

Lecture 5

Quantum algorithms beyond the circuit model
Quantum optimization - Quantum simulation

Basic concepts of computer science

The complexity hierarchy of decision problems

→ Decision problems have a yes/no answer

Complexity: scaling of resources (for a deterministic Turing machine)

Important classes

P: Problem solved in polynomial time

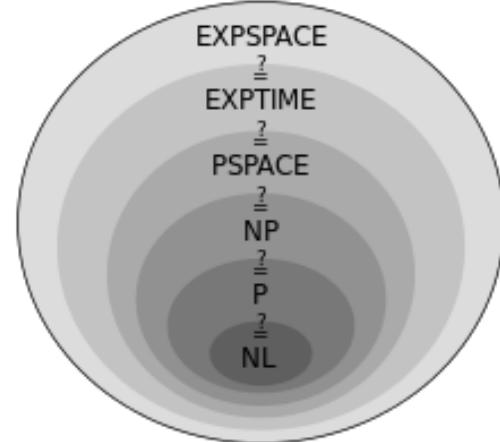
NP: A yes answer can be verified in polynomial time

PSPACE: Problem solved with polynomial resources

EXPTIME: Problem solved in exponential time

NP-HARD: Every problem in NP can be transformed into this problem in polynomial time

NP-COMPLETE: A problem that is both NP and NP-HARD



Today's lecture: Trying to solve some NP-complete problems via 'quantum optimization'

Quantum adiabatic theorem

"A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum" Born-Fock (1928)

Schrodinger equation:

$$i\hbar \frac{d\psi(t)}{dt} = H(t)\psi(t)$$

Instantaneous eigenstates:

$$H(t)\psi_n(t) = E_n(t)\psi_n(t)$$

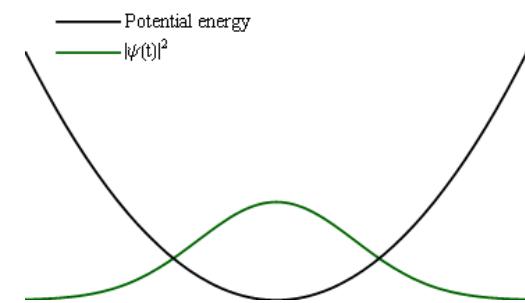
Initial condition:

$$\psi(t=0) = \psi_{n_0}$$

If evolution time **slow enough**
(condition, see later)

$$\psi(t) \approx e^{i\theta(t)}\psi_{n_0}(t)$$

Example:
Particle in an Harmonic trap



Consequence, we can prepare arbitrary ground states via slow time evolution, i.e quantum annealing

Quantum annealing (B. Apolloni, N. Cesa Bianchi and D. De Falco (1988))

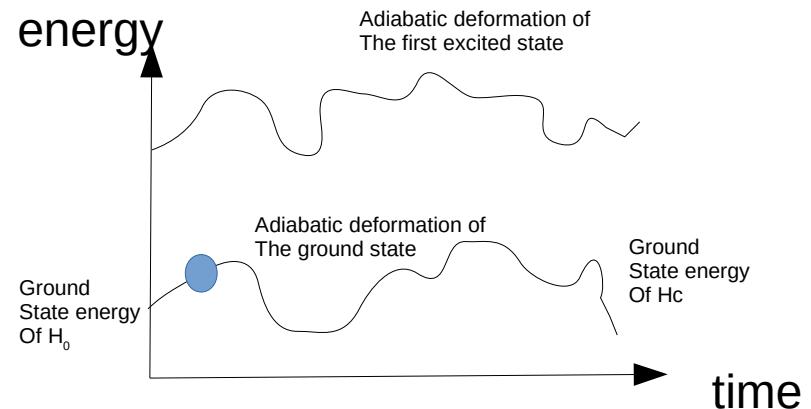
Basic Idea:

- 1) **Encode** the solution of a ‘hard’ computational problem as the ground state of a **classical Hamiltonian H_c** (bitstrings are eigenstates of H_c)
- 2) Initialize the system in the ground state of a trivial quantum Hamiltonian H_0
- 3) ‘Slowly’ change the Hamiltonian towards H_c . Measure the System

$$H(t) = f(t)H_0 + g(t)H_c$$

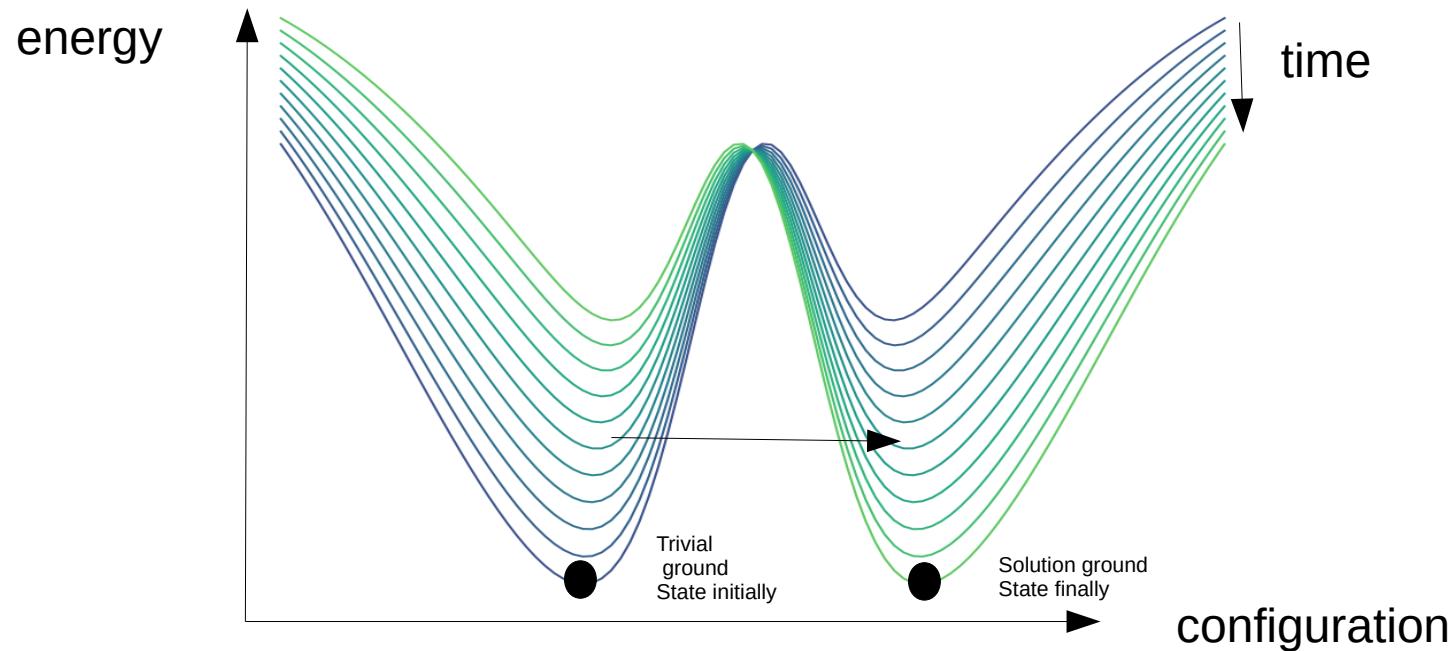
$$f(t = 0) = g(t = t_f) = 1$$

$$f(t = t_f) = g(t = 0) = 0$$



Quantum annealing (B. Apolloni, N. Cesa Bianchi and D. De Falco (1988))

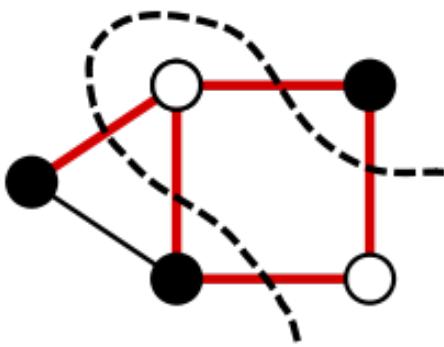
NB: quantum annealing is the quantum analog of simulated annealing



Quantum annealing exploits quantum tunneling

Quantum annealing (B. Apolloni, N. Cesa Bianchi and D. De Falco (1988))

Example: Max-Cut



Problem:

Given a graph, divide the nodes in two parts (white versus black, i.e bit 1 versus bit 0), such as the number of edges ('cuts') between the two parts is maximal

Direct Applications: circuit chip design, data clustering, spin glass physics

Ex:

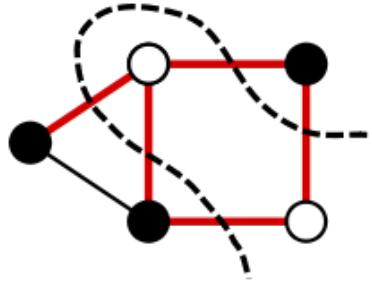
Max-Cut = 5

Complexity : The corresponding decision problem is NP-Complete

Quantum annealing with Max-Cut

Step 1: Encoding in terms of a Ising model

$$H_C = \sum_{(i,j) \in G} \sigma_i^z \sigma_j^z$$



For a given bitstring s (ex 0001001...), s is an eigenstate and

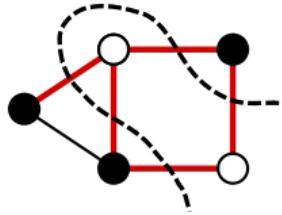
$$\langle s | H_C | s \rangle = \sum_{i,j \in G} Z(s_i)Z(s_j) \quad Z(s_i)Z(s_j) = 1 \ (-1) \quad s_i = s_j \ (s_i \neq s_j)$$

Therefore $H_C = \#[\text{nocuts}] - \#[\text{cuts}] = \text{Constant} - 2\#\text{[cuts]}$

Therefore the ground state s_{sol} is the solution of Max-Cut

This type of mapping extends to many NP-Complete problems (<https://arxiv.org/abs/1302.5843>)

Quantum annealing with Max-Cut



Step 1: Encoding in terms of a Ising model

$$H_C = \sum_{(i,j) \in G} \sigma_i^z \sigma_j^z$$

Step 2: Adiabatic state preparation

$$H(t) = \left(\frac{t}{\tau}\right) H_C + \left(1 - \frac{t}{\tau}\right) \sum_i \sigma_i^x$$

$$\begin{aligned} \psi(t=0) &= |\downarrow_x, \dots, \downarrow_x\rangle \\ |\downarrow_x\rangle &= |\uparrow\rangle - |\downarrow\rangle \end{aligned}$$

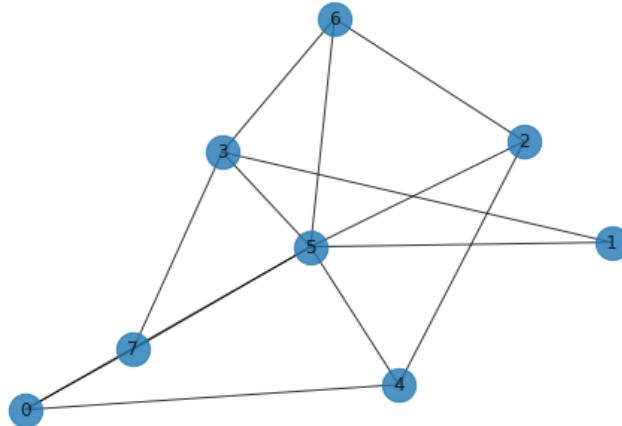
For an adiabatic ramp $\psi(\tau) \approx \psi_c$

Step 3: Readout

Quantum annealing with Max-Cut

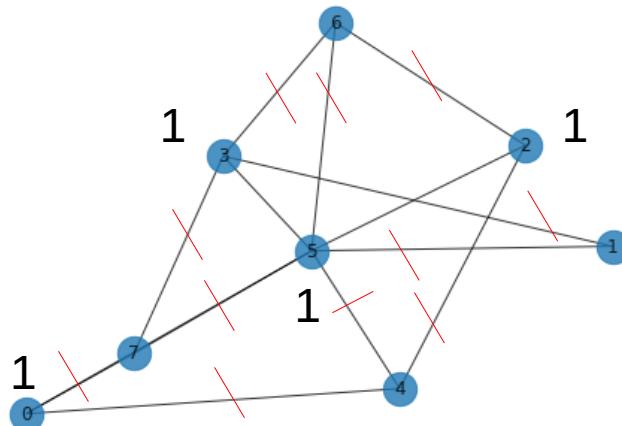
Numerical illustration with QisKit (IBMQ Practical 5)

Step 1 load a graph



Step 2 brute force solution (for benchmarking)

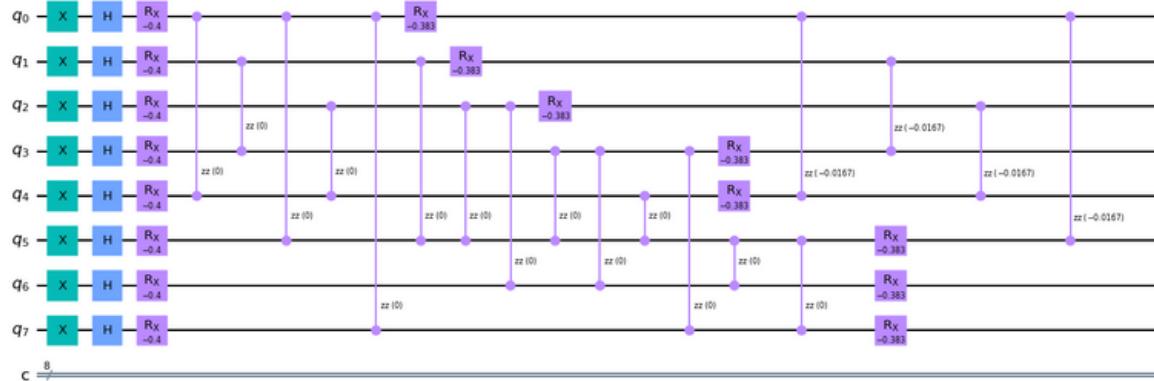
MaxCut: 11
Winning Graphs [(0, 2, 3, 5),



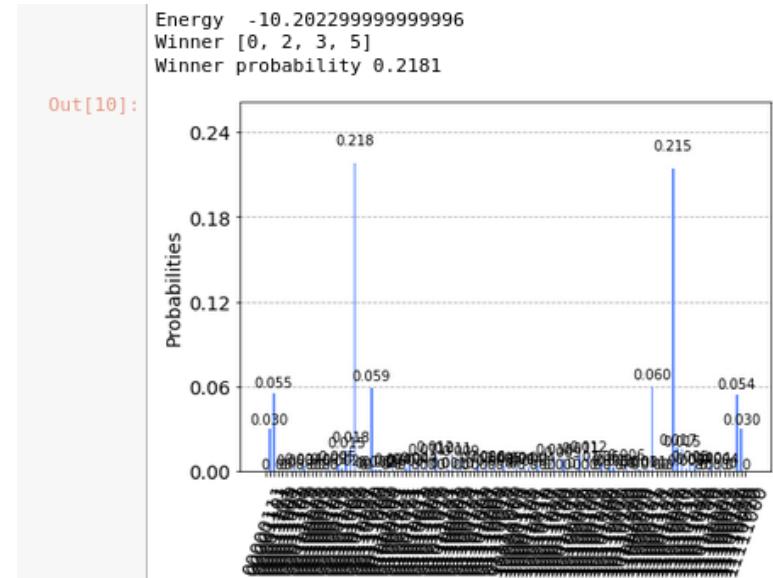
Quantum annealing with Max-Cut

Numerical illustration with QisKit (IBMQ Practical 4)

Step 3 Write a quantum annealing circuit based on the digital quantum simulation algorithm



Step 4 Measure the solution



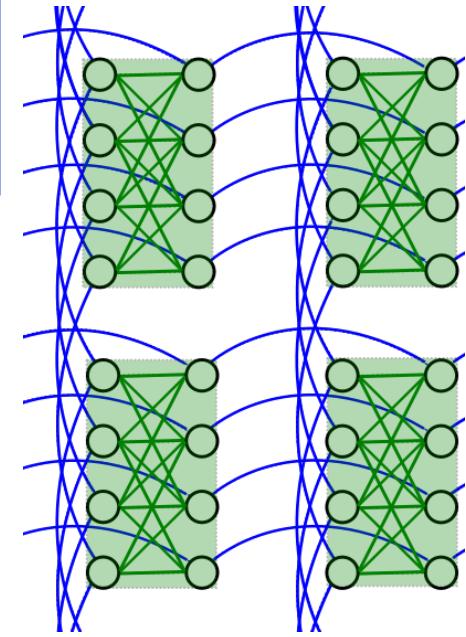
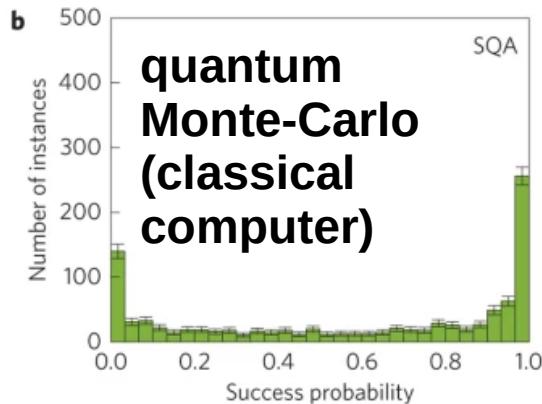
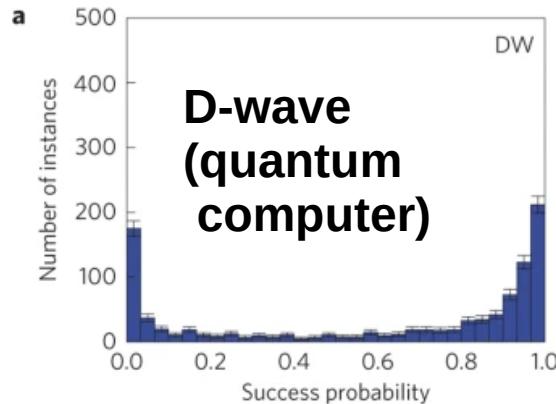
First steps towards quantum annealing

2011: D-wave quantum computing releases D-wave one with 128 qubits!



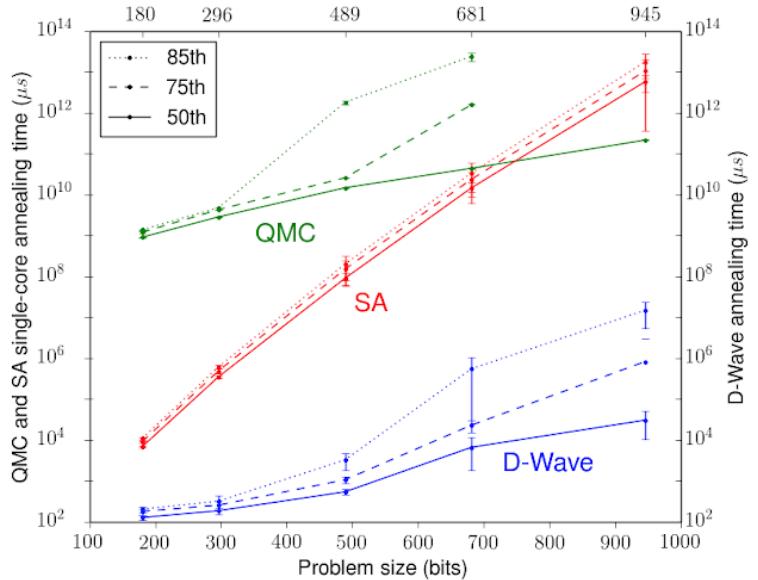
2014: Troyer and co-workers (ETH) show numerical evidence of quantum annealing with 108 qubits
But.. can also 'simulate classically' the whole process via quantum Monte Carlo simulations at finite temperature..

Nature Physics volume 10, pages 218–224 (2014)



First steps towards quantum annealing

<http://arxiv.org/abs/1512.02206> → Google responds with D-wave 2X...

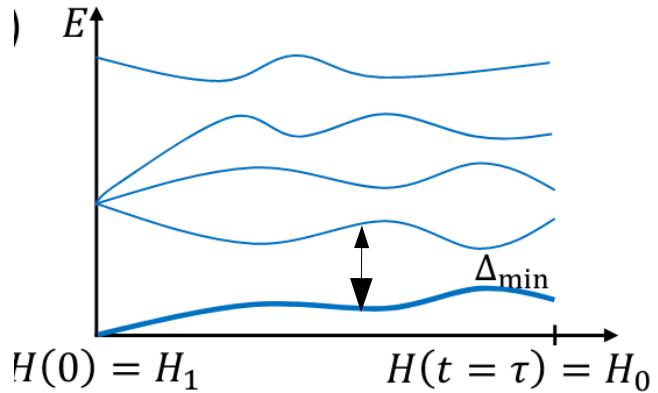


D-wave is promising, but are there **fundamental limitations...**

Limitation of quantum annealing: quantum-phase transitions

Condition for the quantum adiabatic theorem (TD5)

<https://arxiv.org/pdf/1903.06559.pdf>



$$\tau \gg \max_{0 \leq s \leq 1} \frac{\left| \langle 1(s) | \frac{d\tilde{H}(s)}{ds} | 0(s) \rangle \right|}{\Delta_{1,0}(s)^2}. \quad (t = s\tau)$$

↑
Instantaneous ‘gap’ $\Delta_{1,0} = \epsilon_1(s) - \epsilon_0(s)$

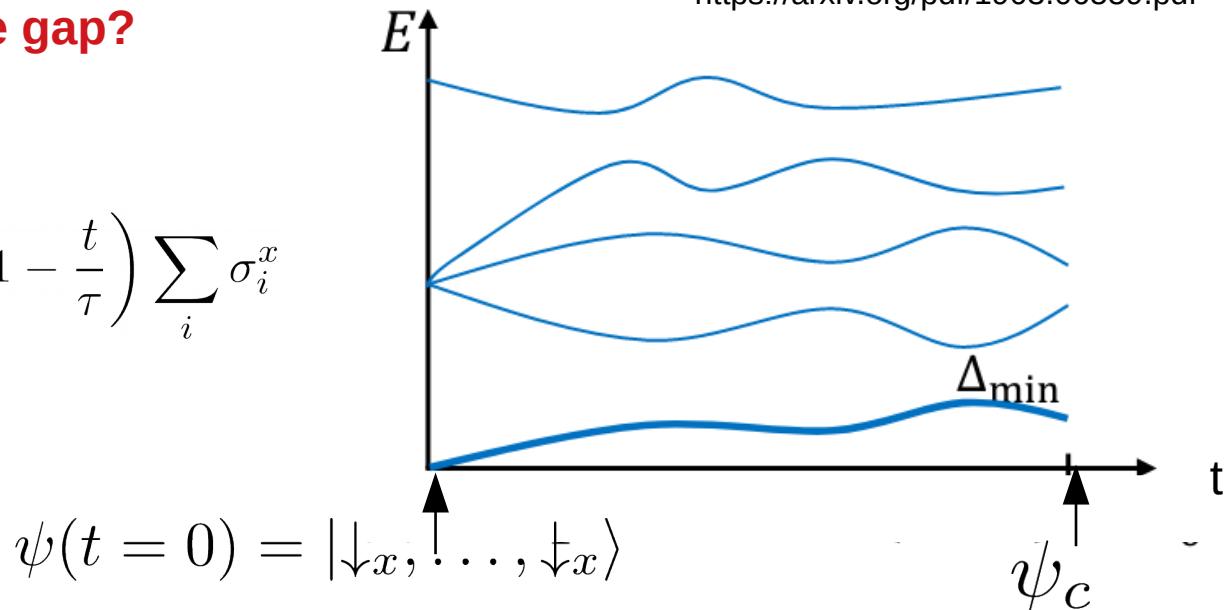
→ Performance of quantum annealing are governed by the size of the gap

Limitation of quantum annealing: quantum-phase transitions

What is the size of the gap?

$$H(t) = \left(\frac{t}{\tau}\right) H_C + \left(1 - \frac{t}{\tau}\right) \sum_i \sigma_i^x$$

<https://arxiv.org/pdf/1903.06559.pdf>



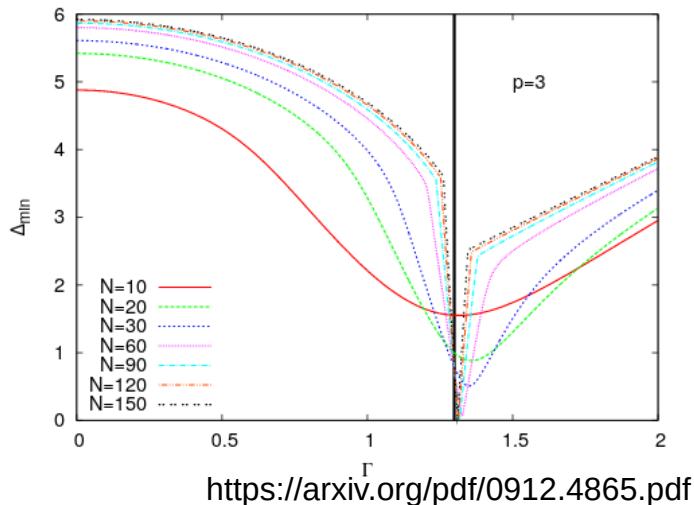
The two ground states correspond to different type of 'orders':

At certain time, the system crosses **a quantum phase transition**

At a QPT, the gap closes as we increase system sizes → It becomes harder to perform annealing

Limitation of quantum annealing: quantum-phase transitions

First order quantum phase transition: gaps closes exponentially with system size → **Hard problem**



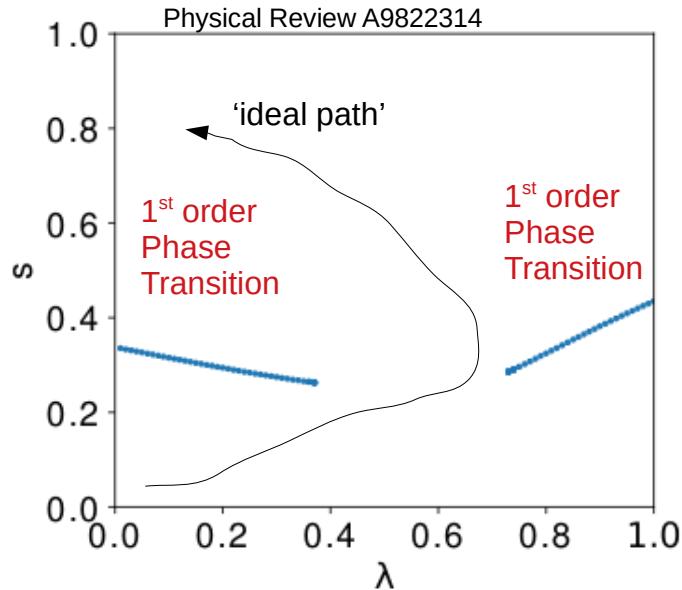
<https://arxiv.org/pdf/0912.4865.pdf>

Second-order phase transition: gap does not close exponentially

Limitation of quantum annealing: quantum-phase transitions

Current efforts:

Solution (1) Try to avoid first order phase transitions



Extra-control parameter $\rightarrow H(s, \lambda)$
$$H(s, \lambda) = f(s)H_0 + g(\lambda)H_1 + H_c$$

Solution (2) Go beyond adiabaticity: Quantum Approximate Optimization Algorithm (QAOA)

The quantum approximate optimization algorithm (E. Farhi 2014)

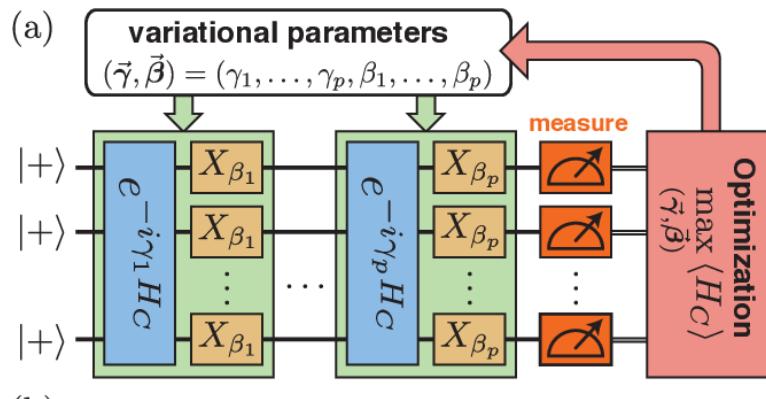
Problem (ex: Max-Cut) H_C

Step 1: Build a ground state candidate for the solution via a quantum computer (a)

$$\psi = \prod_{j=1}^{\eta} e^{i\alpha_j H_C} e^{i\beta_j \sum_l \sigma_l^x} |0000000\rangle$$

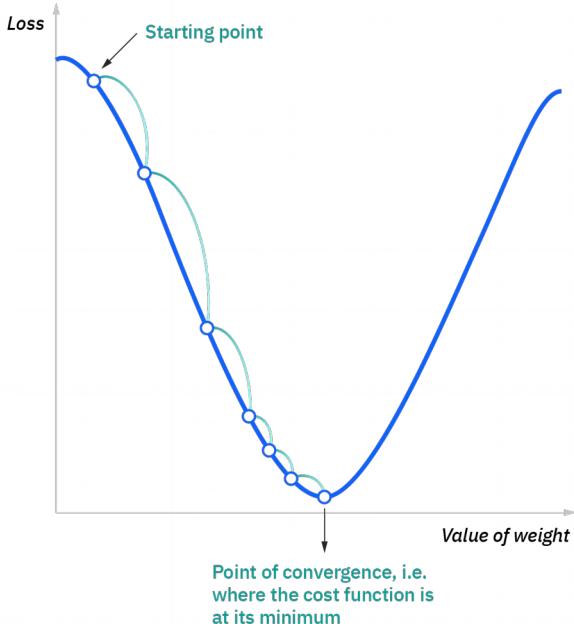
Step 2: Measure the cost-function $\langle \psi | H_c | \psi \rangle$

Step 3: Feed the result into a classical optimization algorithm and try Step 1 with new parameters (α_j, β_j)



The quantum approximate optimization algorithm (E. Farhi 2014)

Example of classical minimization: Gradient-Descent



The structure of algorithm is classical
(example `scipy.optimize`)

However, In QAOA, the evaluation of the cost function
(the energy), is performed on the quantum computer.

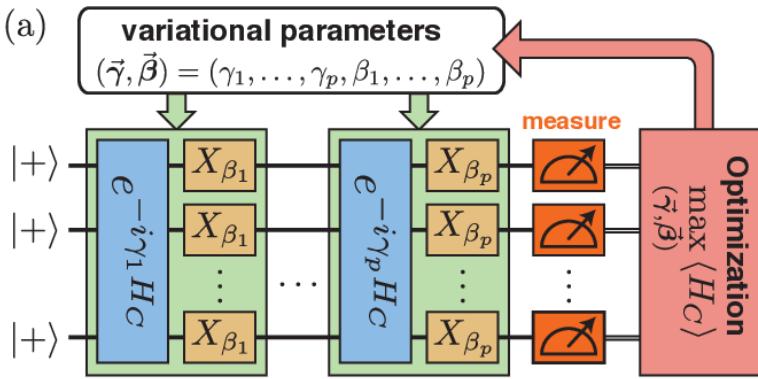
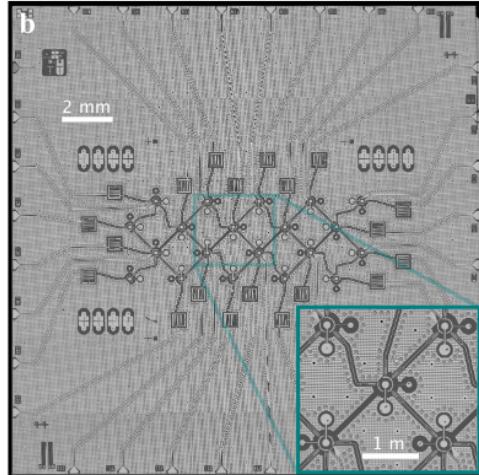
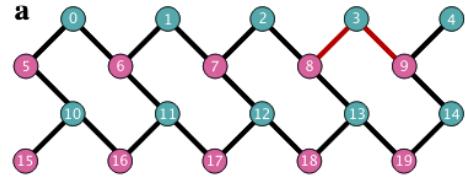
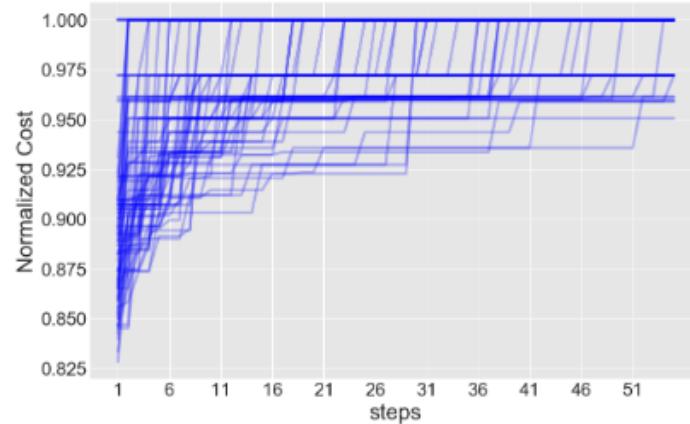


Image Credit: IBM.com

The quantum approximate optimization algorithm (E. Farhi 2014)



Example: Rigetti <https://arxiv.org/pdf/1712.05771.pdf>



See also more recent attempts by Google

The quantum approximate optimization algorithm (E. Farhi 2014)

Quantum annealing → **QAOA**: Its not clear whether this method

- 1) can solve large classical problems (ex: Max-Cut) Phys. Rev. X 10, 021067 (2020)
- 2) outperforms quantum annealing?

In the mean time, can we solve **quantum** problems?

Solving quantum problems with quantum computers

Translating quantum chemistry for quantum computers

Nature Communications volume 5, Article number: 4213 (2014)

$$\mathcal{H}(R) = \sum_{pq} h_{pq}(R) \hat{a}_p^\dagger \hat{a}_q + \sum_{pqrs} h_{pqrs}(R) \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s$$

Goal: Find the ground state of the molecule

Step 1: Express the Hamiltonian has a qubit Hamiltonian: **Jordan-Wigner Transformation**

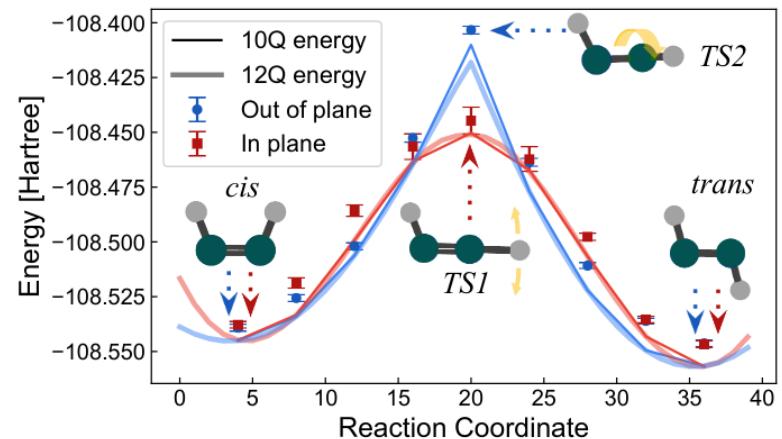
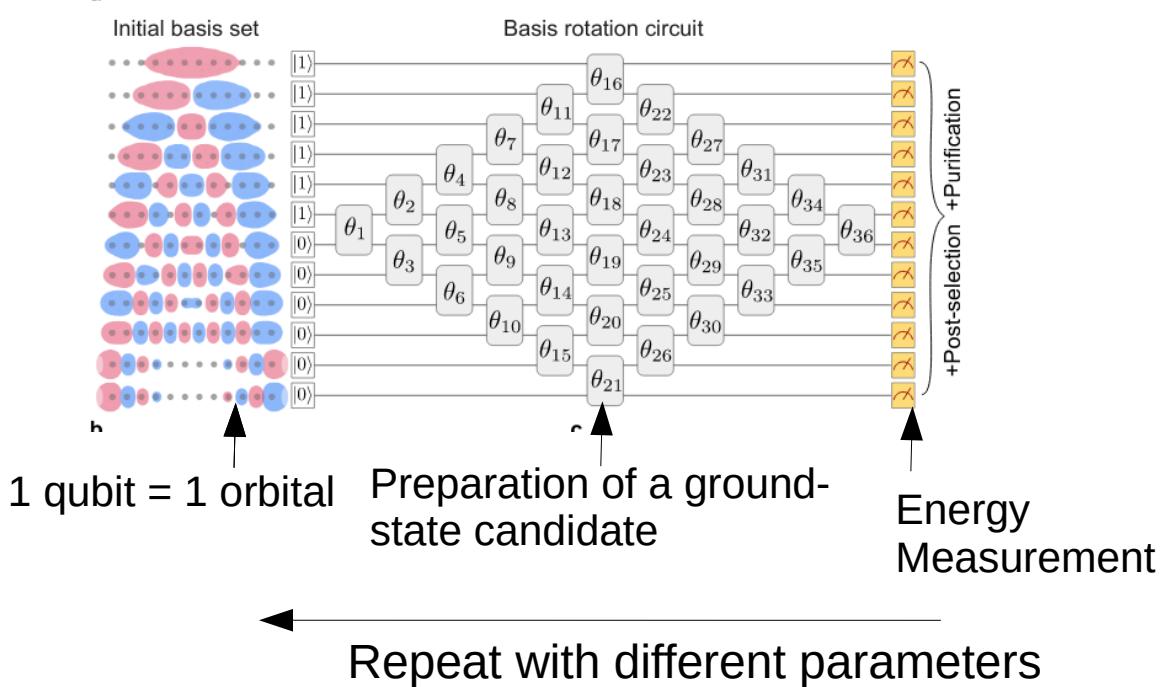
$$\begin{aligned}\hat{a}_j &\rightarrow I^{\otimes j-1} \otimes \sigma_+ \otimes \sigma_z^{\otimes N-j} \\ \hat{a}_j^\dagger &\rightarrow I^{\otimes j-1} \otimes \sigma_- \otimes \sigma_z^{\otimes N-j}\end{aligned}$$

Step 2: Find the ground-state by quantum-approximate-optimization-algorithm (In this context, it's called **Variational Quantum Eigensolver (VQE)**)

Solving quantum problems with quantum computers

Illustration with Google's Sycamore

<https://arxiv.org/pdf/2004.04174.pdf>



Conclusions on quantum optimization

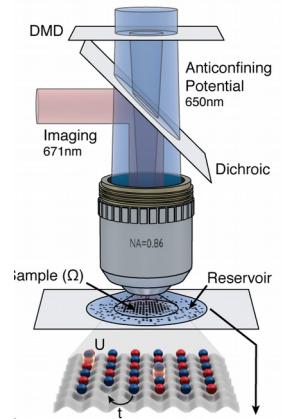
	Quantum annealing	Quantum Approximate-Optimization Algorithm	Variational Quantum Eigensolver
Typical Problem:	Classical	Classical	Quantum
Limitations:	Scaling of the Gap	?	?

Summary: Quantum algorithms

- **Quantum computers do exist** and implement quantum circuits
- There are **quantum algorithms** that offer quantum speedup (Grover, Shor, etc)
- **Quantum error correction:** conceptually 'solved' - technical challenges
- New quantum algorithms that do not rely necessarily on quantum error correction:
Quantum optimization/Quantum simulation

→

Quantum information meets many-body physics!



A very active research field, many developments expected in the next years!