# Lecture Topic: Adiabatic Quantum Computing and Quantum Replacements of Optimization Algorithms

## Main Result to show

Beautifully:

**Theorem**

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

## Main Result to show

Interestingly:

**Theorem**

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

Even more interestingly:

**Theorem**

*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 statemens above.

## Main Result to show

Interestingly:

**Theorem**

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

Even more interestingly:

**Theorem**

*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 statemens above.

## Main Result to show

Interestingly:

**Theorem**

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

Even more interestingly:

**Theorem**

*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 statemens above.

## Experimental Realization

It's 2004 and people really would like to build a quantum computer:

"[The previous theorems] open up the possibility of physically realizing universal quantum computation using adiabatically evolving quantum systems".

## Experimental Realization

It's 2004 and people really would like to build a quantum computer:

"[The previous theorems] open up the possibility of physically realizing universal quantum computation using adiabatically evolving quantum systems".

## Motivation behind AQC

The study of adiabatic quantum computation (AQC) was initiated several years ago by Farhi, Goldstone, Gutmann and Sipser:

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**.

## Motivation behind AQC

The study of adiabatic quantum computation (AQC) was initiated several years ago by Farhi, Goldstone, Gutmann and Sipser:

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**.

## Motivation behind AQC

The study of adiabatic quantum computation (AQC) was initiated several years ago by Farhi, Goldstone, Gutmann and Sipser:

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.

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

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.

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

Nevertheless, AQC has since shown to be promising in other (less ambitious directions):

It possesses several interesting algorithmic capabilities and exhibits inherent robustness against certain types of quantum errors.

Nevertheless, AQC has since shown to be promising in other (less ambitious directions):

It possesses several interesting algorithmic capabilities and exhibits inherent robustness against certain types of quantum errors.

## Dam, Mosca, Vazirani

[...] On the question of whether [AQC] can be used to efficiently solve NP-complete problems on a quantum computer [...] the usual query complexity arguments cannot be used to rule out a polynomial time solution.

On the other hand, we argue that the adiabatic approach may be thought of as a kind of “quantum local search”.

## Adiabatic Quantum Computation

Let us briefly introduce ACQ:A computation in this model is specified by two Hamiltonians named $H_{\text{init}}$ and $H_{\text{final}}$.

The ground state of $H_{\text{init}}$ 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 $H_{\text{final}}$. Hence, we choose an $H_{\text{final}}$ whose ground state represents the solution to our problem.

## Adiabatic Quantum Computation

Let us briefly introduce ACQ:A computation in this model is specified by two Hamiltonians named $H_{\text{init}}$ and $H_{\text{final}}$.

The ground state of $H_{\text{init}}$ 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 $H_{\text{final}}$. Hence, we choose an $H_{\text{final}}$ whose ground state represents the solution to our problem.

## Adiabatic Quantum Computation

Let us briefly introduce ACQ:A computation in this model is specified by two Hamiltonians named $H_{\text{init}}$ and $H_{\text{final}}$.

The ground state of $H_{\text{init}}$ 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 $H_{\text{final}}$. Hence, we choose an $H_{\text{final}}$ whose ground state represents the solution to our problem.

## Adiabatic Quantum Computation

Let us briefly introduce ACQ:A computation in this model is specified by two Hamiltonians named $H_{\text{init}}$ and $H_{\text{final}}$.

The ground state of $H_{\text{init}}$ 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 $H_{\text{final}}$. Hence, we choose an $H_{\text{final}}$ whose ground state represents the solution to our problem.

Additionally, we require the Hamiltonians to be **local**[1].

This, in particular, makes sure that the Hamiltonians have a short classical description since the interactions between qubits are limited to a finite neighborhood.

[1]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)

Additionally, we require the Hamiltonians to be **local**[1].

This, in particular, makes sure that the Hamiltonians have a short classical description since the interactions between qubits are limited to a finite neighborhood.

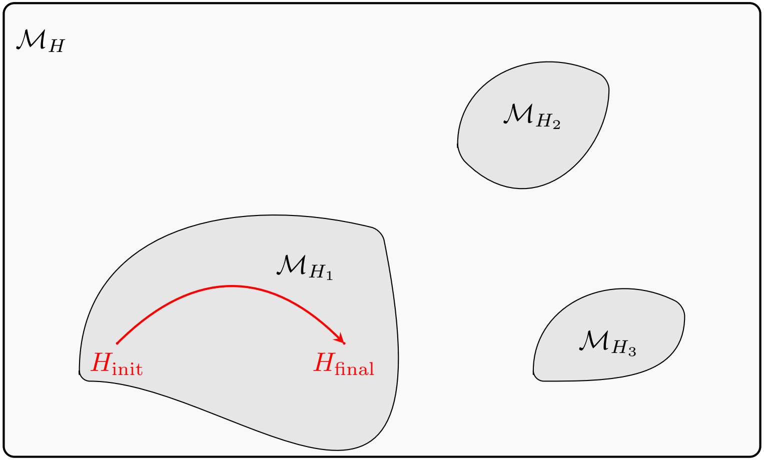
[1]We require them to only involve interactions between a constant number of particles (this can be seen as the equivalent of allowing gates operating on a constant number of qubits in the gate model)

The running time of the adiabatic computation is determined by the minimal spectral gap[2] of all the path connected Hamiltonians along the curve:

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

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

## The space of Hamiltonians



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

$$H(s) = (1-s)H_{\text{init}} + sH_{\text{final}} \tag{0.1}$$

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

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

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

$$H(s) = (1 - s)H_{\text{init}} + sH_{\text{final}} \tag{0.1}$$

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

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

## Motivation: Physics

Let us provide some motivation:

- 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 $H_{\text{init}}$ (which is supposed to be easy to generate) and apply the Hamiltonian $H_{\text{init}}$ to the system. We then slowly modify the Hamiltonian along the path from $H_{\text{init}}$ 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 $H_{\text{final}}$ , as required.

## Motivation: Physics

Let us provide some motivation:

- 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 $H_{\text{init}}$ (which is supposed to be easy to generate) and apply the Hamiltonian $H_{\text{init}}$ to the system. We then slowly modify the Hamiltonian along the path from $H_{\text{init}}$ 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 $H_{\text{final}}$ , as required.

## Motivation: Physics

Let us provide some motivation:

- 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 $H_{\text{init}}$ (which is supposed to be easy to generate) and apply the Hamiltonian $H_{\text{init}}$ to the system. We then slowly modify the Hamiltonian along the path from $H_{\text{init}}$ 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 $H_{\text{final}}$ , as required.

## Motivation: Physics

Let us provide some motivation:

- 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 $H_{\text{init}}$ (which is supposed to be easy to generate) and apply the Hamiltonian $H_{\text{init}}$ to the system. We then slowly modify the Hamiltonian along the path from $H_{\text{init}}$ 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 $H_{\text{final}}$ , as required.

## Motivation: Physics

Let us provide some motivation:

- 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 $H_{\text{init}}$ (which is supposed to be easy to generate) and apply the Hamiltonian $H_{\text{init}}$ to the system. We then slowly modify the Hamiltonian along the path from $H_{\text{init}}$ 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 $H_{\text{final}}$ , as required.

## Motivation: Physics

Let us provide some motivation:

- 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 $H_{\text{init}}$ (which is supposed to be easy to generate) and apply the Hamiltonian $H_{\text{init}}$ to the system. We then slowly modify the Hamiltonian along the path from $H_{\text{init}}$ 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 $H_{\text{final}}$ , as required.

## Motivation: Computational Power

To refer to the **adiabatic model** as a **computational model** that computes classical functions, we consider the result of the adiabatic computation to be the outcome of a measurement of one or more of the qubits, performed on the final ground state.

So, AQC is performed on qubits similar to the ones of the gate-based computers.

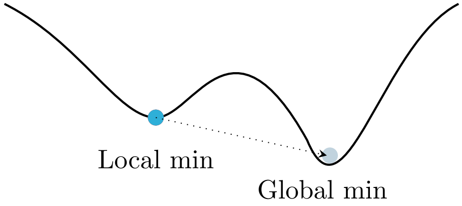
## AQC Facts

Note: adiabatic computation can be efficiently simulated by gate-based quantum computers .

Therefore, its computational power is not greater than that of gate-based computers.

Some positive results are known: e.g. Grover search can be realized AQC!

Moreover, AQC can "tunnel" through wide energy barriers (possibly outperforming simulated annealing).



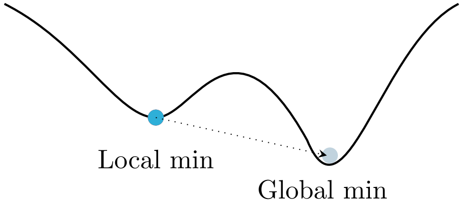
## AQC Facts

Note: adiabatic computation can be efficiently simulated by gate-based quantum computers .

Therefore, its computational power is not greater than that of gate-based computers.

Some positive results are known: e.g. Grover search can be realized AQC!

Moreover, AQC can "tunnel" through wide energy barriers (possibly outperforming simulated annealing).



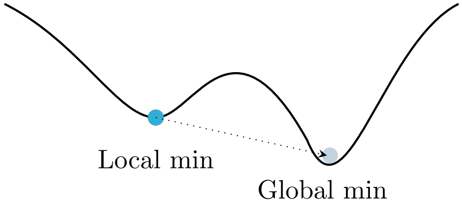
## AQC Facts

Note: adiabatic computation can be efficiently simulated by gate-based quantum computers .

Therefore, its computational power is not greater than that of gate-based computers.

Some positive results are known: e.g. Grover search can be realized AQC!

Moreover, AQC can "tunnel" through wide energy barriers (possibly outperforming simulated annealing).



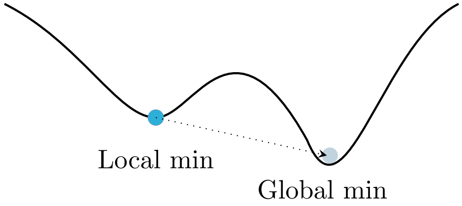
## AQC Facts

Note: adiabatic computation can be efficiently simulated by gate-based quantum computers .

Therefore, its computational power is not greater than that of gate-based computers.

Some positive results are known: e.g. Grover search can be realized AQC!

Moreover, AQC can "tunnel" through wide energy barriers (possibly outperforming simulated annealing).



## Unknowns Power

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.

## Unknowns Power

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.

## Power over classical optimization

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

There, $H_{\text{final}}$ is chosen to be a diagonal matrix, corresponding to a combinatorial optimization problem.

Being diagonal implies that the ground state of $H_{\text{final}}$ is a classical state, (a state in the computational basis).

We want to show something more powerful. We only will assume that the Hamiltonians involved are local.

## Power over classical optimization

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

There, $H_{\text{final}}$ is chosen to be a diagonal matrix, corresponding to a combinatorial optimization problem.

Being diagonal implies that the ground state of $H_{\text{final}}$ is a classical state, (a state in the computational basis).

We want to show something more powerful. We only will assume that the Hamiltonians involved are local.

## Power over classical optimization

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

There, $H_{\text{final}}$ is chosen to be a diagonal matrix, corresponding to a combinatorial optimization problem.

Being diagonal implies that the ground state of $H_{\text{final}}$ is a classical state, (a state in the computational basis).

We want to show something more powerful. We only will assume that the Hamiltonians involved are local.

## Power over classical optimization

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

There, $H_{\text{final}}$ is chosen to be a diagonal matrix, corresponding to a combinatorial optimization problem.

Being diagonal implies that the ground state of $H_{\text{final}}$ is a classical state, (a state in the computational basis).

We want to show something more powerful. We only will assume that the Hamiltonians involved are local.

## *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:

$$|\psi\rangle \in \mathbb{C}^{\otimes n}. \qquad (0.2)$$

In terms of the individual qubits:

$$|\psi\rangle = |i_1\rangle \otimes \dots |i_n\rangle = |i_1 \dots i_n\rangle, \qquad (0.3)$$

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

## *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:

$$|\psi\rangle \in \mathbb{C}^{\otimes n}. \tag{0.2}$$

In terms of the individual qubits:

$$|\psi\rangle = |i_1\rangle \otimes \ldots |i_n\rangle = |i_1 \ldots i_n\rangle, \tag{0.3}$$

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

## Evolution

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:

$$\imath \frac{\mathrm{d}}{\mathrm{d}t} |\psi(t)\rangle = H|\psi(t)\rangle \tag{0.4}$$

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:

$$|\psi(t)\rangle = \exp(-\imath Ht)|\psi(0)\rangle. \tag{0.5}$$

## Evolution

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:

$$\imath \frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle = H|\psi(t)\rangle \tag{0.4}$$

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:

$$|\psi(t)\rangle = \exp(-\imath Ht)|\psi(0)\rangle. \tag{0.5}$$

## Evolution

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:

$$\imath \frac{\mathrm{d}}{\mathrm{d}t} |\psi(t)\rangle = H|\psi(t)\rangle \tag{0.4}$$

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:

$$|\psi(t)\rangle = \exp(-\imath H t)|\psi(0)\rangle. \tag{0.5}$$

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 $t = T$.

Recall: eigenvalues of Hamiltonians as (eigen)energies.

The ground energy of a Hamiltonian is its lowest eigenvalue and the corresponding eigenvector(s) are called ground state(s).

Recall: eigenvalues of Hamiltonians as (eigen)energies.

The ground energy of a Hamiltonian is its lowest eigenvalue and the corresponding eigenvector(s) are called ground state(s).

## Spectral Gap

**Spectral gap** $\Delta(H)$ of a Hamiltonian $H$: difference between the lowest eigenvalue of $H$ and its second lowest eigenvalue.

Note that $\Delta(H) = 0$ if the lowest eigenvalue is degenerate.

## Spectral Gap

**Spectral gap** $\Delta(H)$ of a Hamiltonian $H$: difference between the lowest eigenvalue of $H$ and its second lowest eigenvalue.

Note that $\Delta(H) = 0$ if the lowest eigenvalue is degenerate.

## Locality

One cannot efficiently apply any arbitrary Hamiltonian on a $n$-qubit system (just describing it requires roughly $2^{2n}$ space).

Restrict to $k$-local Hamiltonians.

A Hamiltonian $H$ is $k$-local if $H = \sum_A H^A$ where $A$ runs over all subsets of $k$ qubits.

Commonly $k = 2$ in NISQ devices.

## Locality

One cannot efficiently apply any arbitrary Hamiltonian on a $n$-qubit system (just describing it requires roughly $2^{2n}$ space).

Restrict to $k$-local Hamiltonians.

A Hamiltonian $H$ is $k$-local if $H = \sum_A H^A$ where $A$ runs over all subsets of $k$ qubits.

Commonly $k = 2$ in NISQ devices.

## Locality

One cannot efficiently apply any arbitrary Hamiltonian on a $n$-qubit system (just describing it requires roughly $2^{2n}$ space).

Restrict to $k$-local Hamiltonians.

A Hamiltonian $H$ is $k$-local if $H = \sum_A H^A$ where $A$ runs over all subsets of $k$ qubits.

Commonly $k = 2$ in NISQ devices.

## Locality

One cannot efficiently apply any arbitrary Hamiltonian on a $n$-qubit system (just describing it requires roughly $2^{2n}$ space).

Restrict to $k$-local Hamiltonians.

A Hamiltonian $H$ is $k$-local if $H = \sum_A H^A$ where $A$ runs over all subsets of $k$ qubits.

Commonly $k = 2$ in NISQ devices.

## Adiabatic Theorem

The cornerstone of the adiabatic model of computation is the celebrated **adiabatic theorem**.

Consider a time-dependent Hamiltonian $H(s)$, and a system initialized at time $t = 0$ in the ground state of $H(0)$ (assume that for all $H(s)$ has a unique ground state for all $s$).

## Adiabatic Theorem

The cornerstone of the adiabatic model of computation is the celebrated **adiabatic theorem**.

Consider a time-dependent Hamiltonian $H(s)$, and a system initialized at time $t = 0$ in the ground state of $H(0)$ (assume that for all $H(s)$ has a unique ground state for all $s$).

We let the system evolve according to the Hamiltonian $H(s)$, where $s := t/T$, from $t = 0$ 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 $\Delta(H(s))$.

We let the system evolve according to the Hamiltonian $H(s)$, where $s := t/T$, from $t = 0$ 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 $H(1)$.

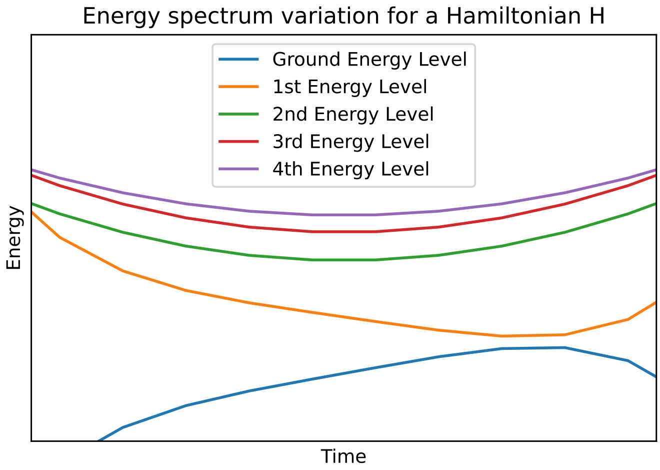
How large $T$ should be for this to happen is determined by the spectral gap of the Hamiltonians $\Delta(H(s))$.

We let the system evolve according to the Hamiltonian $H(s)$, where $s := t/T$, from $t = 0$ 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 $\Delta(H(s))$.

It is crucial that the spectral gap does not change sign.



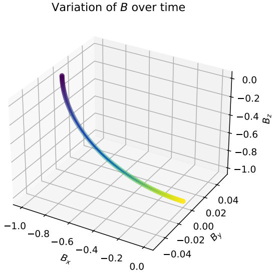
## Physical intuition

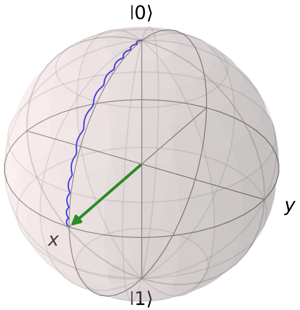
Consider a spin particle (e.g. an electron) in a magnetic field $B$ which rotates from the $x$ direction to the $z$ direction in a total time $T$. The dynamics of the particle are described by the Hamiltonian:

$$H(t) = -\cos\left(\frac{\pi t}{2T}\right)\sigma_x - \sin\left(\frac{\pi t}{2T}\right)\sigma_z. \tag{0.6}$$

## Physical intuition

Assume: at $t = 0$ 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 the particle's spin begins to precess about the new direction of the field, moving it toward the $z$ axis.

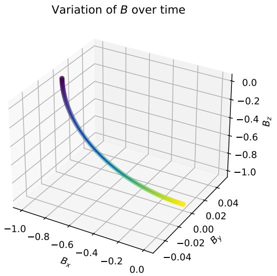


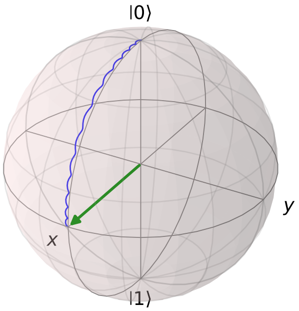


Note that this produces a small wiggling component out of the xz-plane.

## Physical intuition

Assume: at $t = 0$ 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 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

**Adiabaticity**: allow $T$ to be larger and larger, so that the rotation of the field direction happens slower and slower.

At large $T$: 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 $H(t)$.

## Adiabaticity

**Adiabaticity**: allow $T$ to be larger and larger, so that the rotation of the field direction happens slower and slower.

At large $T$: 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 $H(t)$.

## Adiabaticity

**Adiabaticity**: allow $T$ to be larger and larger, so that the rotation of the field direction happens slower and slower.

At large $T$: 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

Generally: let $H(s)$ be a Hermitian operator that varies smoothly as a function of $s := t/T$.

For $T$ large, $H(t)$ varies very slowly as a function of $t$.

An initial quantum state $|\psi(0)\rangle$ evolves according to the Schrödinger equation (0.4), or, equivalently:

$$\imath \frac{\mathrm{d}}{\mathrm{d}s} |\psi(s)\rangle = TH|\psi(s)\rangle. \tag{0.7}$$

## Full statement

Generally: let $H(s)$ be a Hermitian operator that varies smoothly as a function of $s := t/T$.

For $T$ large, $H(t)$ varies very slowly as a function of $t$.

An initial quantum state $|\psi(0)\rangle$ evolves according to the Schrödinger equation (0.4), or, equivalently:

$$\imath \frac{\mathrm{d}}{\mathrm{d}s}|\psi(s)\rangle = TH|\psi(s)\rangle. \tag{0.7}$$

## Full statement

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 (Adiabatic Theorem)**

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

## Full statement

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 (Adiabatic Theorem)**

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

## Full statement

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 (Adiabatic Theorem)**

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

## Proof of the Adiabatic Theorem

...

## A more formal version

**Theorem (Adiabatic Theorem (Proper))**

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

$$T \geq \Omega \left( \frac{\| H_{final} - H_{init} \|^{1+\delta}}{\epsilon^{\delta} \min_{s \in [0,1]} \{ \Delta^{2+\delta}(H(s)) \}} \right) \tag{1.1}$$

*then the final state of an adiabatic evolution according to $H$ for time $T$ (with an appropriate setting of global phase) is $\epsilon$-close in $\ell_2$-norm to the ground state of $H_{final}$ .*

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

## A more formal version

**Theorem (Adiabatic Theorem (Proper))**

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

$$T \geq \Omega\left(\frac{\|H_{final} - H_{init}\|^{1+\delta}}{\epsilon^{\delta} \min_{s\in[0,1]} \{\Delta^{2+\delta}(H(s))\}}\right) \tag{1.1}$$

*then the final state of an adiabatic evolution according to $H$ for time $T$ (with an appropriate setting of global phase) is $\epsilon$-close in $\ell_2$-norm to the ground state of $H_{final}$.*

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

## The AQC model: proper

The adiabatic circuit is determined by $H_{\text{init}}$ and $H_{\text{final}}$ and the output of the computation is (close to) the ground state of $H_{\text{final}}$ .

**Definition**

A $k$-local AQC $(n, d, H_{\text{init}}, H_{\text{final}}, \epsilon)$ is specified by two $k$-local Hamiltonians, $H_{\text{init}}$ and $H_{\text{final}}$ acting on $n$ d-dimensional particles, such that both Hamiltonians have unique ground states.

The ground state of $H_{\text{init}}$ is a tensor product state. The output is a state that is $\epsilon$-close in $\ell_2$-norm to the ground state of $H_{\text{final.}}$ .

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 $\epsilon$-close in $\ell_2$-norm to the ground state of $H_{\text{final}}$ . The running time of the adiabatic algorithm is defined to be $T \cdot \max_s \|H(s)\|$.

## The AQC model: proper

The adiabatic circuit is determined by $H_{\text{init}}$ and $H_{\text{final}}$ and the output of the computation is (close to) the ground state of $H_{\text{final}}$ .

### Definition

A $k$-local AQC $(n, d, H_{\text{init}}, H_{\text{final}}, \epsilon)$ is specified by two $k$-local Hamiltonians, $H_{\text{init}}$ and $H_{\text{final}}$ acting on $n$ d-dimensional particles, such that both Hamiltonians have unique ground states.
The ground state of $H_{\mathrm{init}}$ is a tensor product state. The output is a state that is $\epsilon$-close in $\ell_2$-norm to the ground state of $H_{\text{final.}}$ .
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 $\epsilon$-close in $\ell_2$-norm to the ground state of $H_{\text{final}}$ . The running time of the adiabatic algorithm is defined to be $T \cdot \max_s \|H(s)\|$.

## The AQC model: proper

The adiabatic circuit is determined by $H_{\text{init}}$ and $H_{\text{final}}$ and the output of the computation is (close to) the ground state of $H_{\text{final}}$ .

### Definition

A $k$-local AQC $(n, d, H_{\text{init}}, H_{\text{final}}, \epsilon)$ is specified by two $k$-local Hamiltonians, $H_{\text{init}}$ and $H_{\text{final}}$ acting on $n$ d-dimensional particles, such that both Hamiltonians have unique ground states.
The ground state of $H_{\mathrm{init}}$ is a tensor product state. The output is a state that is $\epsilon$-close in $\ell_2$-norm to the ground state of $H_{\text{final.}}$ .
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 $\epsilon$-close in $\ell_2$-norm to the ground state of $H_{\text{final}}$ . The running time of the adiabatic algorithm is defined to be $T \cdot \max_s \|H(s)\|$.

## Gates to AQC

Main theorem 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.

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.)

## Gates to AQC

Main theorem 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.

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.)

## Gates to AQC

Main theorem 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.

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.)

## Gates to AQC: Theorem

**Theorem**

*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 $(n+2, 2, H_{\mathrm{init}}, H_{\mathrm{final}}, \epsilon)$ whose running time is $\mathrm{poly}(L, 1/\epsilon)$ and whose output is $\epsilon$-close to $U|0\rangle^n = U|0\rangle^{\otimes n}$. Additionally, $H_{\mathrm{init}}$ and $H_{\mathrm{final}}$ can be computed by a polynomial time Turing machine.*

## The Hamiltonian

The Hamiltonian we need is defined in the book of Kitaev (ref in the notes).

We begin by defining a state

$$|\gamma_\ell\rangle := |\alpha(\ell)\rangle \otimes |1^\ell 0^{L-\ell}\rangle^c. \tag{1.2}$$

Here $|\alpha(\ell)\rangle$ denotes the state of the circuit after the application of the $\ell$-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 $\ell$ qubits in the state $|1\rangle$ followed by $(L-\ell)$ qubits in the state $|0\rangle$.

## The Hamiltonian

The Hamiltonian we need is defined in the book of Kitaev (ref in the notes).

We begin by defining a state

$$|\gamma_\ell\rangle := |\alpha(\ell)\rangle \otimes |1^\ell 0^{L-\ell}\rangle^c . \tag{1.2}$$

Here $|\alpha(\ell)\rangle$ denotes the state of the circuit after the application of the $\ell$-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 $\ell$ qubits in the state $|1\rangle$ followed by $(L-\ell)$ qubits in the state $|0\rangle$.

## The Hamiltonian

The Hamiltonian we need is defined in the book of Kitaev (ref in the notes).

We begin by defining a state

$$|\gamma_\ell\rangle := |\alpha(\ell)\rangle \otimes |1^\ell 0^{L-\ell}\rangle^c. \tag{1.2}$$

Here $|\alpha(\ell)\rangle$ denotes the state of the circuit after the application of the $\ell$-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 $\ell$ qubits in the state $|1\rangle$ followed by $(L-\ell)$ qubits in the state $|0\rangle$.

## The Hamiltonian

The Hamiltonian we need is defined in the book of Kitaev (ref in the notes).

We begin by defining a state

$$|\gamma_\ell\rangle := |\alpha(\ell)\rangle \otimes |1^\ell 0^{L-\ell}\rangle^c. \tag{1.2}$$

Here $|\alpha(\ell)\rangle$ denotes the state of the circuit after the application of the $\ell$-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 $\ell$ qubits in the state $|1\rangle$ followed by $(L-\ell)$ qubits in the state $|0\rangle$.

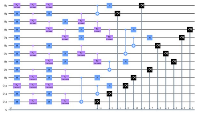
We now define the Hamiltonian $H_{\text{init}}$ with ground state $|\gamma_0\rangle = |0^n\rangle \otimes \left|0^L\right\rangle^c$,

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

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

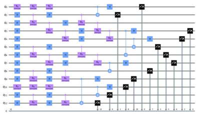
and the local Hamiltonian $H_{\text{final}}$ with ground state $|\eta\rangle = \frac{1}{\sqrt{L+1}} \sum_{\ell=0}^{L} |\gamma_\ell\rangle$.

## The Hilbert Space



$$\mathbb{C}^n \otimes \mathbb{C}^L$$

## The Hilbert Space



$$\mathbb{C}^n \otimes \mathbb{C}^L$$

Typo: $n \to 2n$.

It turns out that the way to do it is:

$$\begin{aligned} H_{\text{init}} &:= H_{\text{clock init}} + H_{\text{input}} + H_{\text{clock}} \\ H_{\text{final}} &:= \frac{1}{2}\sum_{\ell=1}^{L} H_{\ell} + H_{\text{input}} + H_{\text{clock}} \end{aligned} \tag{1.3}$$

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

It turns out that the way to do it is:

$$\begin{aligned} H_{\text{init}} &:= H_{\text{clock init}} + H_{\text{input}} + H_{\text{clock}} \\ H_{\text{final}} &:= \frac{1}{2}\sum_{\ell=1}^{L} H_{\ell} + H_{\text{input}} + H_{\text{clock}} \end{aligned} \tag{1.3}$$

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

## Adiabatic Evolution

The adiabatic evolution then follows the time-dependent Hamiltonian

$$H(s) = (1 - s)H_{\text{init}} + sH_{\text{final}} \tag{1.4}$$

## Adiabatic Evolution

The adiabatic evolution then follows the time-dependent Hamiltonian

$$H(s) = (1 - s)H_{\text{init}} + sH_{\text{final}} \tag{1.4}$$

## The Hamiltonians Explained: $H_{\text{clock}}$

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

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

$$H_{\text{clock}} := \sum_{\ell=1}^{L-1} |01\rangle\langle 01|^{c}_{\ell,\ell+1}. \tag{1.5}$$

## The Hamiltonians Explained: $H_{\text{clock}}$

First, $H_{\text{clock}}$ 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:

$$H_{\text{clock}} := \sum_{\ell=1}^{L-1} |01\rangle\langle 01|^c_{\ell,\ell+1}. \tag{1.5}$$

## The Hamiltonians Explained: $H_{\text{init}}$

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

$$H_{\mathrm{init}} := \sum_{i=1}^{n} |1\rangle\langle 1| \otimes |0\rangle\langle 0|. \tag{1.6}$$

## The Hamiltonians Explained: $H_{\text{clock init}}$ and $J_\ell$

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

$$H_{\text{clock init}} := |1\rangle\langle 1|. \tag{1.7}$$

Finally, we have the term

$$\frac{1}{2}\sum_{\ell=1}^{L} H_\ell \tag{1.8}$$

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

## The Hamiltonians Explained: $H_{\text{clock init}}$ and $J_\ell$

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

$$H_{\text{clock init}} := |1\rangle\langle 1|. \tag{1.7}$$

Finally, we have the term

$$\frac{1}{2}\sum_{\ell=1}^{L} H_\ell \tag{1.8}$$

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

## Summary

$H_{\text{clock init}}$ and $H_{\text{clock}}$: 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 $|1\rangle^c$. $H_{\text{clock}}$ penalizes out-of-order transitions and enforces a step-by-step progression through the circuit.

$H_{\text{input}}$: 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, $H_{\text{final}}$. 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.

## 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. (1.2):

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

## 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. (1.2):

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

## 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. (1.2):

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

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

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

$$\{|\gamma_0\rangle, \dots, |\gamma_1\rangle\} \tag{1.9}$$

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

**Theorem**

*The spectral gap of the restriction of $H(s)$ to $S_0$ satisfies:*

$$\Delta(H_{S_0}(s)) = \Omega(L^{-2}), \tag{1.10}$$

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

Interestingly, the proof uses a **continuous-time quantum walk**.

Aharonov *et. al.*: "[...] From this it follows that the running time of the adiabatic computation is polynomial".

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

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

$$\{|\gamma_0\rangle, \dots, |\gamma_1\rangle\} \tag{1.9}$$

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

**Theorem**

*The spectral gap of the restriction of $H(s)$ to $S_0$ satisfies:*

$$\Delta(H_{S_0}(s)) = \Omega(L^{-2}), \tag{1.10}$$

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

Interestingly, the proof uses a **continuous-time quantum walk**.

Aharonov *et. al.*: "[...] From this it follows that the running time of the adiabatic computation is polynomial".

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

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

$$\{|\gamma_0\rangle, \ldots, |\gamma_1\rangle\} \tag{1.9}$$

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

**Theorem**

*The spectral gap of the restriction of $H(s)$ to $S_0$ satisfies:*

$$\Delta(H_{S_0}(s)) = \Omega(L^{-2}), \tag{1.10}$$

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

Interestingly, the proof uses a **continuous-time quantum walk**.

Aharonov *et. al.*: "[...] From this it follows that the running time of the adiabatic computation is polynomial".

## We omit the proof

The proof is technical but not very hard.

The important thing is to understand the need for the Hamiltonians $H_{\rm init}$ and $H_{\rm final}$ in Eq. (1.3).

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

## We omit the proof

The proof is technical but not very hard.

The important thing is to understand the need for the Hamiltonians $H_{\text{init}}$ and $H_{\text{final}}$ in Eq. (1.3).

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

## We omit the proof

The proof is technical but not very hard.

The important thing is to understand the need for the Hamiltonians $H_{\text{init}}$ and $H_{\text{final}}$ in Eq. (1.3).

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 $H_{\mathrm{init}}$ and $H_{\mathrm{final}}$ 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 $\ell_2$ distance $\epsilon$ of the history state of the circuit. The running time of the AQC algorithm is $\mathcal{O}(TL)$.

Already from 2000 it was known that gate-based algorithms can be encoded as AQC.

With the proof of the main theorem the **universality of AQC** is also proven.

## The Equivalence Statement

Given a quantum circuit on $n$ qubits with $L$ gates, the quantum adiabatic algorithm with $H_{\text{init}}$ and $H_{\text{final}}$ 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 $\ell_2$ distance $\epsilon$ of the history state of the circuit. The running time of the AQC algorithm is $\mathcal{O}(TL)$.

Already from 2000 it was known that gate-based algorithms can be encoded as AQC.

With the proof of the main theorem the **universality of AQC** is also proven.

## The Equivalence Statement

Given a quantum circuit on $n$ qubits with $L$ gates, the quantum adiabatic algorithm with $H_{\mathrm{init}}$ and $H_{\mathrm{final}}$ 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 $\ell_2$ distance $\epsilon$ of the history state of the circuit. The running time of the AQC algorithm is $\mathcal{O}(TL)$.

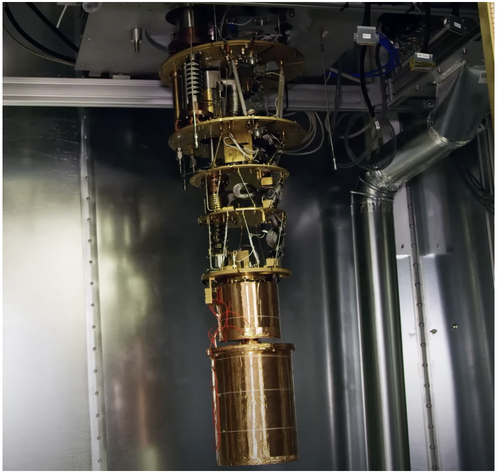
Already from 2000 it was known that gate-based algorithms can be encoded as AQC.

With the proof of the main theorem the **universality of AQC** is also proven.

## Break

Questions?

## Quantum Annealing



## Quantum Annealing

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 to solve computational problems by quantum evolution towards the ground states of final Hamiltonians that encode classical (optimization) problems.

## Quantum Annealing

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 to solve computational problems by quantum evolution towards the ground states of final Hamiltonians that encode classical (optimization) problems.

## Quantum Annealers are Real

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

## Stoquasticity

QA therefore, moves between the idealized assumptions of universal AQC and the unavoidable experimental compromises that happen in a lab.

Compromise in QA: only design of stoquastic quantum annealers.

**Definition (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

$$\langle i|H|j\rangle \leq 0, \quad \forall i,j \in \{0,1\}^n, \quad i \neq j. \tag{2.1}$$

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

## Stoquasticity

QA therefore, moves between the idealized assumptions of universal AQC and the unavoidable experimental compromises that happen in a lab.

Compromise in QA: only design of stoquastic quantum annealers.

**Definition (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

$$\langle i|H|j\rangle \leq 0, \quad \forall i,j \in \{0,1\}^n, \quad i \neq j. \tag{2.1}$$

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

## Stoquasticity

QA therefore, moves between the idealized assumptions of universal AQC and the unavoidable experimental compromises that happen in a lab.

Compromise in QA: only design of stoquastic quantum annealers.

**Definition (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

$$\langle i|H|j\rangle \leq 0, \quad \forall i,j \in \{0,1\}^n, \quad i \neq j. \tag{2.1}$$

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

## Stoquasticity

QA therefore, moves between the idealized assumptions of universal AQC and the unavoidable experimental compromises that happen in a lab.

Compromise in QA: only design of stoquastic quantum annealers.

**Definition (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

$$\langle i|H|j\rangle \leq 0, \quad \forall i,j \in \{0,1\}^n, \quad i \neq j. \tag{2.1}$$

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

## AQC with Stoquastic Hamiltonians

**Definition**

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

Essentially, Quantum Annealing (QA) refers to StoqAQC when considered in (realistic) open quantum systems.

## No Universality

The computational power of stoquastic Hamiltonians has been carefully studied, and is suspected to be limited..

**It is quite unlikely that ground-state StoqAQC is universal**.

## Quantum Annealing

QA follows the same idea of AQC. We still have the same tools:

- An initial, easy-to-prepare state and a Hamiltonian $H_{\mathrm{init}}$,
- A problem of interest whose solution is encoded into the ground state of a Hamiltonian $H_{\mathrm{final}}$,
- Adiabatic evolution using Eq. (0.1):

$$H(s) = (1 - s)H_{\text{init}} + sH_{\text{final}} \tag{2.2}$$

## Quantum Annealing

QA follows the same idea of AQC. We still have the same tools:

- An initial, easy-to-prepare state and a Hamiltonian $H_{\mathrm{init}}$,
- A problem of interest whose solution is encoded into the ground state of a Hamiltonian $H_{\mathrm{final}}$,
- Adiabatic evolution using Eq. (0.1):

$$H(s) = (1 - s)H_{\text{init}} + sH_{\text{final}} \tag{2.2}$$

## Exponential Speedups with QA

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

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

## Exponential Speedups with QA

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

Somma, Nagaj, and Kieferovaá 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.

## 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.

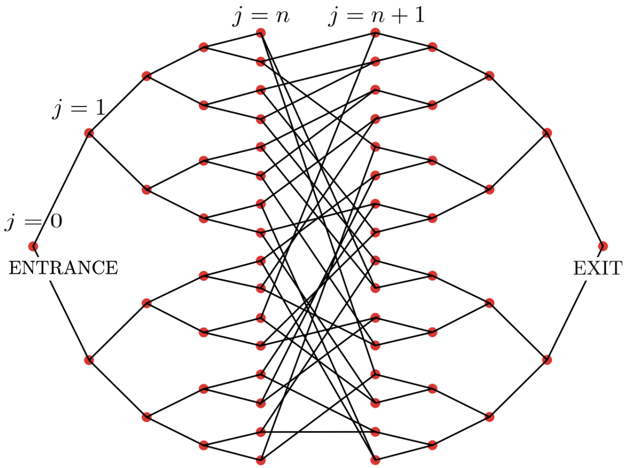
The oracle $O_A$ outputs the names of the adjacent vertices on any given input vertex name.

## 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.

## Glued-Trees Problem



## Glued-Trees Problem

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.

## Glued-Trees Problem

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.

## Glued-Trees Problem

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.

## Glued-Trees Problem

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.

## Break

Questions?

## 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:

$$\min_{x,y} \quad f(x,y) \tag{2.3}$$

$$\text{s.t.} \quad g(x) \geq 0 \tag{2.4}$$

$$x \in \mathbb{R}^m,\ y \in \mathbb{Z}^n \tag{2.5}$$

We would like to see if QA can help towards solving hard optimization problems.

## 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:

$$\min_{x,y} \quad f(x,y) \tag{2.3}$$
$$\text{s.t.} \quad g(x) \geq 0 \tag{2.4}$$
$$x \in \mathbb{R}^m,\ y \in \mathbb{Z}^n \tag{2.5}$$

We would like to see if QA can help towards solving hard optimization problems.

## 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:

$$\begin{aligned} \min_{x,y} \quad & f(x,y) && (2.3) \\ \text{s.t.} \quad & g(x) \geq 0 && (2.4) \\ & x \in \mathbb{R}^m,\ y \in \mathbb{Z}^n && (2.5) \end{aligned}$$

We would like to see if QA can help towards solving hard optimization problems.

## 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:

$$\min_{x,y} \quad f(x,y) \tag{2.3}$$

$$\text{s.t.} \quad g(x) \geq 0 \tag{2.4}$$

$$x \in \mathbb{R}^m,\ y \in \mathbb{Z}^n \tag{2.5}$$

We would like to see if QA can help towards solving hard optimization problems.

## Potential solution: QA

Consider the $k$-th eigenstate state of the Hamiltonian:

$$H(s)|k\rangle = \lambda_k(s)|k\rangle \tag{2.6}$$

with $|0(0)\rangle$ being the ground state of $H_{\mathrm{init}}$ 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:

$$|\psi(s)\rangle = \sum_{\kappa} c_\kappa(s) e^{-\imath T \phi_\kappa(s)} |\kappa(s)\rangle \tag{2.7}$$

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

## Potential solution: QA

Consider the $k$-th eigenstate state of the Hamiltonian:

$$H(s)|k\rangle = \lambda_k(s)|k\rangle \tag{2.6}$$

with $|0(0)\rangle$ being the ground state of $H_{\mathrm{init}}$ 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:

$$|\psi(s)\rangle = \sum_{\kappa} c_\kappa(s) e^{-\imath T \phi_\kappa(s)} |\kappa(s)\rangle \tag{2.7}$$

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

It turns out:

$$c_0(s) \approx 1 + \mathcal{O}(T^{-2}), \tag{2.8}$$

$$c_{\kappa\neq 0}(s) \approx \frac{\mathrm{i}}{T}\left[A_\kappa(0) - \mathrm{e}^{\imath T[\phi_\kappa(s)-\phi_0(s)]}A_\kappa(s)\right] + \mathcal{O}(T^{-2}) \tag{2.9}$$

The adiabaticity condition becomes:

$$\frac{1}{\Delta_\kappa(t)^2}\left|\left\langle \kappa(t)\left|\frac{\mathrm{d}H(t)}{\mathrm{d}t}\right|0(t)\right\rangle\right| = \delta \ll 1. \tag{2.10}$$

It turns out:

$$c_0(s) \approx 1 + \mathcal{O}(T^{-2}), \tag{2.8}$$

$$c_{\kappa\neq 0}(s) \approx \frac{\mathrm{i}}{T}\left[A_\kappa(0) - \mathrm{e}^{\imath T[\phi_\kappa(s)-\phi_0(s)]}A_\kappa(s)\right] + \mathcal{O}(T^{-2}) \tag{2.9}$$

The adiabaticity condition becomes:

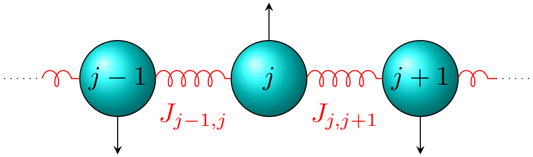
$$\frac{1}{\Delta_\kappa(t)^2}\left|\left\langle \kappa(t)\left|\frac{\mathrm{d}H(t)}{\mathrm{d}t}\right|0(t)\right\rangle\right| = \delta \ll 1. \tag{2.10}$$

## Convergence via Ising model

Suppose that the optimization (2.3) problem we wish to solve can be represented as the ground-state search of an Ising model of general form

$$H_{\mathrm{Ising}} \equiv -\sum_{i=1}^{N} J_i \sigma_i^z - \sum_{i,j=1}^{N} J_{ij}\sigma_i^z \sigma_j^z + \mathcal{O}(\sigma^3). \tag{2.11}$$

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

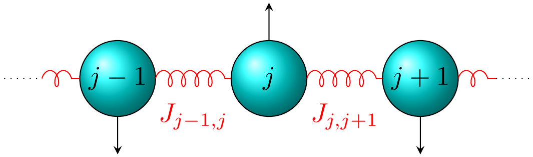


## Convergence via Ising model

Suppose that the optimization (2.3) problem we wish to solve can be represented as the ground-state search of an Ising model of general form

$$H_{\text{Ising}} \equiv -\sum_{i=1}^{N} J_i \sigma_i^z - \sum_{i,j=1}^{N} J_{ij} \sigma_i^z \sigma_j^z + \mathcal{O}(\sigma^3). \tag{2.11}$$

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



## Convergence via Ising model

Eigenvalues of $\sigma_i^z$ is $+1$ or $-1$, which corresponds the classical Ising spin chain.

Most combinatorial optimization problems can be written in this form by mapping binary variables $\{0, 1\}$ to spin variables $\{\pm 1\}$.

An important assumption is that the Hamiltonian (2.11) is proportional to the number of spins $N$ for large $N$.

## Convergence via Ising model

Eigenvalues of $\sigma_i^z$ is $+1$ or $-1$, which corresponds the classical Ising spin chain.

Most combinatorial optimization problems can be written in this form by mapping binary variables $\{0, 1\}$ to spin variables $\{\pm 1\}$.

An important assumption is that the Hamiltonian (2.11) is proportional to the number of spins $N$ for large $N$.

## Convergence via Ising model

Eigenvalues of $\sigma_i^z$ is $+1$ or $-1$, which corresponds the classical Ising spin chain.

Most combinatorial optimization problems can be written in this form by mapping binary variables $\{0, 1\}$ to spin variables $\{\pm 1\}$.

An important assumption is that the Hamiltonian (2.11) 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:

$$H_{\mathrm{TF}}(t) \equiv -\Gamma(t) \sum_{i=1}^{N} \sigma_i^x \qquad (2.12)$$

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

$$H(t) = H_{\text{Ising}} + H_{\mathrm{TF}}(t) \qquad (2.13)$$

is dominated by the second kinetic term.

## Transverse Field

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

$$H_{\mathrm{TF}}(t) \equiv -\Gamma(t) \sum_{i=1}^{N} \sigma_i^x \tag{2.12}$$

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

$$H(t) = H_{\text{Ising}} + H_{\mathrm{TF}}(t) \tag{2.13}$$

is dominated by the second kinetic term.

## Transverse Field

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

$$H_{\mathrm{TF}}(t) \equiv -\Gamma(t) \sum_{i=1}^{N} \sigma_i^x \tag{2.12}$$

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

$$H(t) = H_{\text{Ising}} + H_{\mathrm{TF}}(t) \tag{2.13}$$

is dominated by the second kinetic term.

## Transverse Field

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

$$H_{\mathrm{TF}}(t) \equiv -\Gamma(t) \sum_{i=1}^{N} \sigma_i^x \tag{2.12}$$

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

$$H(t) = H_{\text{Ising}} + H_{\mathrm{TF}}(t) \tag{2.13}$$

is dominated by the second kinetic term.

## 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 $H_{\text{Ising}}$ .

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 (2.13) to the non-trivial ground state of (2.11), 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 evolution of the TF Ising Model

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

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 (2.13) to the non-trivial ground state of (2.11), 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 evolution of the TF Ising Model

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

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 (2.13) to the non-trivial ground state of (2.11), 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 evolution of the TF Ising Model

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

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 (2.13) to the non-trivial ground state of (2.11), 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 evolution of the TF Ising Model

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

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 (2.13) to the non-trivial ground state of (2.11), 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 evolution of the TF Ising Model

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

**Theorem**

*The adiabaticity* (2.10) *for the transverse-field Ising model* (2.11) *yields the time dependence of* $\Gamma(t)$ *as*

$$\Gamma(t) = a(\delta t + c)^{-1/(2N-1)} \tag{2.14}$$

*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* $\delta$ *is a small parameter that controls adiabaticity.*

Point is: **The power decay above satisfies the adiabaticity condition** (2.10) **which guarantees convergence to the ground state of** $H_{\text{Ising}}$ **as** $t \to \infty$.

## The evolution of the TF Ising Model

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

**Theorem**

*The adiabaticity* (2.10) *for the transverse-field Ising model* (2.11) *yields the time dependence of* $\Gamma(t)$ *as*

$$\Gamma(t) = a(\delta t + c)^{-1/(2N-1)} \tag{2.14}$$

*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* $\delta$ *is a small parameter that controls adiabaticity.*

Point is: **The power decay above satisfies the adiabaticity condition** (2.10) **which guarantees convergence to the ground state of** $H_{\text{Ising}}$ **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

$$\min_{x \in \{0,1\}^N} Q(x) \tag{2.15}$$

where the objective function $Q$ is defined as:

$$Q(x) := \sum_{i,j=1}^{N} Q_{ij} x_i x_j + \sum_{i=1}^{N} c_i x_i . \tag{2.16}$$

## 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

$$\min_{x\in\{0,1\}^N} Q(x) \tag{2.15}$$

where the objective function $Q$ is defined as:

$$Q(x) := \sum_{i,j=1}^{N} Q_{ij}x_i x_j + \sum_{i=1}^{N} c_i x_i . \tag{2.16}$$

## 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

$$\min_{x\in\{0,1\}^N} Q(x) \tag{2.15}$$

where the objective function $Q$ is defined as:

$$Q(x) := \sum_{i,j=1}^{N} Q_{ij}x_i x_j + \sum_{i=1}^{N} c_i x_i . \tag{2.16}$$

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.

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.

## $k \geq 3$

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 $w := yz$ and impose the additional constraint

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

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

## $k \geq 3$

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 $w := yz$ and impose the additional constraint

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

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

## $k \geq 3$

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 $w := yz$ and impose the additional constraint

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

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

## $k \geq 3$

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 $w := yz$ and impose the additional constraint

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

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

## Example: 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 $W$.

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} v_i x_i, \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i x_i \leq W, \end{aligned} \tag{2.17}$$

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.

## Example: 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 $W$.

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} v_i x_i, \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i x_i \leq W, \end{aligned} \tag{2.17}$$

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:

$$\lambda_0 \left( \sum_{i=1}^{n} w_i x_i - W + S \right)^2 \tag{2.18}$$

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

# 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:

$$\lambda_0 \left( \sum_{i=1}^{n} w_i x_i - W + S \right)^2 \tag{2.18}$$

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

## The QUBO formulation

the Knapsack problem can be formulated then as:

$$\max \quad \sum_{i}^{n} v_i x_i - \lambda_0 \left( \sum_{i}^{n} w_i x_i - W + \sum_{k=1}^{N} 2^{k-1} s_k \right)^2 , \tag{2.19}$$

## Maping to the Ising model

$$\min \quad -\left(\sum_{i=1}^{n}\sum_{j=1}^{n} J_{ij}s_is_j + \sum_{i=1}^{n} h_is_i + c\right) \tag{2.20}$$

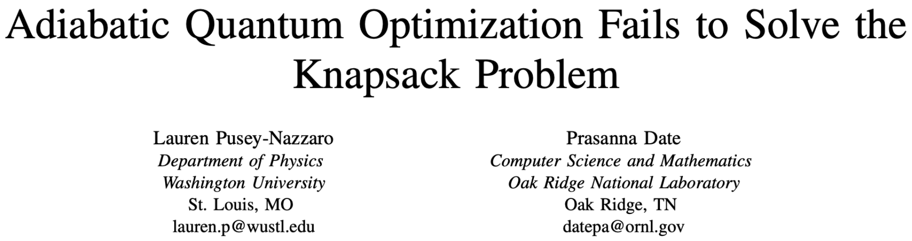
where

$$J_{ij} = \lambda_0 2^{k-1} w_i \delta_{ij}, \tag{2.21}$$

$$h_i = \frac{v_i}{2} - \lambda_0 w_i W, \tag{2.22}$$

$$c = \sum_{i=1}^{n} \frac{v_i}{2} + \lambda_0 \left(\sum_{i=1}^{n} \frac{w_i^2}{4} + \sum_{k=1}^{N} 2^{2k-2}\right). \tag{2.23}$$

## QA Fails?

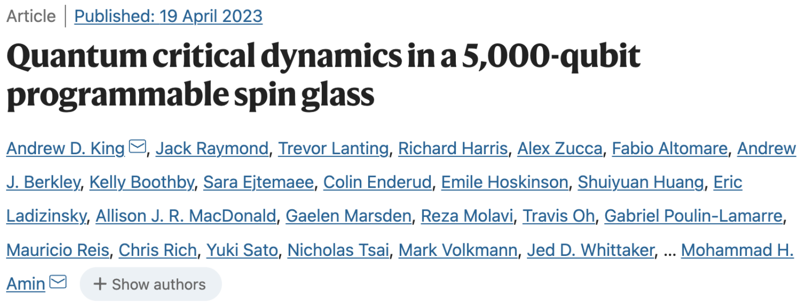


Adiabatic Quantum Optimization Fails to Solve the Knapsack Problem

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

## QA Succeeds?



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

## Break

Questions?

## VQAs: 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**

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):

$$\min_{x \in \{0,1\}^n} \quad f(x). \tag{2.24}$$

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

## VQAs: 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**

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):

$$\min_{x \in \{0,1\}^n} \quad f(x). \tag{2.24}$$

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

## VQAs: 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**

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):

$$\min_{x \in \{0,1\}^n} \quad f(x). \tag{2.24}$$

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

## VQAs: 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**

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):

$$\min_{x \in \{0,1\}^n} f(x). \tag{2.24}$$

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

## VQAs: 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**

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):

$$\min_{x \in \{0,1\}^n} \quad f(x). \tag{2.24}$$

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

## VQAs: 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**

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 $U(\vartheta)$.

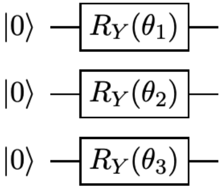
In practice such a quantum circuit is a sequence of universal quantum gates' compositions and/or tensor products.

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

$$\min_{x\in\{0,1\}^n} \quad f(x). \tag{2.24}$$

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

## PQC example



## PQC example

$$U(\theta) = R_Y(\theta_1) \otimes R_Y(\theta_2) \otimes R_Y(\theta_3)$$

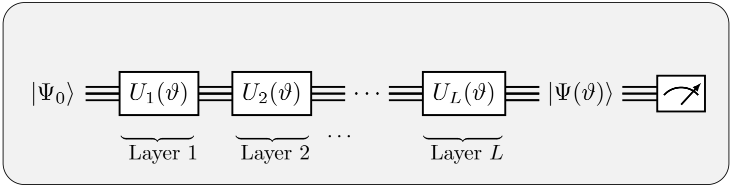
$$= \begin{pmatrix} \cos\frac{\theta_1}{2} & -\sin\frac{\theta_1}{2} \\ \sin\frac{\theta_1}{2} & \cos\frac{\theta_1}{2} \end{pmatrix} \otimes \begin{pmatrix} \cos\frac{\theta_2}{2} & -\sin\frac{\theta_2}{2} \\ \sin\frac{\theta_2}{2} & \cos\frac{\theta_2}{2} \end{pmatrix} \otimes \begin{pmatrix} \cos\frac{\theta_3}{2} & -\sin\frac{\theta_3}{2} \\ \sin\frac{\theta_3}{2} & \cos\frac{\theta_3}{2} \end{pmatrix}$$

$$= \begin{pmatrix}
c_1c_2c_3 & -c_1c_2s_3 & -c_1s_2c_3 & c_1s_2s_3 & -s_1c_2c_3 & s_1c_2s_3 & s_1s_2c_3 & -s_1s_2s_3 \\
c_1c_2s_3 & c_1c_2c_3 & -c_1s_2s_3 & -c_1s_2c_3 & -s_1c_2s_3 & -s_1c_2c_3 & s_1s_2s_3 & s_1s_2c_3 \\
c_1s_2c_3 & -c_1s_2s_3 & c_1c_2c_3 & -c_1c_2s_3 & -s_1s_2c_3 & s_1s_2s_3 & -s_1c_2c_3 & s_1c_2s_3 \\
c_1s_2s_3 & c_1s_2c_3 & c_1c_2s_3 & c_1c_2c_3 & -s_1s_2s_3 & -s_1s_2c_3 & -s_1c_2s_3 & -s_1c_2c_3 \\
s_1c_2c_3 & -s_1c_2s_3 & -s_1s_2c_3 & s_1s_2s_3 & c_1c_2c_3 & -c_1c_2s_3 & -c_1s_2c_3 & c_1s_2s_3 \\
s_1c_2s_3 & s_1c_2c_3 & -s_1s_2s_3 & -s_1s_2c_3 & c_1c_2s_3 & c_1c_2c_3 & -c_1s_2s_3 & -c_1s_2c_3 \\
s_1s_2c_3 & -s_1s_2s_3 & s_1c_2c_3 & -s_1c_2s_3 & c_1s_2c_3 & -c_1s_2s_3 & c_1c_2c_3 & -c_1c_2s_3 \\
s_1s_2s_3 & s_1s_2c_3 & s_1c_2s_3 & s_1c_2c_3 & c_1s_2s_3 & c_1s_2c_3 & c_1c_2s_3 & c_1c_2c_3
\end{pmatrix}$$

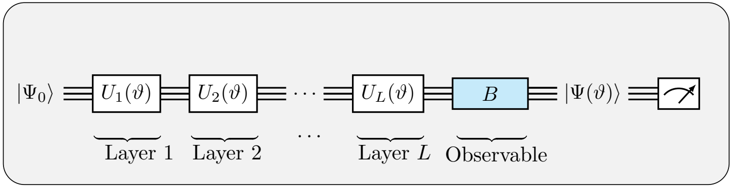
for $\theta = (\theta_1, \theta_2, \theta_3) \in \mathbb{R}^3$, where $c_i = \cos\frac{\theta_i}{2}$ and $s_i = \sin\frac{\theta_i}{2}$ for $i = 1, 2, 3$.

## Generically

The quantum part of a VQA has the following form:

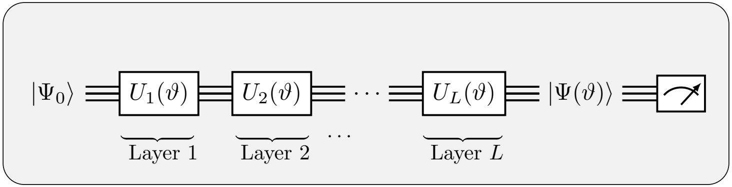


More precisely, we can explicitly include the observable we want to measure:

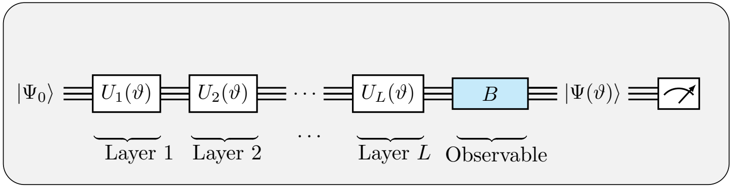


## Generically

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

$$B(\vartheta) = f\Big(\{|\Psi\rangle_0\}, \{B_k\}, U(\vartheta)\Big). \tag{2.25}$$

It involves (some) obsevable quantity: operators $\{O_k\}$ given input states $\{|\Psi\rangle_0\}$ and the PQC $U(\vartheta)$.

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

$$B(\vartheta) = \sum_{k\in I} \mathrm{Tr}\Big(B_k U(\vartheta)\rho_{\mathrm{in}} U^\dagger(\vartheta)\Big), \tag{2.26}$$

or more generically

$$B(\vartheta) = \sum_{k\in I} f_k\left(\mathrm{Tr}\Big(B_k U(\vartheta)\rho_{\mathrm{in}} U^\dagger(\vartheta)\Big)\right), \tag{2.27}$$

for some functions $f_k$.

## VQAs: The Quantum Part

Given a PQC with $\vartheta \in \mathbb{R}^L$ we can define a cost function

$$B(\vartheta) = f\Big(\{|\Psi\rangle_0\}, \{B_k\}, U(\vartheta)\Big). \tag{2.25}$$

It involves (some) obsevable quantity: operators $\{O_k\}$ given input states $\{|\Psi\rangle_0\}$ and the PQC $U(\vartheta)$.

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

$$B(\vartheta) = \sum_{k \in I} \mathrm{Tr}\Big(B_k U(\vartheta)\rho_{\text{in}} U^\dagger(\vartheta)\Big), \tag{2.26}$$

or more generically

$$B(\vartheta) = \sum_{k \in I} f_k\left(\mathrm{Tr}\Big(B_k U(\vartheta)\rho_{\text{in}} U^\dagger(\vartheta)\Big)\right), \tag{2.27}$$

for some functions $f_k$.

## VQAs: The Quantum Part

Given a PQC with $\vartheta \in \mathbb{R}^L$ we can define a cost function

$$B(\vartheta) = f\Big(\{|\Psi\rangle_0\}, \{B_k\}, U(\vartheta)\Big). \tag{2.25}$$

It involves (some) obsevable quantity: operators $\{O_k\}$ given input states $\{|\Psi\rangle_0\}$ and the PQC $U(\vartheta)$.

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

$$B(\vartheta) = \sum_{k\in I} \mathrm{Tr}\Big(B_k U(\vartheta)\rho_{\text{in}} U^\dagger(\vartheta)\Big), \tag{2.26}$$

or more generically

$$B(\vartheta) = \sum_{k\in I} f_k\left(\mathrm{Tr}\Big(B_k U(\vartheta)\rho_{\text{in}} U^\dagger(\vartheta)\Big)\right), \tag{2.27}$$

for some functions $f_k$.

## VQAs: The Quantum Part

Given a PQC with $\vartheta \in \mathbb{R}^L$ we can define a cost function

$$B(\vartheta) = f\Big(\{|\Psi\rangle_0\}, \{B_k\}, U(\vartheta)\Big). \tag{2.25}$$

It involves (some) obsevable quantity: operators $\{O_k\}$ given input states $\{|\Psi\rangle_0\}$ and the PQC $U(\vartheta)$.

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

$$B(\vartheta) = \sum_{k \in I} \mathrm{Tr}\Big(B_k U(\vartheta) \rho_{\text{in}} U^\dagger(\vartheta)\Big), \tag{2.26}$$

or more generically

$$B(\vartheta) = \sum_{k \in I} f_k\left(\mathrm{Tr}\Big(B_k U(\vartheta) \rho_{\text{in}} U^\dagger(\vartheta)\Big)\right), \tag{2.27}$$

for some functions $f_k$.

## VQAs: Measurements

Construct an empirical estimate of $\langle B \rangle_\vartheta$ of the observable:

$$\langle B \rangle_\vartheta := \langle \Psi(\vartheta) | B | \Psi(\vartheta) \rangle \,, \tag{2.28}$$

where $|\Psi(\vartheta)\rangle := U(\vartheta)|\Psi_0\rangle$.

The empirical estimate we measure is:$\mathbb{E}[B_\vartheta]$. This is constructed by measuring the same circuit repeatedly. Out of this we construct a cost function we would like to minimize:

$$\vartheta^* := \arg\min_\vartheta \|\mathbb{E}[B_\vartheta] - \langle B \rangle_\vartheta \|_\ell^p \tag{2.29}$$

## VQAs: Measurements

Construct an empirical estimate of $\langle B \rangle_\vartheta$ of the observable:

$$\langle B \rangle_\vartheta := \langle \Psi(\vartheta) | B | \Psi(\vartheta) \rangle \,, \tag{2.28}$$

where $|\Psi(\vartheta)\rangle := U(\vartheta)|\Psi_0\rangle$.

The empirical estimate we measure is:$\mathbb{E}[B_\vartheta]$. This is constructed by measuring the same circuit repeatedly. Out of this we construct a cost function we would like to minimize:

$$\vartheta^* := \arg\min_{\vartheta} \|\mathbb{E}[B_\vartheta] - \langle B \rangle_\vartheta \|_\ell^p \tag{2.29}$$

## VQAs: Measurements

Construct an empirical estimate of $\langle B \rangle_\vartheta$ of the observable:

$$\langle B \rangle_\vartheta := \langle \Psi(\vartheta) | B | \Psi(\vartheta) \rangle \,, \tag{2.28}$$

where $|\Psi(\vartheta)\rangle := U(\vartheta)|\Psi_0\rangle$.

The empirical estimate we measure is:$\mathbb{E}[B_\vartheta]$. This is constructed by measuring the same circuit repeatedly. Out of this we construct a cost function we would like to minimize:

$$\vartheta^* := \arg\min_\vartheta \|\mathbb{E}[B_\vartheta] - \langle B \rangle_\vartheta\|_\ell^p \tag{2.29}$$

## VQAs: Measurements

Construct an empirical estimate of $\langle B \rangle_\vartheta$ of the observable:

$$\langle B \rangle_\vartheta := \langle \Psi(\vartheta) | B | \Psi(\vartheta) \rangle \,, \tag{2.28}$$

where $|\Psi(\vartheta)\rangle := U(\vartheta)|\Psi_0\rangle$.

The empirical estimate we measure is:$\mathbb{E}[B_\vartheta]$. This is constructed by measuring the same circuit repeatedly. Out of this we construct a cost function we would like to minimize:

$$\vartheta^* := \arg\min_\vartheta \| \mathbb{E}[B_\vartheta] - \langle B \rangle_\vartheta \|_\ell^p \tag{2.29}$$

## VQAs: Measurements

Construct an empirical estimate of $\langle B \rangle_\vartheta$ of the observable:

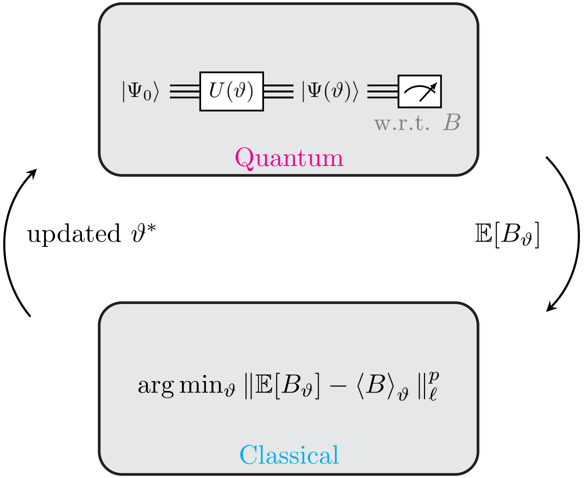
$$\langle B \rangle_\vartheta := \langle \Psi(\vartheta) | B | \Psi(\vartheta) \rangle \,, \tag{2.28}$$

where $|\Psi(\vartheta)\rangle := U(\vartheta)|\Psi_0\rangle$.

The empirical estimate we measure is:$\mathbb{E}[B_\vartheta]$. This is constructed by measuring the same circuit repeatedly. Out of this we construct a cost function we would like to minimize:

$$\vartheta^* := \arg\min_\vartheta \|\mathbb{E}[B_\vartheta] - \langle B \rangle_\vartheta \|_\ell^p \tag{2.29}$$

## VQAs: After the measurement what?


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

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.

## 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 $\vartheta$.

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.

## 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 $\vartheta$.

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.

## 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 $\vartheta$.

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: compute gradients

Consider a cost function as in Eq. (2.27):

$$B(\vartheta) = \mathrm{Tr}\Big(BU(\vartheta)\rho_{\mathrm{in}}U^\dagger(\vartheta)\Big), \tag{2.30}$$

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

$$U(\vartheta_j) = e^{\imath\vartheta_j\sigma_j^a}. \tag{2.31}$$

Then:

$$\frac{\partial B(\vartheta)}{\partial\vartheta_j} \sim \frac{1}{\sin\alpha}\Big(\mathrm{Tr}\big(BU^\dagger(\vartheta_+)\rho U(\vartheta_+)\big) - \mathrm{Tr}\big(BU^\dagger(\vartheta_-)\rho U(\vartheta_-)\big)\Big) \tag{2.32}$$

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 $\alpha$.

## Parameter Shift Rule: compute gradients

Consider a cost function as in Eq. (2.27):

$$B(\vartheta) = \mathrm{Tr}\Big(BU(\vartheta)\rho_{\mathrm{in}}U^\dagger(\vartheta)\Big), \tag{2.30}$$

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

$$U(\vartheta_j) = e^{\imath\vartheta_j\sigma_j^a}. \tag{2.31}$$

Then:

$$\frac{\partial B(\vartheta)}{\partial \vartheta_j} \sim \frac{1}{\sin\alpha}\big(\mathrm{Tr}(BU^\dagger(\vartheta_+)\rho U(\vartheta_+)) - \mathrm{Tr}(BU^\dagger(\vartheta_-)\rho U(\vartheta_-))\big) \tag{2.32}$$

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 $\alpha$.

## Parameter Shift Rule: compute gradients

Consider a cost function as in Eq. (2.27):

$$B(\vartheta) = \mathrm{Tr}\Big(BU(\vartheta)\rho_{\mathrm{in}}U^\dagger(\vartheta)\Big), \tag{2.30}$$

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

$$U(\vartheta_j) = e^{\imath\vartheta_j\sigma_j^a}. \tag{2.31}$$

Then:

$$\frac{\partial B(\vartheta)}{\partial\vartheta_j} \sim \frac{1}{\sin\alpha}\left(\mathrm{Tr}\left(BU^\dagger(\vartheta_+)\rho U(\vartheta_+)\right) - \mathrm{Tr}\left(BU^\dagger(\vartheta_-)\rho U(\vartheta_-)\right)\right) \tag{2.32}$$

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 $\alpha$.

## Parameter Shift Rule: compute gradients

Consider a cost function as in Eq. (2.27):

$$B(\vartheta) = \mathrm{Tr}\Big(BU(\vartheta)\rho_{\mathrm{in}}U^\dagger(\vartheta)\Big), \tag{2.30}$$

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

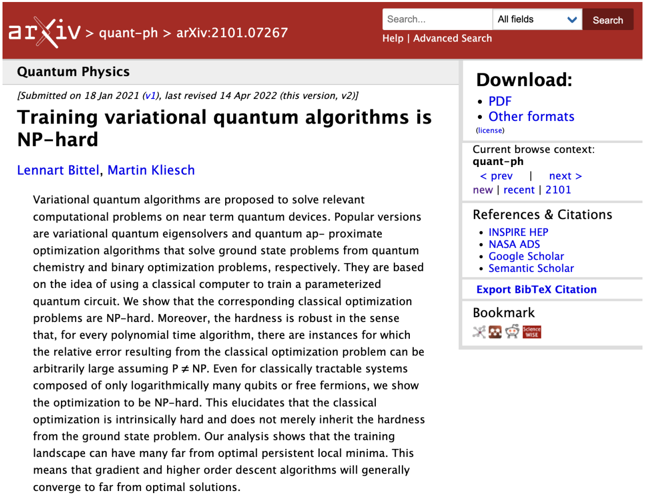
$$U(\vartheta_j) = e^{\imath\vartheta_j\sigma_j^a}. \tag{2.31}$$

Then:

$$\frac{\partial B(\vartheta)}{\partial\vartheta_j} \sim \frac{1}{\sin\alpha}\left(\mathrm{Tr}\left(BU^\dagger(\vartheta_+)\rho U(\vartheta_+)\right) - \mathrm{Tr}\left(BU^\dagger(\vartheta_-)\rho U(\vartheta_-)\right)\right) \tag{2.32}$$

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 $\alpha$.

# It's hard to train VQAs



arXiv > quant-ph > arXiv:2101.07267

Search... All fields Search

Help | Advanced Search

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.

**Download:**

- PDF
- Other formats

(license)

Current browse context:
**quant-ph**
< prev | next >
new | recent | 2101

References & Citations

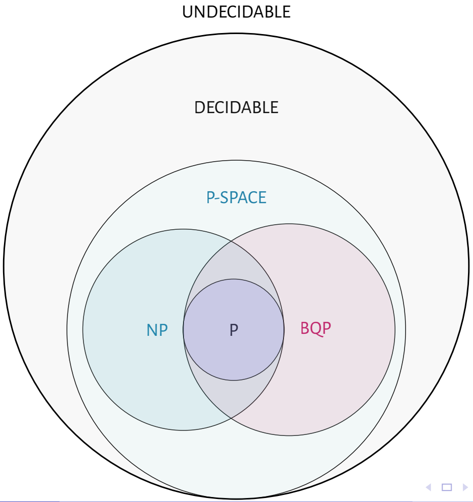
- INSPIRE HEP
- NASA ADS
- Google Scholar
- Semantic Scholar

**Export BibTeX Citation**

Bookmark

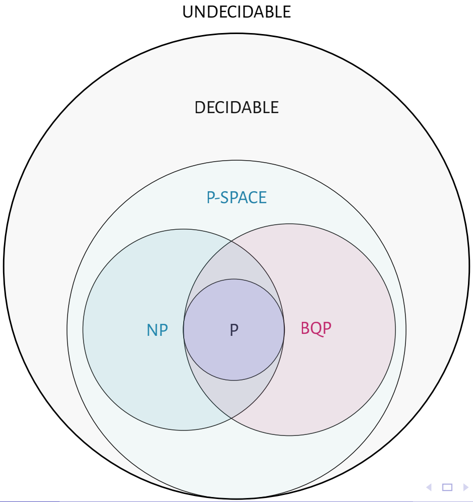
## Undecidability conjecture

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



## 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).

## 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).

## 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).

## 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).

## Break

Questions?

## 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.

## 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.

## 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:

$$H(s) = (1-s)H_{\mathrm{init}} + sH_{\mathrm{final}}. \tag{3.1}$$

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

$$U(T) \sim \exp\left(-\imath \int_0^t H(w)\mathrm{d}w\right). \tag{3.2}$$

Solution: Trotterization. Discretize $U(T) \equiv U(T,0)$ into intervals $\Delta t$ (in total $T = L\Delta t$) small enough that the Hamiltonian is approximately constant over each interval. Then:

$$U(T,0) = U(T, T-\Delta t)U(T-\Delta t, T-2\Delta t)\dots U(\Delta t, 0) \tag{3.3}$$

$$= \prod_{j=0}^{L-1} U((L-j)\Delta T, (L-j-1)\Delta t) \tag{3.4}$$

$$=_{L\to\infty} \prod^{L-1} e^{-\imath H[(L-j)\Delta t]\Delta t} \tag{3.5}$$

## Trotterization

Recall that in the case of AQC we have:

$$H(s) = (1-s)H_{\text{init}} + sH_{\text{final}}. \tag{3.1}$$

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

$$U(T) \sim \exp\left(-\imath \int_0^t H(w)\mathrm{d}w\right). \tag{3.2}$$

Solution:Trotterization. Discretize $U(T) \equiv U(T,0)$ into intervals $\Delta t$ (in total $T = L\Delta t$) small enough that the Hamiltonian is approximately constant over each interval. Then:

$$\begin{aligned} U(T,0) &= U(T, T-\Delta t)U(T-\Delta t, T-2\Delta t)\dots U(\Delta t, 0) && (3.3)\\ &= \prod_{j=0}^{L-1} U((L-j)\Delta T, (L-j-1)\Delta t) && (3.4)\\ &=_{L\to\infty} \prod_{j=0}^{L-1} e^{-\imath H[(L-j)\Delta t]\Delta t} && (3.5)\end{aligned}$$

## Trotterization

Recall that in the case of AQC we have:

$$H(s) = (1-s)H_{\text{init}} + sH_{\text{final}}. \tag{3.1}$$

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

$$U(T) \sim \exp\left(-\imath \int_0^t H(w)\mathrm{d}w\right). \tag{3.2}$$

Solution:Trotterization. Discretize $U(T) \equiv U(T,0)$ into intervals $\Delta t$ (in total $T = L\Delta t$) small enough that the Hamiltonian is approximately constant over each interval. Then:

$$U(T,0) = U(T, T-\Delta t)U(T-\Delta t, T-2\Delta t)\dots U(\Delta t, 0) \tag{3.3}$$

$$= \prod_{j=0}^{L-1} U((L-j)\Delta T, (L-j-1)\Delta t) \tag{3.4}$$

$$=_{L\to\infty} \prod^{L-1} e^{-\imath H[(L-j)\Delta t]\Delta t} \tag{3.5}$$

## Trotterization

Recall that in the case of AQC we have:

$$H(s) = (1-s)H_{\text{init}} + sH_{\text{final}}. \tag{3.1}$$

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

$$U(T) \sim \exp\left(-\imath \int_0^t H(w)\mathrm{d}w\right). \tag{3.2}$$

Solution:Trotterization. Discretize $U(T) \equiv U(T,0)$ into intervals $\Delta t$ (in total $T = L\Delta t$) small enough that the Hamiltonian is approximately constant over each interval. Then:

$$\begin{aligned} U(T,0) &= U(T, T-\Delta t)U(T-\Delta t, T-2\Delta t)\ldots U(\Delta t, 0) && (3.3)\\ &= \prod_{j=0}^{L-1} U((L-j)\Delta T, (L-j-1)\Delta t) && (3.4)\\ &=_{L\to\infty} \prod^{L-1} e^{-\imath H[(L-j)\Delta t]\Delta t} && (3.5)\end{aligned}$$

Using the identity

$$e^{\imath(A+B)x} = e^{\imath(A)x} e^{\imath(B)x} + \mathcal{O}(x^2) \tag{3.6}$$

we deduce that

$$U(T,0) \approx \prod_{j=1}^{p} e^{\{-\imath(1-s(j\Delta t))H_{\mathrm{init}}\Delta t\}} e^{\{-\imath s(j\Delta t)H_{\mathrm{final}}\Delta t\}}. \tag{3.7}$$

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

Using the identity

$$e^{\imath(A+B)x} = e^{\imath(A)x}e^{\imath(B)x} + \mathcal{O}(x^2) \tag{3.6}$$

we deduce that

$$U(T,0) \approx \prod_{j=1}^{p} e^{\{-\imath(1-s(j\Delta t))H_{\mathrm{init}}\Delta t\}} e^{\{-\imath s(j\Delta t)H_{\mathrm{final}}\Delta t\}}. \tag{3.7}$$

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

Using the identity

$$e^{\imath(A+B)x} = e^{\imath(A)x}e^{\imath(B)x} + \mathcal{O}(x^2) \tag{3.6}$$

we deduce that

$$U(T,0) \approx \prod_{j=1}^{p} e^{\{-\imath(1-s(j\Delta t))H_{\text{init}}\Delta t\}} e^{\{-\imath s(j\Delta t)H_{\text{final}}\Delta t\}}. \tag{3.7}$$

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

## 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 $C_\alpha$, each of which takes the form

$$C_\alpha(z) = \begin{cases} 1 \text{ if } z \text{ satisfies the constraint} \\ 0 \text{ otherwise.} \end{cases} \tag{3.8}$$

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

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z) \tag{3.9}$$

## 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 $C_\alpha$, each of which takes the form

$$C_\alpha(z) = \begin{cases} 1 \text{ if } z \text{ satisfies the constraint} \\ 0 \text{ otherwise.} \end{cases} \tag{3.8}$$

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

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z) \tag{3.9}$$

## 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:

$$H_C|z\rangle = C(z)|z\rangle \tag{3.10}$$

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 $C(z)$.

## 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:

$$H_C|z\rangle = C(z)|z\rangle \tag{3.10}$$

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 $C(z)$.

## 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:

$$H_C|z\rangle = C(z)|z\rangle \tag{3.10}$$

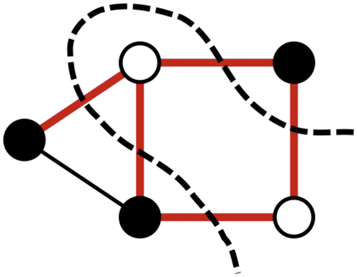
and since $C(z)$ is scalar valued, we can see that each $|z\rangle$ is an eigenstate of $H_C$.

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:

$$C(z) = \frac{1}{2} \sum_{(i,j) \in E(G)} z_i z_j \tag{3.11}$$



## 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. (3.11) you should suspect its the Ising Hamiltonian.

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

$$H_B = \sum_{j=1}^{n} \sigma_j^x \tag{3.12}$$

The corresponding unitaries we need are:

$$U_C = e^{-I\gamma H_C} \tag{3.13}$$

$$U_B = e^{-\imath\beta H_B} \tag{3.14}$$

## 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. (3.11) you should suspect its the Ising Hamiltonian.

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

$$H_B = \sum_{j=1}^{n} \sigma_j^x \tag{3.12}$$

The corresponding unitaries we need are:

$$U_C = e^{-I\gamma H_C} \tag{3.13}$$

$$U_B = e^{-\imath\beta H_B} \tag{3.14}$$

## 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. (3.11) you should suspect its the Ising Hamiltonian.

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

$$H_B = \sum_{j=1}^{n} \sigma_j^x \tag{3.12}$$

The corresponding unitaries we need are:

$$U_C = e^{-I\gamma H_C} \tag{3.13}$$

$$U_B = e^{-\imath\beta H_B} \tag{3.14}$$

## QAOA: Optimization

The goal is to maximize the expression

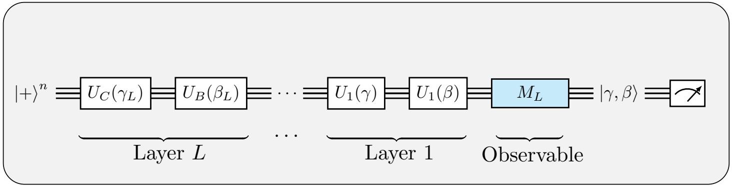
$$M_L(\gamma, \beta) := \langle \gamma, \beta | M_L | \gamma, \beta \rangle \tag{3.15}$$

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

and

$$|\gamma, \beta\rangle = U_C(\gamma_L) U_B(\beta_L) \dots U_C(\gamma_1) U_B(\beta_1) |+\rangle^n. \tag{3.16}$$

Compare with Eq. (3.3). Its basically the same.



## QAOA: Optimization

The goal is to maximize the expression

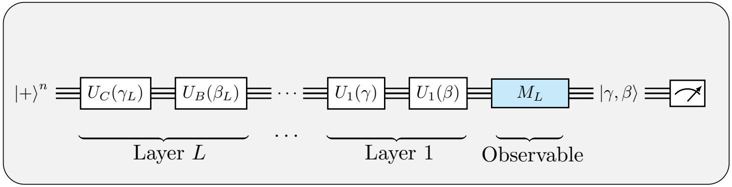
$$M_L(\gamma, \beta) := \langle \gamma, \beta | M_L | \gamma, \beta \rangle \tag{3.15}$$

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

and

$$|\gamma, \beta\rangle = U_C(\gamma_L) U_B(\beta_L) \dots U_C(\gamma_1) U_B(\beta_1) |+\rangle^n. \tag{3.16}$$

Compare with Eq. (3.3). Its basically the same.



## QAOA: Optimization

The goal is to maximize the expression

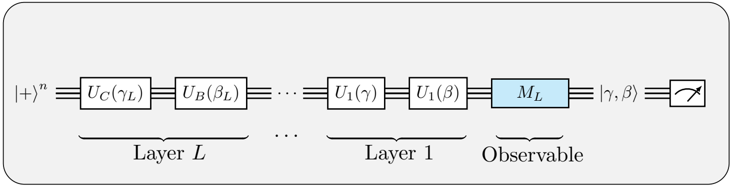
$$M_L(\gamma, \beta) := \langle \gamma, \beta | M_L | \gamma, \beta \rangle \tag{3.15}$$

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

and

$$|\gamma, \beta\rangle = U_C(\gamma_L) U_B(\beta_L) \dots U_C(\gamma_1) U_B(\beta_1) |+\rangle^n. \tag{3.16}$$

Compare with Eq. (3.3). 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

$$(\gamma^*, \beta^*) = \arg\max_{\gamma,\beta} \left\| \mathbb{E}[M_L] - \langle M_L \rangle \right\|_\ell^p \tag{3.17}$$

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.

## 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

$$(\gamma^*, \beta^*) = \arg\max_{\gamma,\beta} \|\mathbb{E}[M_L] - \langle M_L \rangle \|_\ell^p \tag{3.17}$$

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.

## 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

$$(\gamma^*, \beta^*) = \arg\max_{\gamma,\beta} \|\mathbb{E}[M_L] - \langle M_L \rangle \|_\ell^p \tag{3.17}$$

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.

## 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

$$(\gamma^*, \beta^*) = \arg\max_{\gamma,\beta} \|\mathbb{E}[M_L] - \langle M_L \rangle \|_\ell^p \tag{3.17}$$

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.

## 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

$$(\gamma^*, \beta^*) = \arg\max_{\gamma,\beta} \| \mathbb{E}[M_L] - \langle M_L \rangle \|_\ell^p \tag{3.17}$$

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.

## 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

$$(\gamma^*, \beta^*) = \arg\max_{\gamma,\beta} \|\mathbb{E}[M_L] - \langle M_L \rangle\|_\ell^p \tag{3.17}$$

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.