Adiabatic Quantum Computing: An Overview

Charles Epstein
Quantum Complexity Theory 6.845

December 12, 2012

1 Introduction

Quantum computations can be implemented not only by the action of quantum circuits, but by the adiabatic evolution of a system's Hamiltonian. This can be done by initializing the system into the ground state of a simple Hamiltonian, and then adiabatically evolving the Hamiltonian to one whose ground state encodes the solution to the problem. The time complexity of the problem, or more basically, the speed at which the Hamiltonian can be evolved adiabatically, is related to the separation between the energy eigenvalues. In some cases, such as Grover Search, the standard computation complexity can be recovered. Importantly, any quantum circuit can be simulated adiabatically.

2 The Adiabatic Theorem

The adiabatic theorem in quantum mechanics holds that a system with a time-changing Hamiltonian will remain in the same energy level over time as long as the evolution time is slow enough. Typically, we speak of the system remaining in the ground state. Intuitively we expect this to be true, as forcing a system to evolve too quickly can transfer energy to it, causing it to become excited to a higher state. To quantify the speed at which a system can be evolved, we examine the Schrodinger equation:

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

In most cases, $[H(t_0), H(t)] = 0$, and thus we can see that unitary time evolution, $|\psi(t)\rangle = U(t, t_0)|\psi(t_0)\rangle$, is given by

$$U(t, t_0) = e^{-i\int_{t_0}^{t} H(t)\ dt}. \tag{1}$$

Computationally, we are concerned with changing the Hamiltonian over a total time T. We can generalize the Schrodinger equation by introducing a dimensionless parameter $s \in [0,1]$ that maps to the elapsed time $t \in [0,T]$, following [1]. We can thus re-express the Schrodinger equation in terms of s:

$$i\frac{ds}{dt}\frac{d}{ds}|\psi(s)\rangle = H(s)|\psi(s)\rangle$$

and therefore

$$\frac{d}{ds}|\psi(s)\rangle = -i\frac{dt}{ds}H(s)|\psi(s)\rangle.$$

Introducing a "delay factor" $\tau(s) = \frac{dt}{ds}$, we then see that [1]

$$\frac{d}{ds}|\psi(s)\rangle = -i\tau(s)H(s)|\psi(s)\rangle. \tag{2}$$

The delay factor $\tau(s)$ can be thought of as a "speed knob" on the device that changes the Hamiltonian. In order for the evolution of the system to be adiabatic, it must be that

$$\tau(s) \gg \frac{\|\frac{d}{ds}H(s)\|_2}{g(s)^2} \tag{3}$$

where $g(s)$ is the gap between the energy levels - particularly, between the ground state and the first excited state. The numerator is proportional to the maximum gap and is polynomial in the input size, thus it is often ignored [1]. The gap between the levels is most typically the subject of study. In cases where $g(s)$ is not explicitly known, the minimum gap g_{min} is found instead, providing a bound on the necessary evolution time [1]. Generally, it must hold that the evolution time $T \gg 1/g_{min}^2$. Finding g_{min} is of fundamental importance to assessing the capabilities of adiabatic quantum computing.

3 The Adiabatic Algorithm

Adiabatic quantum computing can be used to solve instances of the satisfiability problem. Typically, 3-SAT has been studied. The basic concept of the adiabatic algorithm is that the system is initialized into the ground state of a simple Hamiltonian, which is then adiabatically evolved to a more complex Hamiltonian whose ground state encodes the solution to the problem. Provided the evolution has been adiabatic, the system will be in the solution-encoding ground state. We can define this solution Hamiltonian, H_S, as the sum of local Hamiltonians H_i on each of the C satisfiability clauses [2], such that

$$H_S = \sum_{i=0}^{C} I^{\otimes i} H_i I^{\otimes C-i} \tag{4}$$

Each local Hamiltonian H_i acts only on clause i and encodes the correct assignment of variables by assigning an “energy penalty” to all of the incorrect configurations - these assignments typically have eigenvalue 1. The correct configuration typically has eigenvalue 0, making it the lowest-energy state. Since the energy levels of H_S are determined by the sum of the energies contributed by each clause, it follows that the ground state H_S is the one in which all clauses are satisfied.

Since the solution Hamiltonian is often taken to encode the solution in the computational ($|0\rangle$, $|1\rangle$) basis, the initial Hamiltonian H_I is often taken to have the Hadamard $|+\rangle^{\otimes n}$ as its ground state. This ensures that H_I and H_S do not commute. If $[H_I, H_S] = 0$, then simultaneous eigenstates can be found, meaning that the levels manifestly cross, sending $g_{min} \to 0$ and $T \to \infty$. We introduce the hat notation in which $|+\rangle = H|0\rangle = |\hat{0}\rangle$ [1]. Thus, an example H_I with $|\hat{0}\rangle^{\otimes n}$ as its ground state is [1]:

$$H_I = \sum_{x \in \{0,1\}^n \setminus \{0^n\}} |\hat{x}\rangle\langle\hat{x}| \tag{5}$$

We time-evolve the system from this H_I to H_S as s goes from 0 to 1:

$$H(s) = (1-s)H_I + sH_S. \tag{6}$$

At time $s = 1$, the Hamiltonian is $H(1) = H_S$, and if the evolution time has been slow enough, the system will be in its ground state.

4 Adiabatic Time Complexity

In general, the time complexity for the adiabatic algorithm’s action on a satisfiability instance is not known. It can be shown that the adiabatic model does not provide any extra computing power over “standard” quantum computation [1]. In some cases it fares far worse, such as for the Perturbed Hamming Weight Problem [1], an otherwise easy problem that takes exponentially long for an adiabatic quantum computer to solve. This is due to a strong local energy minimum that differs from the global one. For some examples, notably Grover Search, the standard quantum complexity can be recovered. We can show that the $O(\sqrt{N})$ query complexity of the “conventional” Grover algorithm is the same as the time complexity of the adiabatically-implemented version [1]. We can further show that this is optimal [3].

4.1 Grover Search Complexity

To perform Grover Search adiabatically, we first definite the initial and solution Hamiltonians. We take the H_I to be as defined in Eq. (5), with $|\hat{0}\rangle^{\otimes n}$ as the ground state. The solution Hamiltonian H_S is defined like the Oracle in the standard Grover implementation, such that [1]

$$H_S = \sum_{z \in \{0,1\}^n \setminus \{A\}} |z\rangle\langle z| \tag{7}$$

where A is the target string. Note that this Hamiltonian assigns an energy penalty of 1 to all strings aside from the target, which has eigenvalue 0. The gap between the ground state and first excited state of this Hamiltonian can be found analytically, yielding [1]

$$g(s) = \sqrt{\frac{N + 4(N-1)(s^2 - s)}{N}}. \tag{8}$$

At first glance, it would appear that $g_{min} \propto 1/\sqrt{N}$, indicating an $O(N)$ time complexity. However, we can make use of the fact that the energy levels are far apart for most of the transition, and we can speed up the system in these regions. We revisit $\tau(s) = \frac{dt}{ds}$ of Eq. (2) and can find the total necessary time by integrating:

$$T = \int_{s=0}^{1} \left(\frac{dt}{ds}\right) ds = \int_{0}^{1} \tau(s)\, ds \propto \frac{N \cdot \arctan(\sqrt{N-1})}{\sqrt{N-1}} = O(\sqrt{N}) \tag{9}$$

This recovers the typical $O(\sqrt{N})$ complexity for the Grover Search algorithm.

4.2 Grover Optimality in the Adiabatic Model

In the adiabatic model, Grover Search is still optimal and retains $\Omega(\sqrt{N})$ complexity. This is proven by Farhi and Gutmann in [3]. Following their reasoning, we start by recasting the problem as one in which we are given a Hamiltonian H_z with one eigenvalue E and all others zero, i.e.,

$$H_z = E|z\rangle\langle z| \tag{10}$$

with $|z\rangle$ both normalized and unspecified. We can write the Grover search as one in which we will find the specific choice of $|z\rangle$. Suppose we add a "driving" Hamiltonian $H_D(t)$ to the system, such that [3]

$$H(t) = H_z + H_D(t). \tag{11}$$

The addition of $H_D(t)$ will drive the system to a state in which $|z\rangle$ can be determined [3]. This can be done by starting with two copies of a $|z\rangle$-independent state $|\psi_z, t = 0\rangle$ and $|\psi, t = 0\rangle$, with $|\psi_z, t = 0\rangle = |\psi, t = 0\rangle$. On the former, we act with H_z and $H_D(t)$:

$$i\frac{d}{dt}|\psi_z, t\rangle = [H_z + H_D(t)]|\psi_z, t\rangle \tag{12}$$

and on the latter only $H_D(t)$,

$$i\frac{d}{dt}|\psi, t\rangle = H_D(t)|\psi, t\rangle. \tag{13}$$

We hold that if $|z\rangle$ can be distinguished from some other state $|z'\rangle$, then $|\psi_z, t\rangle$ must be significantly different from $|\psi, t\rangle$. Mathematically, it must be that

$$\| |\psi_z, t\rangle - |\psi, t\rangle \|^2 \geq \varepsilon$$

and thus [3]

$$\sum_z \| |\psi_z, t\rangle - |\psi, t\rangle \|^2 \geq N\varepsilon. \tag{14}$$

It can be shown that [3]

$$\frac{d}{dt} \sum_z \| |\psi_z, t\rangle - |\psi, t\rangle \|^2 \leq 2EN^{1/2}, \tag{15}$$

and therefore since $|\psi_z, t = 0\rangle = |\psi, t = 0\rangle$,

$$\sum_z \| |\psi_z, t\rangle - |\psi, t\rangle \|^2 \leq 2EN^{1/2}t. \tag{16}$$

We can substitute the left-hand side using Eq. (14) and see that in order for the difference between the states to be bounded, it must hold that

$$t \geq \left(\frac{\varepsilon}{2E}\right)\sqrt{N}. \tag{17}$$

This shows that $|z\rangle$ can be determined in $\Omega(\sqrt{N})$ time, and not faster [3].

5 Notes on Universality

Adiabatic quantum computing is polynomially equivalent to the standard quantum-circuit implementation. Foremost, any quantum circuit can be simulated by an adiabatic quantum computer with polynomial overhead. To show this, we will follow the reasoning of Aharonov, et al. [4]. First we will reconsider our definition of H_S. In the standard model of quantum computation, let us apply a quantum circuit of L gates, with the state after gate ℓ being $|\alpha(\ell)\rangle$. In our earlier adiabatic algorithm, the ground state of H_S is $|\alpha(L)\rangle$. However, it is not always possible to construct H_S without explicitly knowing its ground state (earlier we were effectively given oracle access). We consider Kitaev's history state

$$|\eta\rangle = \frac{1}{\sqrt{L+1}} \sum_{\ell=0}^{L} |\alpha(\ell)\rangle \otimes |1^l 0^{L-\ell}\rangle \tag{18}$$

as a possible replacement for $|\alpha(L)\rangle$ [4]. This state consists of each step of the calculation tensored with a "clock state" encoding the step number in unary. The benefit of using $|\eta\rangle$ as the ground state of H_S is that H_S can now be constructed with no knowledge of $|\alpha(L)\rangle$. Under this construction, we would like $|\alpha(0)\rangle \otimes |0\rangle^{\otimes L} = |0\rangle^{\otimes n} \otimes |0\rangle^{\otimes L}$ to be the ground state of H_I. Following [4], we define

$$H_I = H_{\text{clockinit}} + H_{\text{clock}} + H_{\text{init}} \tag{19}$$

$$H_S = \frac{1}{2}\sum_{\ell=1}^{L} H_\ell + H_{\text{clock}} + H_{\text{init}}. \tag{20}$$

We time-evolve the state according to Eq. (6). The terms of the Hamiltonians can be explained as follows. We adopt the notation in which $|x\rangle\langle x|_i$ refers to to an operation on qubit i, with a superscript ^{c} indicating clock cubit indices. First, H_{init} ensures that at the start of the computation, the computation qubits are all zero. It takes the form [4]

$$H_{\text{init}} = \sum_{i=1}^{n} |1\rangle\langle 1|_i \otimes |0\rangle\langle 0|_1^c \tag{21}$$

which enforces an energy penalty on all states in which the computation qubits do not start out as $|0\rangle^{\otimes n}$. H_{clock} ensures that the clock register has a legal value consisting of a string of ones followed by a string of zeros. It takes the form [4]

$$H_{\text{clock}} = \sum_{\ell=1}^{L-1} |01\rangle\langle 01|_{\ell,\ell+1}^c \tag{22}$$

which gives an energy penalty to illegal states containing a zero followed by a one. Continuing, we examine $H_{\text{clockinit}} = |1\rangle\langle 1|_1^c$, which ensures that the initial clock state is $|0\rangle^{\otimes L}$ [4]. Finally, we examine the first term in Eq. (20). Each term H_ℓ in the sum checks the movement of the clock state; if it is unchanged it applies the identity, otherwise it imposes a factor of -1 and applies a unitary time-evolution operator to advance the computation [4]:

$$H_\ell = I\otimes|100\rangle\langle 100|_{\ell-1,\ell,\ell+1}^c - U_\ell^\dagger\otimes|100\rangle\langle 110|_{\ell-1,\ell,\ell+1}^c - U_\ell\otimes|110\rangle\langle 100|_{\ell-1,\ell,\ell+1}^c + I\otimes|110\rangle\langle 110|_{\ell-1,\ell,\ell+1}^c. \tag{23}$$

For the boundary terms where $\ell = 1$ or $\ell = L$, the unnecessary clock qubits are omitted. The ground state of this Hamiltonian is $|\eta\rangle$ with $H_S|\eta\rangle = 0$.

Through this method, any general quantum circuit can be implemented adiabatically using 5-local Hamiltonians acting on two computation and three clock qubits. Aharonov, et al., are able to show that this results in a polynomial overhead of $O(L^5)$ [4]. They are able to reduce this to a 3-local Hamiltonian case with $O(L^{14})$ overhead, and finally a system with 2-local Hamiltonians acting on a two-dimensional grid of six-state particles. The six states are necessary to store both the computational and clock registers.

6 Conclusions

Adiabatic quantum computing is a clever, generalized, universal method of implementing quantum computations. Its simplicity may provide a pathway to experimental implementation. The time complexity of the algorithm is related to the separation between the energy eigenvalues of the time-changing Hamiltonian that evolves the system into a solution-encoding ground state [2]. Some quantum computations can be easily implemented, while in general all quantum circuits can be simulated with (unfortunately large) polynomial overhead [4]. It can be shown that Grover Search retains its $O(\sqrt{N})$ complexity and optimality [1, 3].

References

[1] W. van Dam, M. Mosca, and U. Vazirani. How powerful is adiabatic quantum computation? In *Proceedings, 42nd IEEE Symposium on Foundations of Computer Science*, pages 279–287, 2001, arXiv:quant-ph/0206003v1.

[2] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution. 2000, arXiv:quant-ph/0001106v1.

[3] E. Farhi and S. Gutmann. An analog analogue of a digital quantum computation. *Phys. Rev. A*, 57(4):2403–2406, 1998, arXiv:quant-ph/9612026v1.

[4] D. Aharonov, W. van Dam, J. Kempe, et al. Adiabatic quantum computing is equivalent to standard quantum computation. *SIAM Review*, 50(4):755–787, 2008, arXiv:quant-ph/0405098v2.