CHAPTER 7

Adiabatic Quantum Computing and Quantum Replacements of Optimization Algorithms

1. Adiabatic Quantum Computation is Universal

In this lecture we would like to explain the following beautiful result by Aharonov et al. [2008]:

Theorem 1. The model of adiabatic computation is polynomially equivalent to the standard model of quantum computation.

More specifically, Theorem 1 can be also described as:

Theorem 2. The model of adiabatic computation with explicit sparse Hamiltonians is polynomially equivalent to the standard model of quantum computation.

Even more interestingly, in Aharonov et al. [2008] it is explicitly shown that:

Theorem 3. Any quantum computation can be efficiently simulated by an adiabatic computation with 2-local nearest neighbor Hamiltonians operating on six-state particles set on a two dimensional grid.

We will discuss how one begins to even show these theorems above.

1.1. Motivation. Let us provide some motivation. The study of adiabatic quantum computation (AQC) was initiated several years ago by Farhi, Goldstone, Gutmann and Sipser Farhi et al. [2000], who suggested a novel quantum algorithm for solving classical optimization problems such as Satisfiability (SAT).

Their algorithm, that for what follows will abbreviated as AQC (abusing notation) and will explicitly describe later on, is based on a celebrated theorem in quantum mechanics known as the *adiabatic theorem*.

The exact worst-case behavior of AQC is not known. On one the positive side, several simulations on random instances of up to 20 quantum bits led to various optimistic speculations. Fill in modern details.

On the negative side, there is some evidence that AQC takes exponential time in the worst-case for NP-complete problems.

Nevertheless, adiabatic computation was since shown to be promising in other less ambitious directions: it possesses several interesting algorithmic capabilities, as we will soon review, and in addition, it exhibits inherent robustness against certain types of quantum errors.

1.2. Adiabatic Quantum Computation. Let us re-introduce AQC for another time (sorry for being repetitive): A computation in this model is specified by two Hamiltonians named H_{init} and H_{final}.

The ground state[1] of H_{init} is required to be an easy to prepare state (it can be done efficiently) and serves as the input of the computation.

The output of AQC is the ground state of the final Hamiltonian H_{final}. Hence, we choose an H_{final} whose ground state represents the solution to our problem.

Additionally, we require the Hamiltonians to be **local**[2].

[1] Recall that this refers to the eigenvector with smallest eigenvalue of a Hamiltonian.

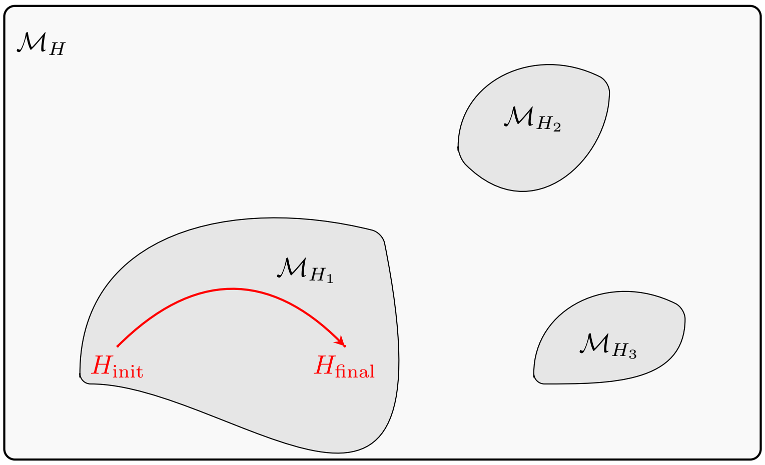
[2] We require them to only involve interactions between a constant number of particles (this can be seen as the equivalent of allowing gates operating on a constant number of qubits in the gate model)

This, in particular, makes sure that the Hamiltonians have a short classical description (simply by listing the matrix entries of each local term).

The running time of the adiabatic computation is determined by the minimal spectral gap[3] of all the path connected Hamiltonians along the curve:

$$\begin{aligned} s : [0,1] &\to \mathcal{M}_H \\ H_{\text{init}} &\mapsto H_{\text{final}} \end{aligned}$$

Below we see a schematic description of the space of Hamiltonians. Note that within each disconnected component we only consider Hamiltonians related by homotopy equivalent paths.



Concretely for any $s \in [0,1]$ we have and infinite family of path parametrized Hamiltonians:

$$H(s) = (1-s)H_{\text{init}} \ + sH_{\text{final}} \tag{7.1}$$

and of course we are interested in reachign $s = 1$ to obtain H_{final}.

If this is done "slowly enough" we say we perform adiabatic computation and it is polynomial time if the corresponding minimal spectral gap is at least inverse polynomial.

Let us provide further motivation from a physics point of view:

- Recall that H corresponds to the energy of the quantum system.
- To be physically realistic and implementable it must be local.
- Ground state of H is the state of lowest energy.
- We can set up a quantum system in the ground state of H_{init} (which is supposed to be easy to generate) and apply the Hamiltonian H_{init} to the system. We then slowly modify the Hamiltonian along the path from H_{init} towards $H_{\text{final.}}$.

From the **adiabatic theorem** it follows that if this transformation is performed slowly enough (determined by the minimal spectral gap), the final state of the system will be in the ground state of H_{final} , as required.

What about **Computational Power** van Dam et al. [2001]? What is the computational power of this model? In order to refer to the adiabatic model as a computational model that computes classical functions (rather than quantum states), we consider the result of the adiabatic computation to be the

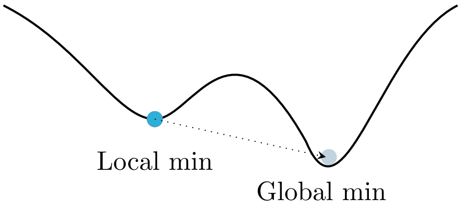
[3] The difference between the ground state eigenenergy and the first excited state eigenenergy.

outcome of a measurement of one or more of the qubits, performed on the final ground state. AQC is performed on qubits similar to the ones of the gate-based computers.

It is known from the very early 2000's that adiabatic computation can be efficiently simulated by standard quantum computers Van Dam et al. [2001], Farhi et al. [2001]. It follows that, its computational power it cannot be greater than that of standard (gate-based) quantum computers.

Several positive results are known, regarding the paower of AQC. For example Grover search can be realized as adiabatic quantum computation Van Dam et al. [2001], Roland and Cerf [2002]

Moreover, AQC can "tunnel" through wide energy barriers and thus outperform simulated annealing Farhi et al. [2002].



However, whether adiabatic computation can achieve the full power of quantum computation was not known. Even whether adiabatic computation can simulate general classical computations efficiently was unknown prior to Aharonov et al. [2008].

The interest for optimization problems: What was known is the potential of AQC on a restricted class of adiabatic algorithms that can be referred to as adiabatic optimization algorithms.

In these algorithms, H_{final} is chosen to be a diagonal matrix, corresponding to a combinatorial optimization problem. Being diagonal implies that the ground state of H_{final} (the output of the computation) is a classical state, (a state in the computational basis). Nevertheless, we want to show something more powerful. We only will assume that the Hamiltonians involved are local.

For that let us discuss n**-qubit systems**: An n-qubit is described by a state in Hilbert space of dimension 2^n, the tensor product of 2-dimensional Hilbert spaces $\mathcal{H} = \mathbb{C}$, that is:

$$|\psi\rangle \in \mathbb{C}^{\otimes n}. \tag{7.2}$$

In terms of the individual qubits:

$$|\psi\rangle = |i_1\rangle \otimes \ldots |i_n\rangle = |i_1 \ldots i_n\rangle, \tag{7.3}$$

where $i_j \in \{0, 1\}$.

Evolution. In the standard model of quantum computation, the state of n qubits evolves in discrete time steps by unitary operations. Of course, the underlying physical description of this evolution is continuous, and is governed by Schrödinger's equation:

$$\imath \frac{\mathrm{d}}{\mathrm{d}t} |\psi(t)\rangle = H|\psi(t)\rangle \tag{7.4}$$

where H is the system's Hamiltonian and $|\psi(t)\rangle$ is the state of the n qubits at time t.

We have already seen that the solution of Schrödinger equation is given as:

$$|\psi(t)\rangle = \exp(-\imath H t)|\psi(0)\rangle. \tag{7.5}$$

Given that the state of the system at time $t = 0$ is equal to $|\psi(0)\rangle$, one can in principle solve Schrödinger's equation with this initial condition, to get $|\psi(T)\rangle$, the state of the system at a later (terminal) time $t = T$.

Spectral Gap. We define $\Delta(H)$, the **spectral gap** of a Hamiltonian H, to be the difference between the lowest eigenvalue of H and its second lowest eigenvalue. Note that $\Delta(H) = 0$ if the lowest eigenvalue is degenerate.

What is **Locality**? One cannot efficiently apply any arbitrary Hamiltonian on a n-qubit system (just describing it requires roughly 2^{2n} space). For this we restrict to k-local Hamiltonians. We say that a Hamiltonian H is k-local if H can be written as $\sum_A H^A$ where A runs over all subsets of k particles.

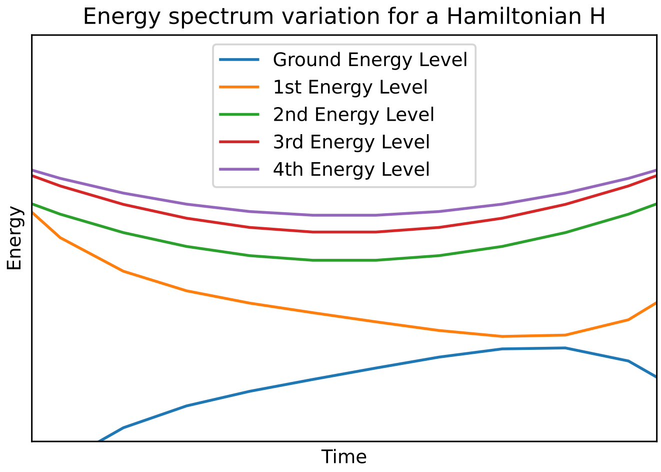
Notice that for any constant k, a k-local Hamiltonian on n-qubits can be described by $2^{2k}n^k = \text{poly}(n)$ numbers. We say that H is local if H is k-local for some constant k. (Commonly $k = 2$ in NISQ devices.)

1.3. Adiabatic Theorem. The cornerstone of the adiabatic model of computation is the celebrated adiabatic theorem Kato [1950]. Consider a time-dependent Hamiltonian $H(s)$, and a system initialized at time $t = 0$ in the ground state of $H(0)$ (here and in the following we assume that for all $H(s)$ has a unique ground state for all s).

We let the system evolve according to the Hamiltonian $H(s)$, for $s := t/T$, from time $t = 0$ to the terminal time $t = T$. As said before, The adiabatic theorem affirms that for large enough T the final state of the system is very close to the ground state of $H(1)$.

How large T should be for this to happen is determined by the spectral gap of the Hamiltonians $\Delta(H(s))$ Aharonov et al. [2008].

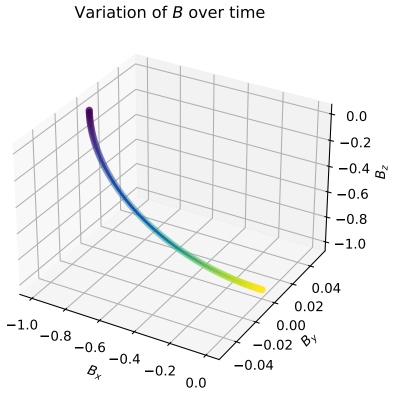
In the figure below we see the change of the few lowest eigenenergies for a certain evolution. It is crucial that the spectral gap does not change sign: the lowest blue eigenenergy nowhere crosses with the orange one.

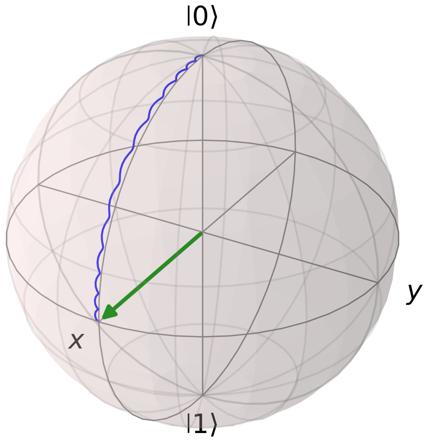


Physical intuition: Consider a spin particle (e.g. an electron) in a magnetic field B which rotates from the x direction to the z direction in a total time T. The dynamics of the particle are described by the Hamiltonian:

$$H(t) = -\cos\left(\frac{\pi t}{2T}\right)\sigma_x - \sin\left(\frac{\pi t}{2T}\right)\sigma_z. \tag{7.6}$$

Suppose that initially, the particle points in the x direction: $|\psi(0)\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, the ground state of $H(0)$. As the magnetic field is slowly rotated toward the z direction, see Fig. below, the particle's spin begins to precess about the new direction of the field, moving it toward the z axis.





Note that this produces a small wiggling component out of the xz-plane. **Adiabaticity** is seen by allowing T to be made larger and larger, so that the rotation of the field direction happens more and more slowly (as compared to the speed of precession). Then, the state will precess in a tighter and tighter orbit about the field direction (aligning completely with the geodesic). In the limit of arbitrarily slow rotation of the field, the state simply tracks the field, remaining in the instantaneous ground state of $H(t)$.

Full statement (at last): More generally, let $H(s)$ be a Hermitian operator that varies smoothly as a function of $s := t/T$. Then for T arbitrarily large, $H(t)$ varies arbitrarily slowly as a function of t. An initial quantum state $|\psi(0)\rangle$ evolves according to the Schrödinger equation (7.4), or, equivalently:

$$\imath \frac{\mathrm{d}}{\mathrm{d}s}|\psi(s)\rangle = TH|\psi(s)\rangle. \tag{7.7}$$

Now suppose that $|\psi(0)\rangle$ is an eigenstate of $H(0)$, which we assume for simplicity is the ground state, and is nondegenerate. Furthermore, suppose that the ground state of $H(s)$ is nondegenerate for all s.

Theorem 4 (Adiabatic Theorem). Given the above, in the limit $T \to \infty$, $|\psi(T)\rangle$ will be the ground state of $H(1)$.

Remark: The proof of the adiabatic theorem is a very interesting exercise.

2. Sketch Proofs for the Universality of AQC

Let us repeat ousrselves (again) but a bit more formally:

Theorem 5 (Adiabatic Theorem (Proper)). Let H_{init} and H_{final} be two Hamiltonians acting on a quantum system and consider the time-dependent Hamiltonian $H(s) := (1-\mathrm{s})\ H_{\text{init}} + sH_{\text{final.}}$. Assume that for all $s, H(s)$ has a unique ground state. Then for any fixed $\delta > 0$, if

$$T \geq \Omega\left(\frac{\|H_{\text{final}} - H_{\text{init}}\|^{1+\delta}}{\epsilon^{\delta} \min_{s\in[0,1]}\{\Delta^{2+\delta}(H(s))\}}\right) \tag{7.8}$$

then the final state of an adiabatic evolution according to H for time T (with an appropriate setting of global phase) is ϵ-close in ℓ_2-norm to the ground state of H_{final} .

The matrix norm is the spectral norm $\|H\| := \max_w \|Hw\|/\|w\|$.

The AQC model (again): Let us now describe the model of adiabatic computation. The adiabatic circuit is determined by H_{init} and H_{final} and the output of the computation is (close to) the ground state of H_{final} .

Definition 1. A k-local AQC $(n, d, H_{\text{init}}, H_{\text{final}}, \epsilon)$ is specified by two k-local Hamiltonians, H_{init} and H_{final} acting on n d-dimensional particles, such that both Hamiltonians have unique ground states. The ground state of H_{init} is a tensor product state. The output is a state that is ϵ-close in ℓ_2-norm to the ground state of H_{final}. Let T be the smallest time such that the final state of an adiabatic evolution according to $H(s) := (1-s)H_{\text{init}} + sH_{\text{final}}$ for time T is ϵ-close in ℓ_2-norm to the ground state of H_{final}. The running time of the adiabatic algorithm is defined to be $T \cdot \max_s \|H(s)\|$.

2.1. Proof Sketch of the Equivalence 1.

Gates to AQC.

Theorem 1 can be proved by simulating a quantum circuit with L (two-qubit) gates on n qubits by an adiabatic computation on $n + L$ qubits.

Note that the opposite direction can also be shown Farhi et al. [2000].

We will show this by considering 5-qubit interactions. However, it is possible to reduce it to three. (Note that the practical implementation of 5-qubit interactions is still not easy.)

A Theorem:

Theorem 6. Given a quantum circuit on n qubits with L two-qubit gates implementing a unitary U and $\epsilon > 0$, there exists a 5-local adiabatic computation $(n+2, 2, H_{\text{init}}, H_{\text{final}}, \epsilon)$ whose running time is poly$(L, 1/\epsilon)$ and whose output is ϵ-close to $U|0\rangle^n = U|0\rangle^{\otimes n}$. Additionally, H_{init} and H_{final} can be computed by a polynomial time Turing machine.

The Hamiltonian: The Hamiltonian we need is defined in Kitaev et al. [2002]. We begin by defining a state

$$|\gamma_\ell\rangle := |\alpha(\ell)\rangle \otimes |1^\ell 0^{L-\ell}\rangle^c. \tag{7.9}$$

Here $|\alpha(\ell)\rangle$ denotes the state of the circuit after the application of the ℓ-th gate (and the superscript c denotes the clock qubits required for the proof of the theorem). The notation $|1^\ell 0^{L-\ell}\rangle$ means that there are ℓ qubits in the state $|1\rangle$ followed by $(L-\ell)$ qubits in the state $|0\rangle$.

We now define the Hamiltonian H_{init} with ground state $|\gamma_0\rangle = |0^n\rangle \otimes |0^L\rangle^c$,

and the local Hamiltonian H_{final} with ground state $|\eta\rangle = \frac{1}{\sqrt{L+1}} \sum_{\ell=0}^{L} |\gamma_\ell\rangle$.

We know what our initial and final eigenvectors need be. It turns out that the way to do it is:

$$\begin{aligned} H_{\text{init}} &:= H_{\text{clock init}} + H_{\text{input}} + H_{\text{clock}} \\ H_{\text{final}} &:= \frac{1}{2}\sum_{\ell=1}^{L} H_\ell + H_{\text{input}} + H_{\text{clock}} \end{aligned} \tag{7.10}$$

Note that the terms in the two Hamiltonians are defined such that the only state whose energy is 0 is the desired ground state.

This is done by assigning an energy penalty to any state that does not satisfy the required properties of the ground state. The different terms, which correspond to different properties of the ground states, are described in the following paragraphs.

Adiabatic Evolution: The adiabatic evolution then follows the time-dependent Hamiltonian

$$H(s) = (1-s)H_{\text{init}} + sH_{\text{final}} \tag{7.11}$$

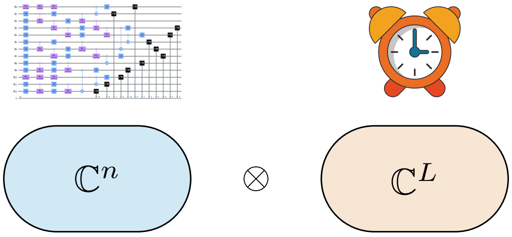
Notice that as s goes from 0 to 1, $H_{\text{clock init}}$ is slowly replaced by $\frac{1}{2}\sum_{\ell=1}^{L} H_\ell$ while H_{input} and H_{clock} are held constant.

The Hamiltonians Explained: H_{clock} First, H_{clock} checks that the clock's state is of the form $|1^\ell 0^{L-\ell}\rangle^c$ for some $0 \leq \ell \leq L$ (thus "clock").

To do this we give a penalty to any state (of the clock register) that contain a sequence 01, that is:

$$H_{\text{clock}} \; := \sum_{\ell=1}^{L-1} |01\rangle\langle 01|^{c}_{\ell,\ell+1}. \tag{7.12}$$

Below, we present a schematic of the Hilbert space of our system. Note that there is a typo in the figure, it should read $\mathbb{C}^{2n}$ and $\mathbb{C}^{2L}$.



The subscript indicates which clock qubits the projection operates on. The term $|01\rangle\langle 01|^{c}_{\ell,\ell+1}$ has an energy penalty (eigenvalue 1) when the clock qubits at positions ℓ and $\ell+1$ are in the state $|01\rangle$, and no energy penalty (eigenvalue 0) otherwise. By summing these terms over all pairs of adjacent clock qubits (from $\ell = 1$ to $L-1$), the clock Hamiltonian enforces a penalty whenever there is an out-of-order pair of clock qubits.

This means that the lowest energy states of the clock Hamiltonian correspond to the correct progression of the computation, where the clock qubits represent a valid encoding of the computation stages. In this way, H_{clock} helps control the adiabatic evolution of the quantum system and ensures that it follows the desired gate sequence.

The Hamiltonians Explained: H_{init} H_{input} checks that if the clock is at $|0\rangle^{\otimes L}$ (we ommited the c-clock index here, clearly referring to $\mathcal{H}_{\text{clock}}$) then the computation qubits must be in the state $|0\rangle^{\otimes n}$. This is given by:

$$H_{\text{init}} := \sum_{i=1}^{n} |1\rangle\langle 1| \otimes |0\rangle\langle 0|. \tag{7.13}$$

Let us discuss $H_{\text{clock init}}$. The goal of $H_{\text{clock init}}$ is to check that the clock's state is $|0\rangle^{\otimes L}$:

$$H_{\text{clock init}} := |1\rangle\langle 1|. \tag{7.14}$$

Finally, we have the term

$$\frac{1}{2}\sum_{\ell=1}^{L} H_{\ell} \tag{7.15}$$

which is the term representing the gate-based Hamiltonian and it is only apparent in the end of the AQC (in principle it is unknown).

Summary: $H_{\text{clock init}}$ and H_{clock}: These terms are related to the clock qubits. $H_{\text{clock init}}$ sets the initial state of the clock qubits and ensures that the computation starts with all clock qubits in the state $|1\rangle^{c}$. H_{clock} penalizes out-of-order transitions and enforces a step-by-step progression through the circuit.

H_{input}: This term sets the initial state of the quantum circuit. It essentially encodes the input data of the problem you want to solve.

$\frac{1}{2}\sum_{\ell=1}^{L} H_{\ell}$: This term is present only in the final Hamiltonian, H_{final}. It represents the quantum gates in the circuit. The factor $\frac{1}{2}$ ensures that the spectrum of the Hamiltonian is non-negative, which is a requirement for the adiabatic theorem to hold.

The final Hamiltonian: We now proceed to the first term in H_{final} . The Hamiltonian H_{ℓ} checks that the propagation from step $\ell - 1$ to ℓ is correct. It checks that it corresponds to the application of the gate U_{ℓ}.

For $1 < \ell < L$, it is defined as:

$$\begin{aligned} H_\ell \coloneqq & \mathbf{1} \otimes |100\rangle\langle 100|^c_{\ell-1,\ell,\ell+1} - U_\ell \otimes |110\rangle\langle 100|^c_{\ell-1,\ell,\ell+1} \\ & - U_\ell^\dagger |100\rangle\langle 110|^c_{\ell-1,\ell,\ell+1} + \mathbf{1} \otimes |110\rangle\langle 110|^c_{\ell-1,\ell,\ell+1}. \end{aligned} \tag{7.16}$$

Intuitively, the three-qubit terms above move the state of the clock one step forward, one step backward, or leave it unchanged (this reminds us a quantum walk). The accompanying matrices $U_\ell, U_\ell^\dagger$ describe the associated time evolution. We have two boundary cases $\ell = 1, L$ (initial and terminal time) for which we omit one clock qubit from these terms and define

$$H_1 \coloneqq \mathbf{1} \otimes |00\rangle\langle 00|_{1,2} - U_1 \otimes |10\rangle\langle 00|_{1,2} - U_1^\dagger \otimes |00\rangle\langle 10|_{1,2} + \mathbf{1} \otimes |10\rangle\langle 10|_{1,2} \tag{7.17}$$

$$H_L \coloneqq = \mathbf{1} \otimes |10\rangle\langle 10|_{L-1,L} - U_L \otimes |11\rangle\langle 10|_{L-1,L} - U_L^\dagger \otimes |10\rangle\langle 11|_{L-1,L} + \mathbf{1} \otimes |11\rangle\langle 11|. \tag{7.18}$$

Spectral gap inverse in L.

We have now seen what are the Hamiltonians needed to transform a gate-based problem to an AQC.

We need to understand the spectral gap now.

Recall the state given by Eq. (7.9):

$$|\gamma_\ell\rangle \coloneqq |\alpha(\ell)\rangle \otimes |1^\ell 0^{L-\ell}\rangle^c.$$

Spectral gap inverse in L: $s > 1/3$.

Let S_0 a subspace of $\mathbb{C}^n \otimes \mathbb{C}^L$ spanned by

$$\{|\gamma_0\rangle, \dots, |\gamma_1\rangle\} \tag{7.19}$$

which are equivariant states (w.r.t. the action of Hamiltonians on S). In other words, we have some form of symmetry.

Theorem 7. The spectral gap of the restriction of $H(s)$ to S_0 satisfies:

$$\Delta(H_{S_0}(s)) = \Omega(L^{-2}), \tag{7.20}$$

for all $s \in [0, 1]$.

Interestingly, the proof uses a continuous-time quantum walk. The proof is technical (not very hard) but we omit it here. The important thing is to understand the need for the Hamiltonians H_{init} and H_{final} in Eq. (7.10).

With the proof on the (inverse in L) polynomial runtime, we claim the following.

The Equivalence Statement: Given a quantum circuit on n qubits with L gates, the quantum adiabatic algorithm with H_{init} and H_{final} as defined in the previous slides, with $T = \mathcal{O}(\epsilon^{-\delta} L^{4+2\delta})$, for fixed $\delta > 0$, outputs a final state $|\eta\rangle$ that is within ℓ_2 distance ϵ of the history state of the circuit. The running time of the AQC algorithm is $\mathcal{O}(TL)$.

Recall, that already from 2000 it was known that gate-based algorithms can be encoded as AQC. With the proof of Theorem 1, the universality of AQC is also proven. A detailed introduction with many complexity theoretical aspects is Albash and Lidar [2018].

3. Practicalities of Adiabatic Quantum Computing

We have discussed that the solution of computational problem can be encoded into the ground state of a time-dependent quantum Hamiltonian $H(s)$ which evolves following the paradigm of AQC.

Quantum annealing (QA) is a framework that incorporates algorithms Kadowaki and Nishimori [1998], Santoro et al. [2002], Das and Chakrabarti [2008] and hardware designed to solve computational problems by quantum evolution towards the ground states of final Hamiltonians that encode classical (optimization) problems.

Note that Quantum Annealers are real:

This is the D-Wave 2000Q system. This is a system that performs quantum annealing using superconducting qubits. The qubits live in the very end of a dilution refridgerator cooled at approximately -273.5 degrees Celcius.

3.1. Stoquasticity.

QA therefore, moves between the idealized assumptions of universal AQC and the unavoidable experimental compromises.

Perhaps the most significant of these compromises has been the design of stoquastic quantum annealers.

Definition 2 (Stoquastic Hamiltonian). A Hamiltonian H is called stoquastic, with respect to a basis B, if and only if H has real nonpositive off-diagonal matrix elements in the basis B.

For example, a Hamiltonian is stoquastic if and only

$$\langle i|H|j\rangle \leq 0, \quad \forall i,j \in \{0,1\}^n, \quad i \neq j. \tag{7.21}$$

This means the ground state of H can be expressed as a classical probability distribution.

AQC with Stoquastic Hamiltonians:

Definition 3. Stoquastic adiabatic quantum computation (StoqAQC) is the special case of AQC restricted to k-local (k fixed) stoquastic Hamiltonians.

Essentially, Quantum Annealing (QA) refers to StoqAQC when considered in (realistic) open quantum systems. For what follows, these two terms are identical.

No Universality: The computational power of stoquastic Hamiltonians has been carefully studied, and is suspected to be limited in the ground-state AQC setting Albash and Lidar [2018].

In other words, **it is quite unlikely that ground-state StoqAQC is universal** Farhi and Harrow [2016].

3.2. Quantum Annealing is very similar to AQC. QA follows the same idea of AQC. We still have the same tools:

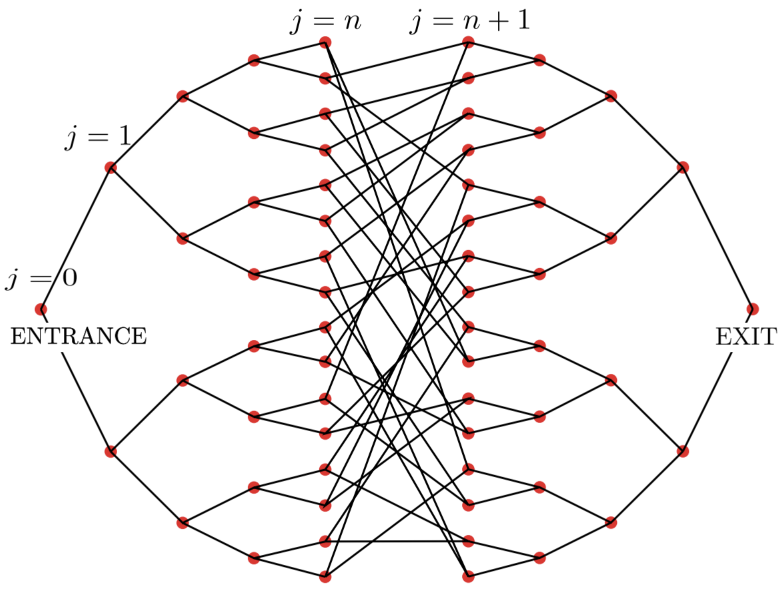
- An initial, easy-to-prepare state and a Hamiltonian H_{init},
- A problem of interest whose solution is encoded into the ground state of a Hamiltonian H_{final},
- Adiabatic evolution using Eq. (7.1):

$$H(s) = (1-s)H_{\text{init}} \; + sH_{\text{final}} \tag{7.22}$$

Exponential Speedups with QA It turns out that QA can be used to obtain exponential speedups!

Somma, Nagaj, and Kieferovaá Somma et al. [2012] showed that, similarly to the case of quantum walks, utilizing QA on the glued-trees problem one obtains an exponential speedup.

Glued-Trees Problem



In this problem we are given an oracle O_A that concists of the adjacency matrix A of two binary trees that are randomly glued. There are $\mathcal{O}(2^n)$ vertices named with randomly chosen $2n$-strings.

The oracle O_A outputs the names of the adjacent vertices on any given input vertex name.

There are two special vertices:

- ENTRANCE
- EXIT

which are the roots of the binary trees. They can be identified because they are the only vertices of degree two in the graph.

Glued-Trees Problem: Given an oracle O_A for the graph and the name x of the ENTRANCE, find the name y of the EXIT.

An efficient method based on quantum walks can solve this problem with constant probability, while no classical algorithm that uses less than a subexponential (in n) number of oracles exists.

3.3. Optimization. An **optimization problem** is a problem to minimize or maximize a real single-valued function of multivariables called the cost function.

If the problem is to maximize the cost function f, it suffices to minimize $-f$.

Additional constraints can be imposed on the objective function:

$$\begin{aligned} \min_{x,y} \quad & f(x,y) && (7.23) \\ \text{s.t.} \quad & g(x) \geq 0 && (7.24) \\ & x \in \mathbb{R}^m,\ y \in \mathbb{Z}^n && (7.25) \end{aligned}$$

(Just a reminder) Optimization problems are classified roughly into two types, easy and hard ones. Loosely speaking, easy problems are those for which we have algorithms to solve in steps(=time) polynomial in the system size (polynomial complexity). In contrast, for hard problems, all known algorithms take exponentially many steps to reach the exact solution (exponential complexity). A potential solution is offered by Quantum Annealing.

Suppose we can solve such problems with QA. Does it converge?

Consider the k-th eigenstate state of the Hamiltonian:

$$H(s)|k\rangle = \lambda_k(s)|k\rangle \tag{7.26}$$

with $|0(0)\rangle$ being the ground state of H_{init} and generically $|0(s)\rangle$ the ground state of $H(s)$.

If $|0(s)\rangle$ is non-degenerate and if initial ground state is $|0(0)\rangle$ then the final state vector, at large T, take the form:

$$|\psi(s)\rangle = \sum_\kappa c_\kappa(s) e^{-\imath T \phi_\kappa(s)} |\kappa(s)\rangle \tag{7.27}$$

with

$$\phi_\kappa(s) = \int_0^s \lambda_\kappa(s')\mathrm{d}s'.$$

Maybe some homework on the topic? It turns out:

$$c_0(s) \approx 1 + \mathcal{O}(T^{-2}), \tag{7.28}$$

$$c_{\kappa\neq 0}(s) \approx \frac{\mathrm{i}}{T}\Big[A_\kappa(0) - \mathrm{e}^{\imath T[\phi_\kappa(s)-\phi_0(s)]} A_\kappa(s)\Big] + \mathcal{O}(T^{-2}) \tag{7.29}$$

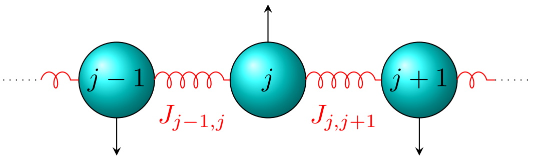
The adiabaticity condition becomes:

$$\frac{1}{\Delta_\kappa(t)^2}\left|\left\langle \kappa(t) \middle| \frac{\mathrm{d}H(t)}{\mathrm{d}t} \middle| 0(t) \right\rangle\right| = \delta \ll 1. \tag{7.30}$$

Convergence via Ising model: Suppose that the optimization (7.23) problem we wish to solve can be represented as the ground-state search of an Ising model of general form

$$H_{\text{Ising}} \equiv -\sum_{i=1}^N J_i \sigma_i^z - \sum_{i,j=1}^N J_{ij}\sigma_i^z\sigma_j^z + \mathcal{O}(\sigma^3). \tag{7.31}$$

Here, $\sigma_i^\alpha(\alpha = x,y,z)$ are the Pauli matrices.



Recall, the eigenvalue of σ_i^z is $+1$ or -1, which corresponds the classical Ising spin.

Most combinatorial optimization problems can be written in this form by, for example, mapping binary variables $\{0,1\}$ to spin variables $\{\pm 1\}$. Another important assumption is that the Hamiltonian (7.31) is proportional to the number of spins N for large N.

Transverse Field To realize QA, a (kinetic) energy term is introduced typically by the so-called time-dependent transverse field:

$$H_{\mathrm{TF}}(t) \equiv -\Gamma(t) \sum_{i=1}^{N} \sigma_i^x \tag{7.32}$$

which results in a variety of possible quantum mechanical effects to the chain:spin flips, quantum fluctuations or quantum tunneling, between the two states $\sigma_i^z = 1$ and $\sigma_i^z = -1$.

Essentially this allows a quantum search of the phase space of the system.

Initially the strength of the transverse field $\Gamma(t)$ is chosen to be very large, and the total Hamiltonian

$$H(t) = H_{\text{Ising}} \; + H_{\mathrm{TF}}(t) \tag{7.33}$$

is dominated by the second kinetic term. (If you know about Simulated Annealing (SA), this is the quantum analogue of the high-temperature limit.)

The evolution of the TF Ising Model: The coefficient $\Gamma(t)$ is then gradually and monotonically decreased toward 0, leaving eventually only the potential term H_{Ising} .

Accordingly the state vector $|\psi(t)\rangle$, which follows the real-time Schrödinger equation, is expected to evolve from the trivial initial ground state of the transverse-field (7.33) to the non-trivial ground state of (7.31), which is the solution of the optimization problem.

An important issue is *how slowly we should decrease* $\Gamma(t)$ *to keep the state vector arbitrarily close to the instantaneous ground state of the total Hamiltonian.*

The following Theorem provides a solution to this problem as a sufficient condition.

Theorem 8. The adiabaticity (7.30) for the transverse-field Ising model (7.31) yields the time dependence of $\Gamma(t)$ as

$$\Gamma(t) = a(\delta t + c)^{-1/(2N-1)} \tag{7.34}$$

for $t > t_0$ (for given $t_0 > 0$) as a sufficient condition of convergence of QA. Here a, c are small constants $\mathcal{O}(1)$ and δ is a small parameter that controls adiabaticity.

Point is: **The power decay above satisfies the adiabaticity condition** (7.30) **which guarantees convergence to the ground state of** H_{Ising} **as** $t \to \infty$.

QA in Practice: Optimization.

In practical situations QA is used as heuristic optimization method.

Due to hardware constructions, at the moment only Quadratic Binary Optimization (QUBO) problems can be implemented.

A QUBO problem reads

$$\min_{x \in \{0,1\}^N} Q(x) \tag{7.35}$$

where the objective function Q is defined as:

$$Q(x) \coloneqq \sum_{i,j=1}^{N} Q_{ij} x_i x_j + \sum_{i=1}^{N} c_i x_i . \tag{7.36}$$

The problem to be optimized is then fully specified by Q_{ij} and c_i.

A broad class of paradigmatic optimization problems from Vertex Cover to the Traveling Salesperson problem have been mapped to QUBO form.

What happens if $k \geq 3$? (That is, if we have problem with interactions of degree 3 or higher.) If the problem of interest has a cost function of high-order interactions, than the quadratic, one has to encode this information in ancilla qubits.

For example, assume a problem encoding involves the 3-local expression

$$xyz, \quad x, y, z \in \mathbb{R}.$$

This has to be mapped to the expression

$$xw,$$

where $w \coloneqq yz$ and impose the additional constraint

$$3w + yz - 2yw - 2zw.$$

The solution is (zero penalization) $w = yz$.

Example: The Knapsack Problem:

We are given a set of weights $w \in \mathbb{Z}^n_{\geq 0}$ and their corresponding values $v \in \mathbb{Z}^n_{\geq 0}$, and the objective is to maximize the total value of the items that can be packed into a knapsack subject to a given weight limit W.

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} v_i x_i, \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i x_i \leq W, \end{aligned} \tag{7.37}$$

where W is the maximum weight limit (threshold) of the knapsack and x_i is the binary variable representing whether the i-th item is to be placed in the knapsack.

MILP to QUBO: In converting MILPs to QUBOs we introduce a slack variable S for each linear inequality and transform it into an equivalent linear equality. We add to the objective a penalty term:

$$\lambda_0 \left(\sum_{i=1}^{n} w_i x_i - W + S \right)^2 \tag{7.38}$$

where the purpose of the auxiliary slack variable S is to reduce this term to 0 once the constraint has been satisfied, $0 \leq S \leq \max_x \sum_i^n w_i x_i - W$.

Note that in practice, S is decomposed into binary representation using variables $s_k \in \{0, 1\}$ as follows:

$$S = \sum_{k=1}^{N_s} 2^{k-1} s_k. \tag{7.39}$$

The parameter N_s corresponds to the number of binary variables required to represent the maximum value that can be assigned to the slack variable, and in the case of Knapsack, $N_s = \lceil \log_2(W) \rceil$, where $\lceil x \rceil$ is the ceiling function. This is often called "log-encoding".

The QUBO formulation: the Knapsack problem can be formulated then as:

$$\max \quad \sum_{i}^{n} v_i x_i - \lambda_0 \left(\sum_{i}^{n} w_i x_i - W + \sum_{k=1}^{N} 2^{k-1} s_k \right)^2, \tag{7.40}$$

Maping to the Ising model:

$$\min \quad - \left(\sum_{i=1}^{n} \sum_{j=1}^{n} J_{ij} s_i s_j + \sum_{i=1}^{n} h_i s_i + c \right) \tag{7.41}$$

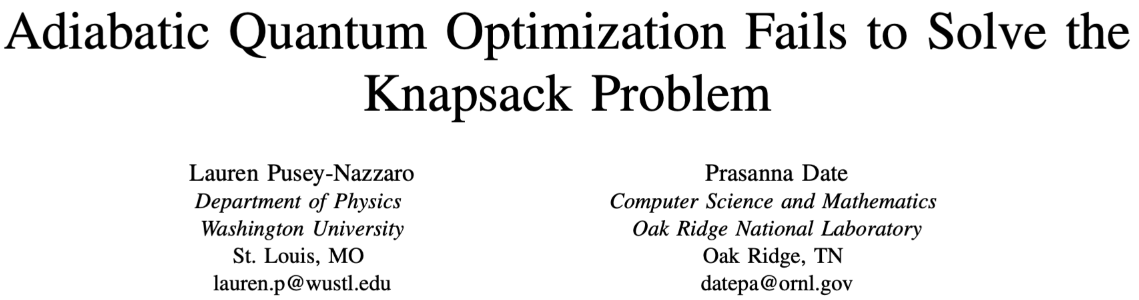
where

$$J_{ij} = \lambda_0 2^{k-1} w_i \delta_{ij}, \tag{7.42}$$

$$h_i = \frac{v_i}{2} - \lambda_0 w_i W, \tag{7.43}$$

$$c = \sum_{i=1}^{n} \frac{v_i}{2} + \lambda_0 \left(\sum_{i=1}^{n} \frac{w_i^2}{4} + \sum_{k=1}^{N} 2^{2k-2} \right). \tag{7.44}$$

Does QA Fail?

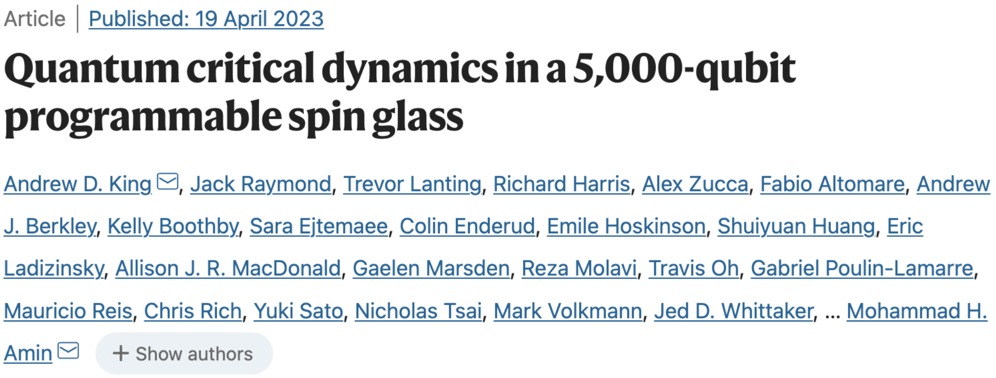


Adiabatic Quantum Optimization Fails to Solve the Knapsack Problem

Lauren Pusey-Nazzaro
Department of Physics
Washington University
St. Louis, MO
lauren.p@wustl.edu

Prasanna Date
Computer Science and Mathematics
Oak Ridge National Laboratory
Oak Ridge, TN
datepa@ornl.gov

Can QA Succeed?



Article | Published: 19 April 2023

Quantum critical dynamics in a 5,000-qubit programmable spin glass

Andrew D. King, Jack Raymond, Trevor Lanting, Richard Harris, Alex Zucca, Fabio Altomare, Andrew J. Berkley, Kelly Boothby, Sara Ejtemaee, Colin Enderud, Emile Hoskinson, Shuiyuan Huang, Eric Ladizinsky, Allison J. R. MacDonald, Gaelen Marsden, Reza Molavi, Travis Oh, Gabriel Poulin-Lamarre, Mauricio Reis, Chris Rich, Yuki Sato, Nicholas Tsai, Mark Volkmann, Jed D. Whittaker, ... Mohammad H. Amin

4. Variational Quantum Algorithms

Two fundamental references on VQAs are the original paper on the Variational Quantum Eigensolver Peruzzo et al. [2014] and the subsequent paper on the Quantum Approximate Optimization Algorithm Farhi et al. [2014]. A beautiful survey is given by Cerezo et al. [2021].

Parametrized Quantum Circuits (PQCs): Variational Quantum Algorithms (VQAs) provide a general framework that can be used to solve a variety of problems.

For that we first need the idea of a parametrized quantum circuit.

Definition 4. A parametrized quantum circuit (PQC) is a continuous function $U : \mathbb{R}^L \to U(N)$ mapping any real parameter vector $\vartheta \in \mathbb{R}^L$ to a unitary $U(\vartheta)$.

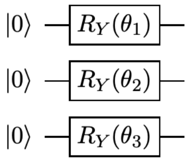
In practice such a quantum circuit is a sequence of universal quantum gates' compositions and/or tensor products.

Consider, for a moment, the following optimization problem (and keep it in mind):

$$\min_{x \in \{0,1\}^n} \quad f(x). \tag{7.45}$$

A VQA is, essentially, a (quantum) continuous relaxation of this problem.

Below we see a PQC example:

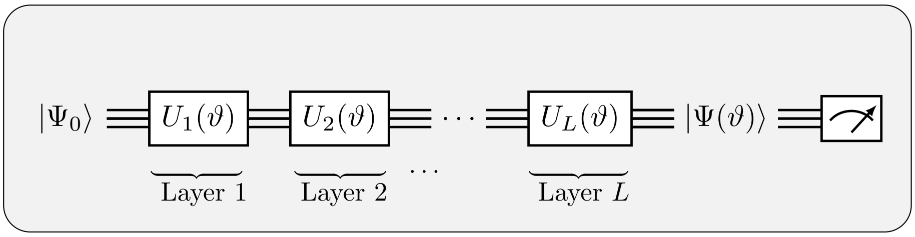


Here $U(\theta)$ is given as:

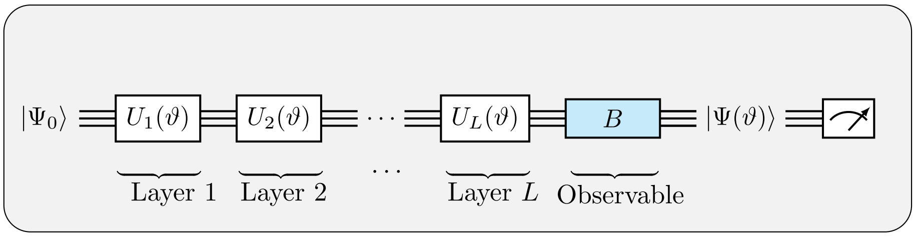
$$\begin{aligned}
U(\theta) &= R_Y\left(\theta_1\right) \otimes R_Y\left(\theta_2\right) \otimes R_Y\left(\theta_3\right) \\
&= \begin{pmatrix} \cos\frac{\theta_1}{2} & -\sin\frac{\theta_1}{2} \\ \sin\frac{\theta_1}{2} & \cos\frac{\theta_1}{2} \end{pmatrix} \otimes \begin{pmatrix} \cos\frac{\theta_2}{2} & -\sin\frac{\theta_2}{2} \\ \sin\frac{\theta_2}{2} & \cos\frac{\theta_2}{2} \end{pmatrix} \otimes \begin{pmatrix} \cos\frac{\theta_3}{2} & -\sin\frac{\theta_3}{2} \\ \sin\frac{\theta_3}{2} & \cos\frac{\theta_3}{2} \end{pmatrix} \\
&= \begin{pmatrix}
c_1c_2c_3 & -c_1c_2s_3 & -c_1s_2c_3 & c_1s_2s_3 & -s_1c_2c_3 & s_1c_2s_3 & s_1s_2c_3 & -s_1s_2s_3 \\
c_1c_2s_3 & c_1c_2c_3 & -c_1s_2s_3 & -c_1s_2c_3 & -s_1c_2s_3 & -s_1c_2c_3 & s_1s_2s_3 & s_1s_2c_3 \\
c_1s_2c_3 & -c_1s_2s_3 & c_1c_2c_3 & -c_1c_2s_3 & -s_1s_2c_3 & s_1s_2s_3 & -s_1c_2c_3 & s_1c_2s_3 \\
c_1s_2s_3 & c_1s_2c_3 & c_1c_2s_3 & c_1c_2c_3 & -s_1s_2s_3 & -s_1s_2c_3 & -s_1c_2s_3 & -s_1c_2c_3 \\
s_1c_2c_3 & -s_1c_2s_3 & -s_1s_2c_3 & s_1s_2s_3 & c_1c_2c_3 & -c_1c_2s_3 & -c_1s_2c_3 & c_1s_2s_3 \\
s_1c_2s_3 & s_1c_2c_3 & -s_1s_2s_3 & -s_1s_2c_3 & c_1c_2s_3 & c_1c_2c_3 & -c_1s_2s_3 & -c_1s_2c_3 \\
s_1s_2c_3 & -s_1s_2s_3 & s_1c_2c_3 & -s_1c_2s_3 & c_1s_2c_3 & -c_1s_2s_3 & c_1c_2c_3 & -c_1c_2s_3 \\
s_1s_2s_3 & s_1s_2c_3 & s_1c_2s_3 & s_1c_2c_3 & c_1s_2s_3 & c_1s_2c_3 & c_1c_2s_3 & c_1c_2c_3
\end{pmatrix}
\end{aligned}$$

for $\theta = (\theta_1, \theta_2, \theta_3) \in \mathbb{R}^3$, where $c_i = \cos\frac{\theta_i}{2}$ and $s_i = \sin\frac{\theta_i}{2}$ for $i = 1, 2, 3$.

Generically: The quantum part of a VQA has the following form:



More precisely, we can explicitly include the observable we want to measure:



VQAs: The Quantum Part Given a PQC with $\vartheta \in \mathbb{R}^L$ we can define a cost function

$$B(\vartheta) = f\Big(\{|\Psi\rangle_0\}, \{B_k\}, U(\vartheta)\Big). \tag{7.46}$$

Essentially, this is a cost function that involves (some) obsevable quantity - Hermitian operators $\{O_k\}$ given input states $\{|\Psi\rangle_0\}$ and the PQC $U(\vartheta)$. Here $k \in I$, where I is some indexed set.

Let $\rho_{\text{in}} \coloneqq |\Psi\rangle_0\langle\Psi|_0$ (assume norm 1). A common choice is (using Born's rule) to define the "observable" function

$$B(\vartheta) = \sum_{k\in I} \text{Tr}\Big(B_k U(\vartheta)\rho_{\text{in}} U^\dagger(\vartheta)\Big), \tag{7.47}$$

or more generically

$$B(\vartheta) = \sum_{k\in I} f_k\left(\text{Tr}\Big(B_k U(\vartheta)\rho_{\text{in}} U^\dagger(\vartheta)\Big)\right), \tag{7.48}$$

for some functions f_k.

VQAs: Measurements: The observable function is one that we end up measuring (several times) in order to construct an empirical estimate of its expectation value $\langle B\rangle_\vartheta$ of the observable:

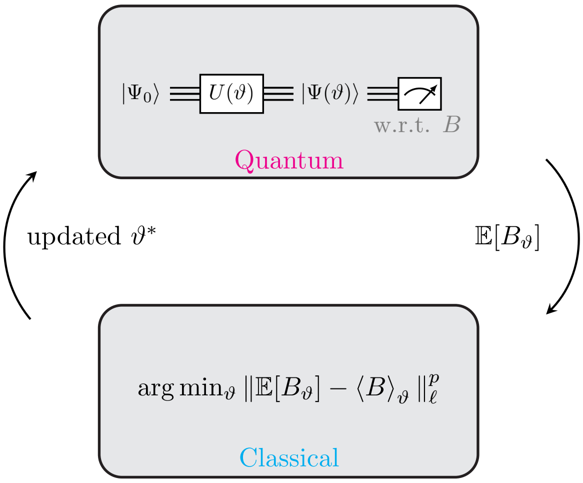
$$\langle B\rangle_\vartheta \coloneqq \langle\Psi(\vartheta)|B|\Psi(\vartheta)\rangle\,, \tag{7.49}$$

where $|\Psi(\vartheta)\rangle \coloneqq U(\vartheta)|\Psi_0\rangle$. The empirical estimate

$$\mathbb{E}[B_\vartheta]. \tag{7.50}$$

This is constructed by measuring the same circuit repeatedly. Out of this we construct a cost function we would like to minimize:

$$\vartheta^* \coloneqq \arg\min_\vartheta \|\mathbb{E}[B_\vartheta] - \langle B\rangle_\vartheta\|_\ell^p \tag{7.51}$$


$|\Psi_0\rangle$
$U(\vartheta)$
$|\Psi(\vartheta)\rangle$
w.r.t. B
Quantum
updated ϑ^*
$\mathbb{E}[B_\vartheta]$
$\arg\min_\vartheta \|\mathbb{E}[B_\vartheta] - \langle B\rangle_\vartheta\|_\ell^p$
Classical

VQA: The Classical Part: During the optimization, one uses a finite statistic estimator of the cost or its gradients.

Essentially we are "training" the VQA by learning the parameters ϑ.

It is known that for many optimization tasks using information in the cost function gradient can help in speeding up and guaranteeing the convergence of the optimizer. One of the main advantages of many VQAs is that often one can analytically evaluate the cost function gradient.

Parameter Shift Rule: Consider a cost function as in Eq. (7.48):

$$B(\vartheta) = \mathrm{Tr}\Big(BU(\vartheta)\rho_{\mathrm{in}}U^\dagger(\vartheta)\Big), \tag{7.52}$$

$(f_k = \mathrm{Id},\ k = 1)$. Furthermore, let the unitaries read:

$$U(\vartheta_j) = e^{\imath\vartheta_j\sigma_j^a}. \tag{7.53}$$

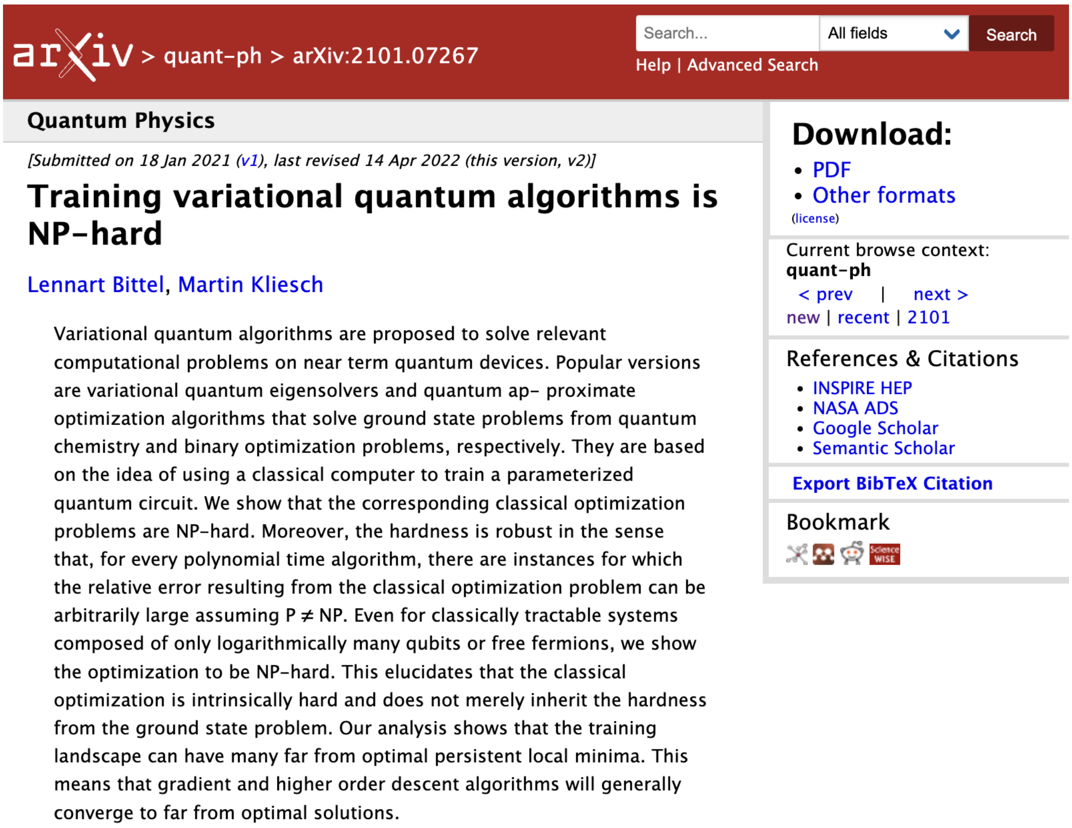
Then:

$$\frac{\partial B(\vartheta)}{\partial\vartheta_j} \sim \frac{1}{\sin\alpha}\left(\mathrm{Tr}\left(BU^\dagger(\vartheta_+)\rho U(\vartheta_+)\right) - \mathrm{Tr}\left(BU^\dagger(\vartheta_-)\rho U(\vartheta_-)\right)\right) \tag{7.54}$$

where $\vartheta_\pm = \vartheta \pm \alpha e$. Here e_j is a vector having 1 as its j-th element and 0 otherwise. Thus, one can evaluate the gradient by shifting the l-th parameter by some amount α.

Note that the accuracy of the evaluation depends on the coefficient $1/(2\sin\alpha)$ since each of the $\pm\alpha$ terms are evaluated by sampling B. This accuracy is maximized at $\alpha = \pi/4$. So, the parameter-shift rule looks like some form of finite difference. However, it evaluates the analytic gradient of the parameter by virtue of the coefficient $1/(2\sin\alpha)$. Note that several variations of the parameter shift rule exist [Kungurtsev et al., 2022, Fig. 1].

It's hard to train VQAs.



Quantum Physics

[Submitted on 18 Jan 2021 (v1), last revised 14 Apr 2022 (this version, v2)]

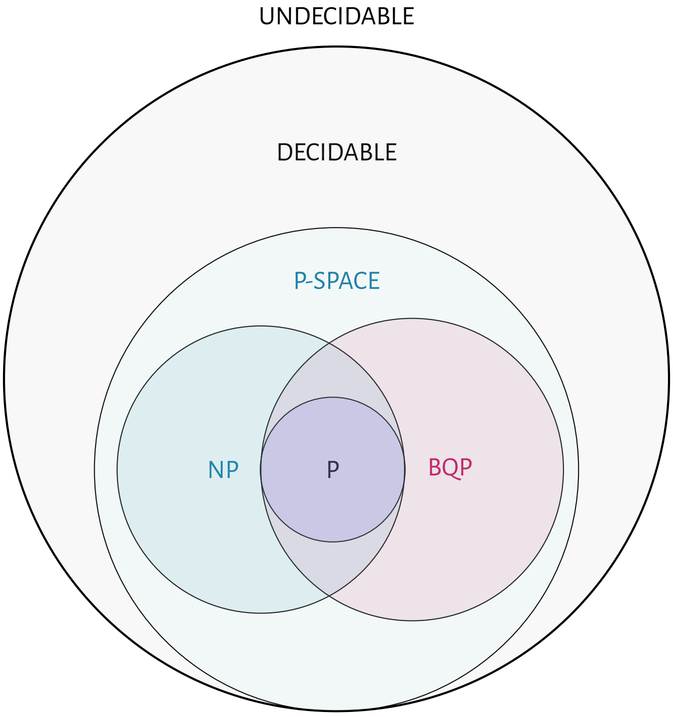
Training variational quantum algorithms is NP-hard

Lennart Bittel, Martin Kliesch

Variational quantum algorithms are proposed to solve relevant computational problems on near term quantum devices. Popular versions are variational quantum eigensolvers and quantum ap- proximate optimization algorithms that solve ground state problems from quantum chemistry and binary optimization problems, respectively. They are based on the idea of using a classical computer to train a parameterized quantum circuit. We show that the corresponding classical optimization problems are NP-hard. Moreover, the hardness is robust in the sense that, for every polynomial time algorithm, there are instances for which the relative error resulting from the classical optimization problem can be arbitrarily large assuming P ≠ NP. Even for classically tractable systems composed of only logarithmically many qubits or free fermions, we show the optimization to be NP-hard. This elucidates that the classical optimization is intrinsically hard and does not merely inherit the hardness from the ground state problem. Our analysis shows that the training landscape can have many far from optimal persistent local minima. This means that gradient and higher order descent algorithms will generally converge to far from optimal solutions.

Undecidability conjecture.

I conjecture that actually the situation is worse. VQAs are undecidable.



Training VQAs: The success of a VQA depends on the efficiency and reliability of the optimization method used.

As we saw the training can be NP-Hard. Training a VQA one can encounter new challenges:

- huge number of local minima
- barren plateaus
- stochastic environment due to the finite budget for measurements
- hardware noise affecting $\mathbb{E}[B_\vartheta]$
- restricted qubit connectivity
- statepreparation-and-measurement (SPAM) errors
- ...

This has led to the development of many quantum hardware-aware optimizers, with the optimal choice still being an active topic of debate. A common choice is the family of SGD (e.g. SPSA).

5. QAOA

Quantum Approximation Optimization Algorithm (QAOA) can be implemented in NISQ devices.

QAOA is an approximation algorithm: it does not deliver the "best" result, but only the "good enough" result, which is characterized by a lower bound of the approximation ratio.

Interestingly QAOA can be applied to the MaxCut problem via a traverse Ising filed model.

Trotterization: Recall that in the case of AQC we have:

$$H(s) = (1-s)H_{\text{init}} + sH_{\text{final}}. \tag{7.55}$$

Time evolution under this *time-dependent* Hamiltonian involves is hard:

$$U(T) \sim \exp\left(-\imath \int_0^t H(w)\mathrm{d}w\right). \tag{7.56}$$

We can mitigate this with the Trotterization technique: discretize $U(T) \equiv U(T,0)$ into intervals Δt (in total $T = L\Delta t$) small enough that the Hamiltonian is approximately constant over each interval. Then:

$$U(T,0) = U(T, T-\Delta t)U(T-\Delta t, T-2\Delta t)\dots U(\Delta t, 0) \tag{7.57}$$

$$= \prod_{j=0}^{L-1} U((L-j)\Delta T, (L-j-1)\Delta t) \tag{7.58}$$

$$=_{L\to\infty} \prod_{j=0}^{L-1} e^{-\imath H[(L-j)\Delta t]\Delta t} \tag{7.59}$$

Using the identity

$$e^{\imath(A+B)x} = e^{\imath(A)x}e^{\imath(B)x} + \mathcal{O}(x^2) \tag{7.60}$$

we deduce that

$$U(T,0) \approx \prod_{j=1}^{p} e^{\{-\imath(1-s(j\Delta t))H_{\text{init}}\Delta t\}} e^{\{-\imath s(j\Delta t)H_{\text{final}}\Delta t\}}. \tag{7.61}$$

Thus we can approximate AQC by repeatedly letting the system evolve under H_{final} for $s(j\Delta t)$ and then under H_{init} for $(1 - s(j\Delta t))$. In this way we can construct arbitrary unitaries.

QAOA: Combinatorial Optimization: Recall that a combinatorial optimization problem amounts to finding the n-bit string z that (approximately) satisfies the maximal amount of m constraints C_α, each of which takes the form

$$C_\alpha(z) = \begin{cases} 1 \text{ if } z \text{ satisfies the constraint} \\ 0 \text{ otherwise.} \end{cases} \tag{7.62}$$

We wish to find a string z that approximately maximizes the objective function

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z) \tag{7.63}$$

Quantum Analogue: For the quantum analogue of the previous problem we define a diagonal operator: H_C acting on the 2^n-dimensional Hilbert space where each bitstring z is a basis vector $|z\rangle$.

H_C acts on $|z\rangle$ as follows:

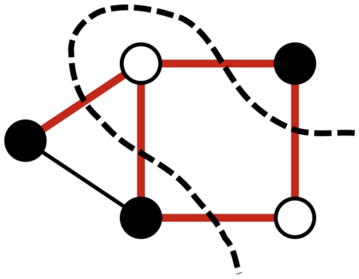
$$H_C|z\rangle = C(z)|z\rangle \tag{7.64}$$

and since $C(z)$ is scalar valued, we can see that each $|z\rangle$ is an eigenstate of H_C.

Let us view $\hat{C}$ as a Hamiltonian and the highest energy eigenstate $|z\rangle$ is the solution to the combinatorial optimization problem, as it gives the highest value of $C(z)$.

Max-Cut: In the case of Max-Cut we have:

$$C(z) = \frac{1}{2} \sum_{(i,j)\in E(G)} z_i z_j \tag{7.65}$$



QAOA at last: QAOA leverages approximate adiabatic quantum computation via Trotterization. We use two Hamiltonians: The first one is the **problem Hamiltonian** H_C which just by looking at Eq. (7.65) you should suspect its the Ising Hamiltonian.

The other one is called **mixer Hamiltonian** which is

$$H_B = \sum_{j=1}^{n} \sigma_j^x \tag{7.66}$$

The corresponding unitaries we need are:

$$U_C = e^{-\imath\gamma H_C} \tag{7.67}$$

$$U_B = e^{-\imath\beta H_B} \tag{7.68}$$

QAOA: Optimization: The goal is to maximize the expression

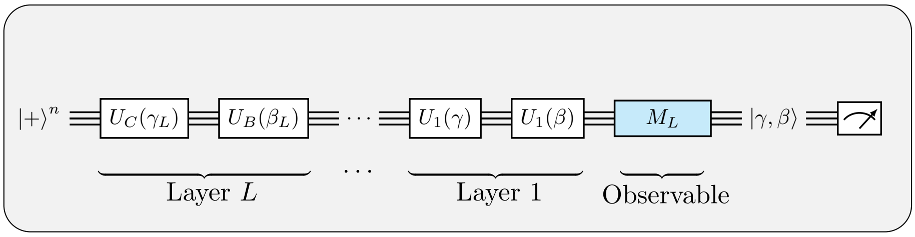
$$M_L(\gamma,\beta) \coloneqq \langle\gamma,\beta|M_L|\gamma,\beta\rangle \tag{7.69}$$

$\gamma \in [0, 2\pi]^L$, $\beta \in [0,\pi]^L$.

and

$$|\gamma,\beta\rangle = U_C(\gamma_L)U_B(\beta_L)\dots U_C(\gamma_1)U_B(\beta_1)|+\rangle^n. \tag{7.70}$$

Compare with Eq. (7.57). Its basically the same.



QAOA: Intuition: We begin in an eigenstate of H_B and then repeatedly let the system evolve under H_C and H_B, alternating between the two.

The approximation increase as $L \to \infty$.

We are trying to find

$$(\gamma^*, \beta^*) = \arg\max_{\gamma,\beta} \|\mathbb{E}[M_L] - \langle M_L\rangle\|_\ell^p \tag{7.71}$$

In the end we measure $|\gamma,\beta\rangle$ in the computational basis to get some bitstring z, and evaluate $C(z)$.

We repeat the above steps $\mathcal{O}(m\log m)$ (m number of constraints) such that we bound $C(z)$ with high probability.

Key result: QAOA with $L = 1$ achieves an approximation ratio of $r_c = C(z)/C_{\max} = 0.6924$ when performing Max-Cut on 3-regular graphs.

Bibliography

Dorit Aharonov, Wim Van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review*, 50(4):755–787, 2008.

Tameem Albash and Daniel A Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90 (1):015002, 2018.

Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

Arnab Das and Bikas K. Chakrabarti. Colloquium: Quantum annealing and analog quantum computation. *Rev. Mod. Phys.*, 80:1061–1081, Sep 2008. doi: 10.1103/RevModPhys.80.1061. URL https://link.aps.org/doi/10.1103/RevModPhys.80.1061.

Edward Farhi and Aram W Harrow. Quantum supremacy through the quantum approximate optimization algorithm. *arXiv preprint arXiv:1602.07674*, 2016.

Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.

Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*, 292(5516):472–475, 2001.

Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. Quantum adiabatic evolution algorithms versus simulated annealing. *arXiv preprint quant-ph/0201031*, 2002.

Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 58(5):5355, 1998.

Tosio Kato. On the adiabatic theorem of quantum mechanics. *Journal of the Physical Society of Japan*, 5(6):435–439, November 1950. doi: 10.1143/jpsj.5.435. URL https://doi.org/10.1143/jpsj.5.435.

A. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, May 2002. doi: 10.1090/gsm/047. URL https://doi.org/10.1090/gsm/047.

Vyacheslav Kungurtsev, Georgios Korpas, Jakub Marecek, and Elton Yechao Zhu. Iteration complexity of variational quantum algorithms. *arXiv preprint arXiv:2209.10615*, 2022.

Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):4213, 2014.

Jérémie Roland and Nicolas J Cerf. Quantum search by local adiabatic evolution. *Physical Review A*, 65(4):042308, 2002.

Giuseppe E Santoro, Roman Martoňák, Erio Tosatti, and Roberto Car. Theory of quantum annealing of an ising spin glass. *Science*, 295(5564):2427–2430, 2002.

Rolando D. Somma, Daniel Nagaj, and Mária Kieferová. Quantum speedup by quantum annealing. *Phys. Rev. Lett.*, 109:050501, Jul 2012. doi: 10.1103/PhysRevLett.109.050501. URL https://link.aps.org/doi/10.1103/PhysRevLett.109.050501.

W. van Dam, M. Mosca, and U. Vazirani. How powerful is adiabatic quantum computation? In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. IEEE, 2001. doi: 10.1109/sfcs.2001.959902. URL https://doi.org/10.1109/sfcs.2001.959902.

Wim Van Dam, Michele Mosca, and Umesh Vazirani. How powerful is adiabatic quantum computation? In *Proceedings 42nd IEEE symposium on foundations of computer science*, pages 279–287. IEEE, 2001.