text{end of piece}))\rangle = e^{-iH\text{length of piece}}|\Psi(\text{beginning of piece})\rangle$ and we can apply $U = e^{-iHt}$ on it, as described in the previous lecture. For that to be possible efficiently, the Hamiltonian should satisfy some conditions like being efficiently sparse. So to simulate the Adiabatic evolution we need to choose the Hamiltonian from one of the good Hamiltonians as introduced in the previous lecture. Another good category are local Hamiltonians, like the one that is used in Equation 2

$$H = H' \otimes I,$$

where H' acts on a constant number of qubits. Then, by using Taylor expansion of the matrix exponential, we can write the unitary operator:

$$U = e^{-iHt} = e^{-iH't} \otimes I.$$

Since $e^{-iH't}$ acts on constant number of qubits using the closure under sum from the previous lecture, we can efficiently construct the unitary operator for small sum of local Hamiltonians. Hence, we can efficiently simulate such an Adiabatic evolution process with our Quantum circuit model. In the other direction, we know how to simulate Quantum circuits with Adiabatic evolution using a small sum of local Hamiltonians but we do not know how to do it by Adiabatic optimization.

3 Universality of Sums of local Hamiltonians

Universality of a model means we can simulate every computation in the Quantum circuit model with a polynomial overhead time. In this section we consider the universality of sums of local Hamiltonians. We see that we can simulate every computation by Quantum circuits of size k, using a sum of a polynomial number of local Hamiltonians, where the time overhead is poly(k). In this section we briefly consider how we can simulate a) a Hamiltonian Quantum Walk and b) a Hamiltonian Adiabatic Evolution processes, which both are based on sum of local Hamiltonians. The general sketches are as follows:
a) Quantum Walk: We only need to setup a Quantum system, starting from a certain state, and let it evolve according to a Hamiltonian which is sum of a small number of local Hamiltonians. At the end, we need to observe the state.
b) Adiabatic Evolution: We start the system in the ground state of a Hamiltonian and then evolve it Adiabatically -using the Adiabatic evolution-. At the end, we can extract the result from the ground state of the system.

3.1 Quantum Walks

We can simulate this process with a sum of time independent local Hamiltonians, in time polynomial in k. This simulation is sometimes called a *Feynman computer*, because he came up with this idea of simulation for simulating classical reversible computation. But *Feynman*'s idea works for simulating any Quantum computations.
Setup: We have $U_k U_{k-1} \dots U_1$ where each U_j is a local unitary, corresponding to a Quantum gate acting on a constant number of qubits. Let us define our Hamiltonian $H = \sum_{j=1}^{k} H_j$ such that we have one local Hamiltonian for each step of computation. The system consists of two components: one for the state of our system and one which is used as a clock, so the Hamiltonian acts on two registers. We define

$$H_j \left|\Psi\right\rangle \left|j-1\right\rangle = U_j \left|\Psi\right\rangle \left|j\right\rangle$$

and

$$H_j \left|\Psi\right\rangle \left|j\right\rangle = U_j^{\dagger} \left|\Psi\right\rangle \left|j-1\right\rangle$$

H_j acts locally on the first register. If the second register is represented in binary then the process is not local. To make it local, we represent our clock in unary to keep $H_j \left|\Psi\right\rangle \left|j-1\right\rangle$ a local process. After defining the Hamiltonian, we start the process in $\left|\Psi\right\rangle \left|0\right\rangle$ and evolve according to the Hamiltonian H as defined above. We can show that state remains in span of

$$\left|\Psi_j\right\rangle \left|j\right\rangle = U_j U_{j-1} \dots U_1 \left|\Psi\right\rangle \left|j\right\rangle, 0 \leq j \leq k.$$

Here, we are interested to know the final state $\left|\Psi_k\right\rangle$.
We can show that

$$|\left\langle \Psi_k | e^{-iH\frac{k}{2}} | \Psi_0 \right\rangle|^2 = \Omega(\frac{1}{k^{\frac{2}{3}}}),$$

where H is our Hamiltonian. This means that if we start the system with the state $|\Psi\rangle = |0\rangle$ and we observe the second register at time $\frac{k}{2}$, then second register equals k with the probability $\Omega(\frac{1}{k^{\frac{2}{3}}})$ and we have the final state $(U_k U_{k-1} \dots U_1 |0\rangle)$ in the first register. In other words, after $\frac{k}{2}$ step we only observe the second register, if it is not equal to k, we restart the process and otherwise we observe the first register, which would be the final state with the probability $\Omega(\frac{1}{k^{\frac{2}{3}}})$. We repeat the process $\Theta(k^{\frac{2}{3}})$ times to have a good probability of success.
This way, we show that we can simulate Quantum circuits using Quantum walks with a Hamiltonian that is a sum of small number of local Hamiltonians.

3.2 Adiabatic Evolution

In this simulation, we start from the ground state of an evolving Hamiltonian and we want that the final state of the Hamiltonian be the state that we are interested in.
The first attempt is to simpily use the same Hamiltonian as above. However, it turns out that each uniform superposition of the form

$$\frac{1}{\sqrt{k+1}} \sum_{j=0}^{k} U_k \dots U_1 |\Psi\rangle |j\rangle .$$

is a ground state for any choice of $|\Psi\rangle$. So we like to enforce additionally that $|\Psi\rangle$ corresponds to the initial state of our Quantum algorithm, typically all qubits should be zero. For that reason we change the Hamiltonian slightly by using an additional penalty to it when $|\Psi\rangle \neq |0\rangle$:

$$H(1) = H + H_{penalty},$$

where

$$H_{penalty} = \sum_{j=1}^{n} \Pi_{x_j=1} \otimes \Pi_0$$

In this manner, we set the Hamiltonian as we like and at the end of the process we observe $U_k U_{k-1} \dots U_1 |\Psi\rangle |0\rangle$ with probability $\frac{1}{k+1}$. And again, we repeat this process $k+1$ times, to get a good probability of succeess.
At the beginning of the process we set up our Hamiltonian to

$$H(0) = -I \otimes \Pi_0 + H_{penalty}$$

with the same penalty to be in state $|0\rangle |0\rangle$ as the unique ground state. We evolve the system by using a linear function $g(s)$.
We need to know how long we should run this Adiabatic process to get the final state. This is governed by the Adiabatic Theorem, for which we need to know a lower bound for the gap function. With this setup one can show $\Delta(s) \geq \Omega(\frac{1}{k^2})$ so $T = O(poly(k))$ suffices. Hence, we show a simulation of Adiabatic Evolution by sum of local Hamiltonians with a polynomial overhead in time.

This finishes the first part of the course, where we considered the standard computational setting in which we want to realize a certain transformation from inputs to outputs. Next lecture we will start talking about processes with more than one party involved: Quantum communication and other interactive processes.