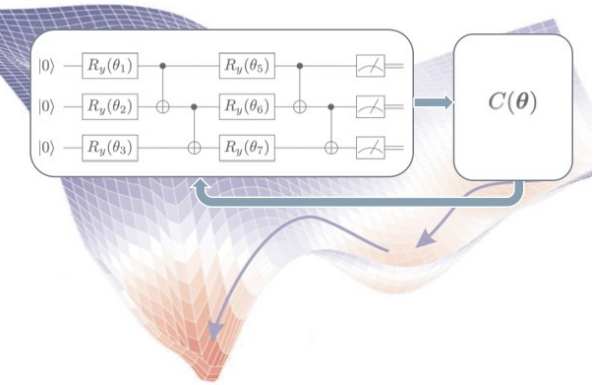
IGFAE
Instituto Galego de Física de Altas Enerxías
USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA
XUNTA DE GALICIA
MARIE CURIE ACTIONS
Funded by the European Union

Variational quantum algorithms: VQE, QAOA



Wenyang Qian

University of Santiago de Compostela

Mar 7, 2025, USC QC Workshop

https://indico.cern.ch/event/1480598/

Quantum computing

The Idea: *"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical"*

Simulating Physics with Computers

Richard P. Feynman

Department of Physics, California Institute of Technology, Pasadena, California 91107

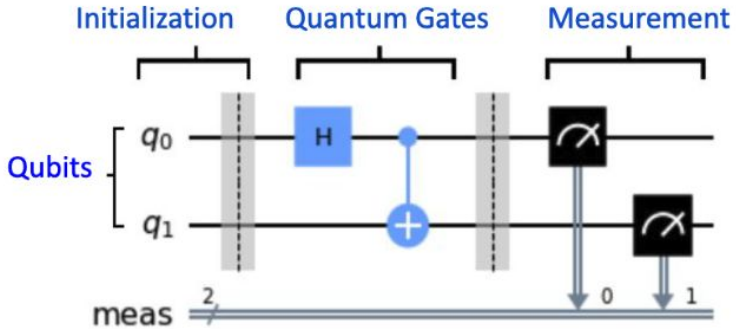
Received May 7, 1981

Quantum computing (QC) is a rapidly-emerging technology that harnesses the laws of quantum mechanics to solve problems.

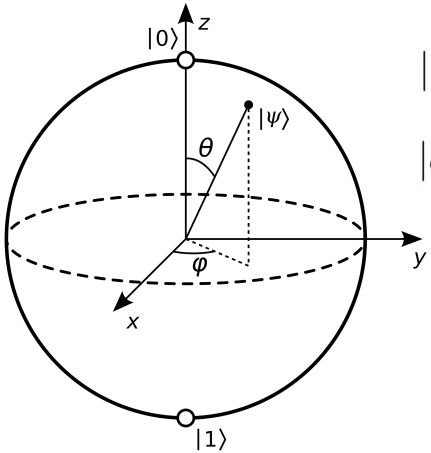
After 40+ years, we are almost there with emerging QC technology…

Quantum computing

Basic idea: Spin chain = qubits (lines) + unitary gates (operators)



Qubits = Digitalization



$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

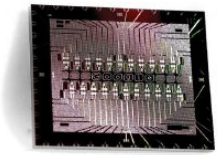
$$|\alpha|^2 + |\beta|^2 = 1$$

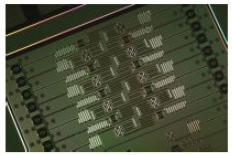
Conceptually clean for universal simulation

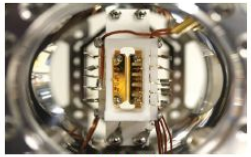
Noisy, intermediate-scale (NISQ) era, ~100 qubits

Quantum computing

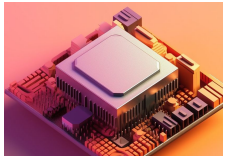
State-of-the-art: Noisy intermediate-scale quantum (NISQ) era = substantially imperfect and insufficient qubits. However, this can change fast!


Google


IBM


Rigetti


Intel


USTC

...

Qiskit

Cirq

XANADU

...

Optimist's prediction:

Neven's law = Double-exponential scaling

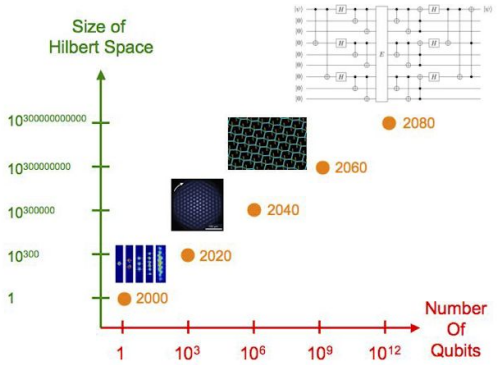
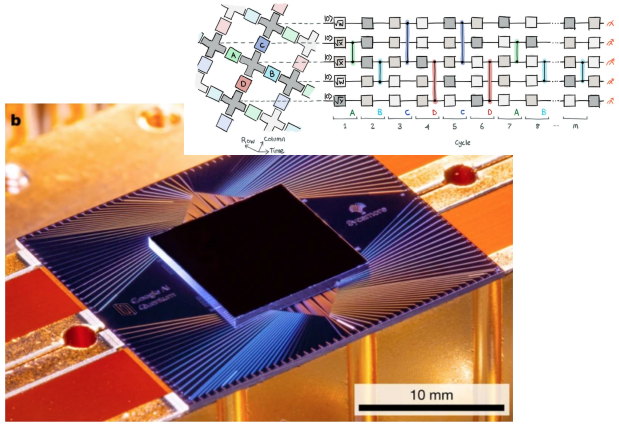


Chart: Quantum Pundit

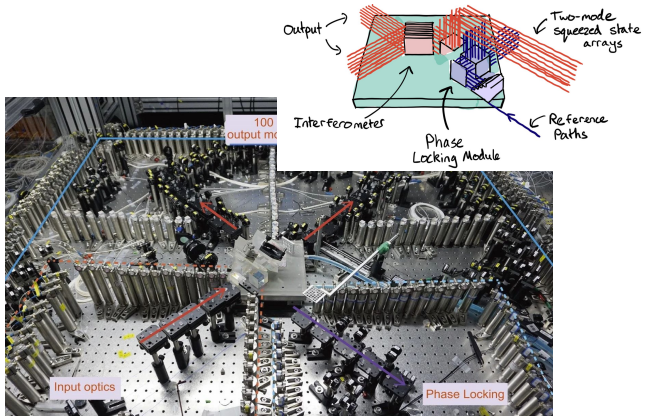
Quantum supremacy

Quantum supremacy = **anything** with a quantum device “cannot” be performed classically

Specific evidence for supremacy are found in sampling distributions!



random circuit 53 qubits, Google Quantum, 1910.11333

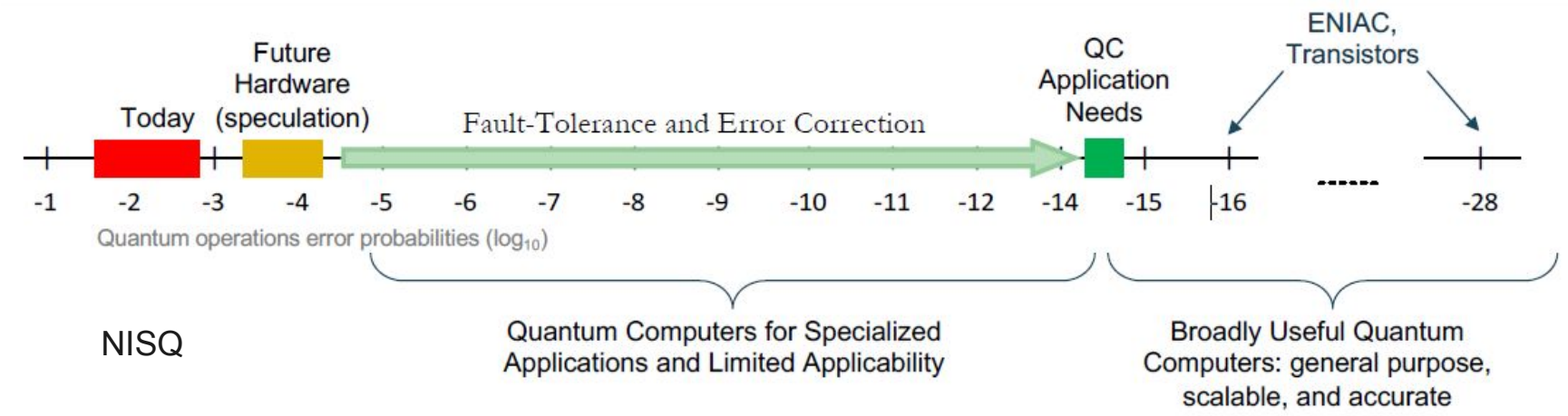


GBS 76 qubits, UTSC, 2012.01625
Schematic: Pennylane

Quantum advantage = **sth useful** involving a quantum device “cannot” be performed classically

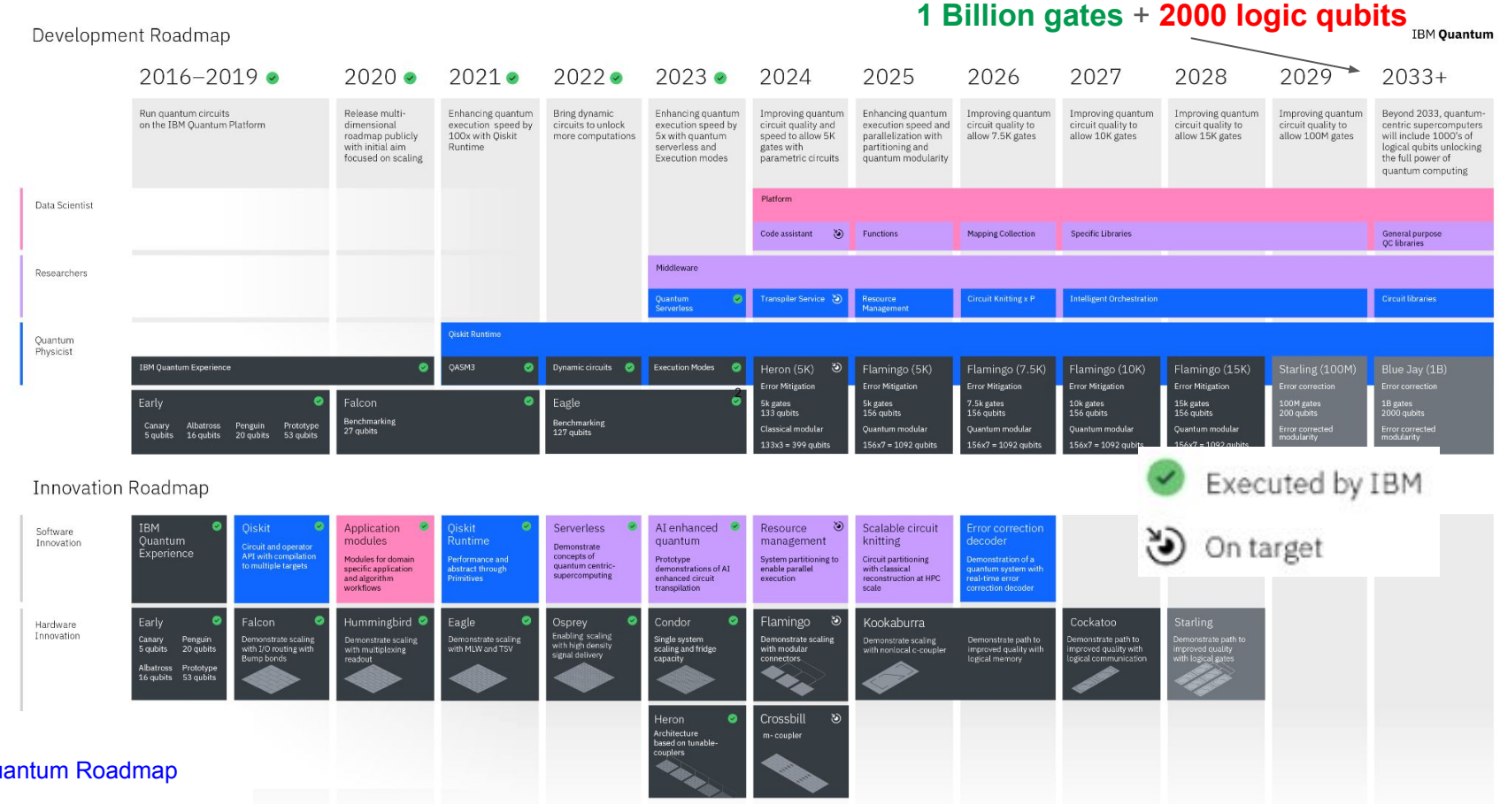
Quantum supremacy

Simply a matter of time before QC revolutionizes the modern research (**sth useful**)



Schematic: De Jong

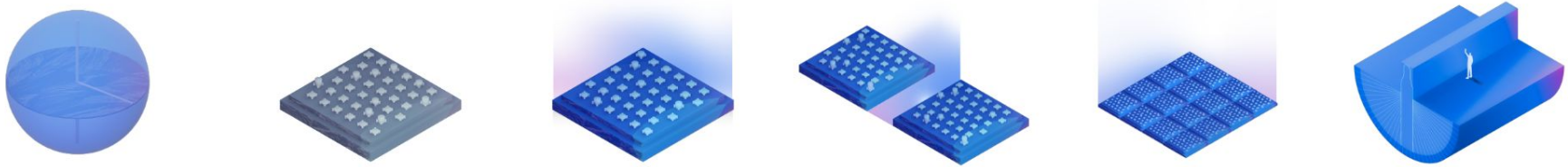
Quantum technology is growing at fast pace



IBM Quantum Roadmap

Quantum technology is growing at fast pace

Our focus is to unlock the full potential of quantum computing by developing a large-scale computer capable of complex, error-corrected computations. We're guided by a roadmap featuring six milestones that will lead us toward top-quality quantum computing hardware and software for meaningful applications.



MILESTONE 1

BEYOND CLASSICAL

Physical Qubits: 54
Logical Qubit Error Rate: -

2019

MILESTONE 2

QUANTUM ERROR CORRECTION

Physical Qubits: 10^2
Logical Qubit Error Rate: 10^{-2}

2023

MILESTONE 3

BUILDING A LONG-LIVED LOGICAL QUBIT

Physical Qubits: 10^3
Logical Qubit Error Rate: 10^{-6}

MILESTONE 4

CREATING A LOGICAL GATE

Physical Qubits: 10^4
Logical Qubit Error Rate: 10^{-6}

MILESTONE 5

ENGINEERING SCALE UP

Physical Qubits: 10^5
Logical Qubit Error Rate: 10^{-6}

MILESTONE 6

LARGE ERROR-CORRECTED QUANTUM COMPUTER

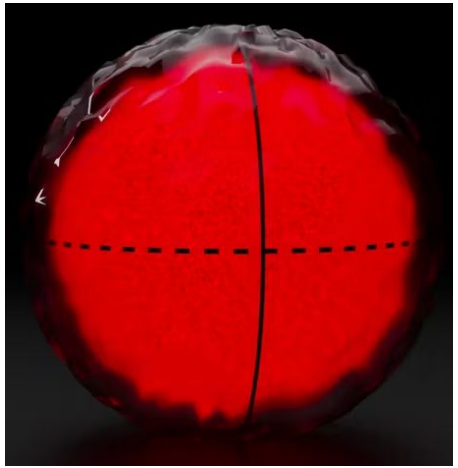
Physical Qubits: 10^6
Logical Qubit Error Rate: 10^{-13}

1000 qubits, eror rate: 10^-6

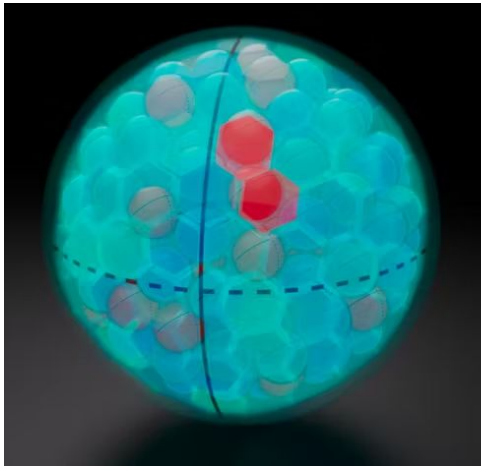
Error correction, 100000 qubits, 10^-13 error

Google Quantum AI Roadmap

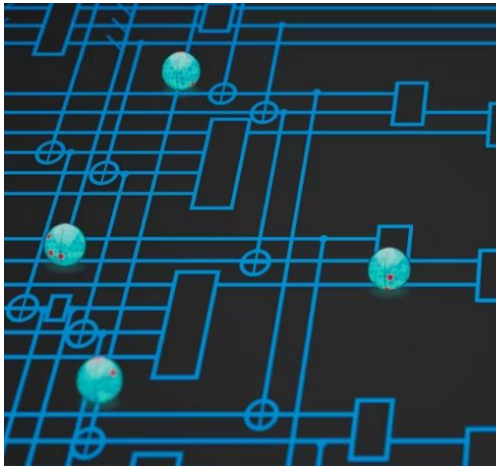
Quantum technology is growing at fast pace



Noisy physical qubits



Reliable logical qubits

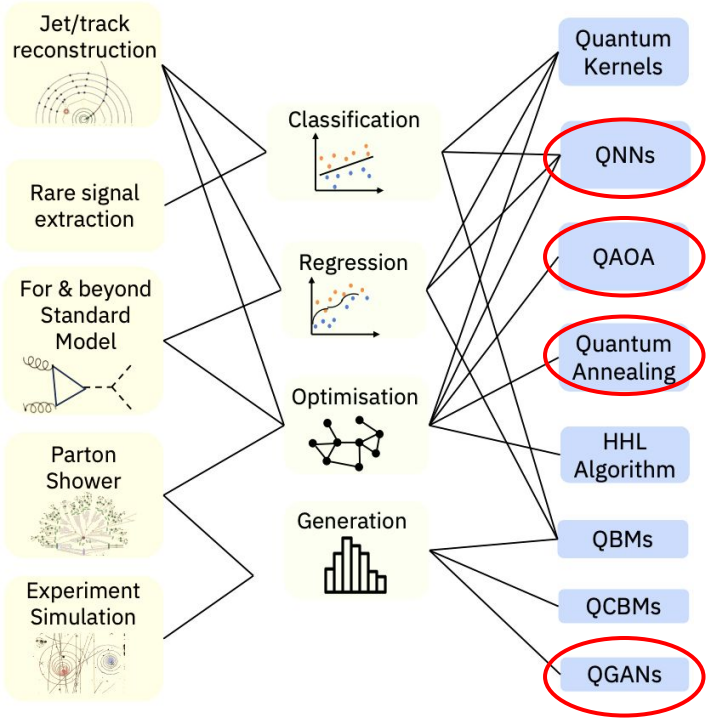


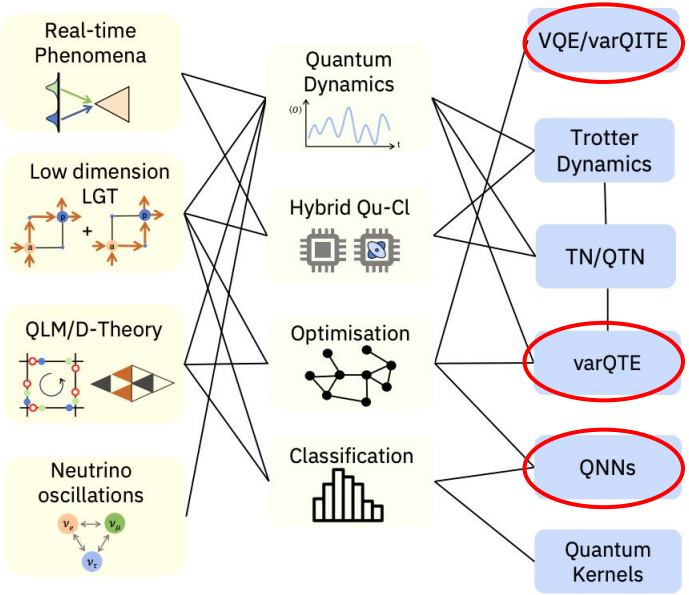
Quantum supercomputers

Variational quantum computing

See recent/comprehensive review:

Delgado et al 2203.08805
Bauer et al, 2204.03381
Di Meglio et al, 2307.03236





Motivations

LHC Physics = large data
Quantum search provides theoretical speedup

Classical simulation encounters inherent problems
Quantum simulation is an ultimate path to simulate QFT

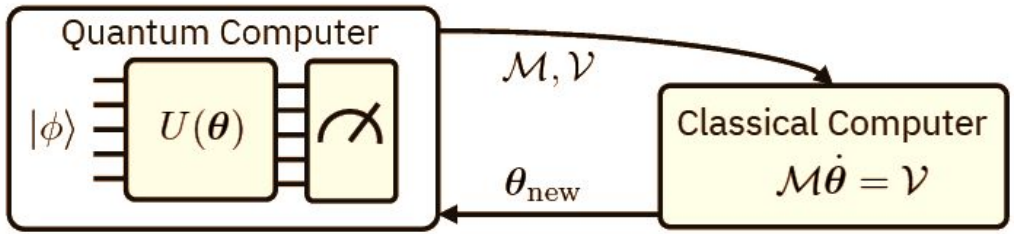
Variational Quantum Algorithm

Perhaps the most successful applications of QC is the variational quantum algorithm (VQA)

Based on the Variational Principle $\langle\psi_{\text{guess}}| H |\psi_{\text{guess}}\rangle \geq \langle E_0\rangle = \langle\psi_0| H |\psi_0\rangle$

By tweaking the ansatz (guess wavefunction), we can obtain the ground state energy of Hamiltonian

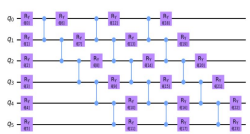
Optimal circuit (i.e. solution) to the chemistry or physics problem is obtained by collaboration between quantum device (QPU) and CPU.



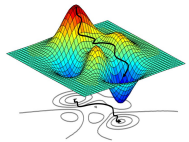
Variational Quantum Algorithm

Peruzzo, et al, 1304.3061 (2014)
Nakanishi, Mitarai, Fujii, 1810.09434 (2019)

VQA is a collaborative effort between the quantum simulation and classical optimization



Problem-inspired
Heuristic ansatz



$|\psi(\theta)\rangle = U(\theta)\,|\psi_{\text{init}}\rangle$

Update θ

Quantum simulation

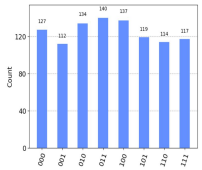
Optimization Loop

Classical optimization

Probability Histogram

$C = \langle\psi(\theta)|H|\psi(\theta)\rangle$

Cost function



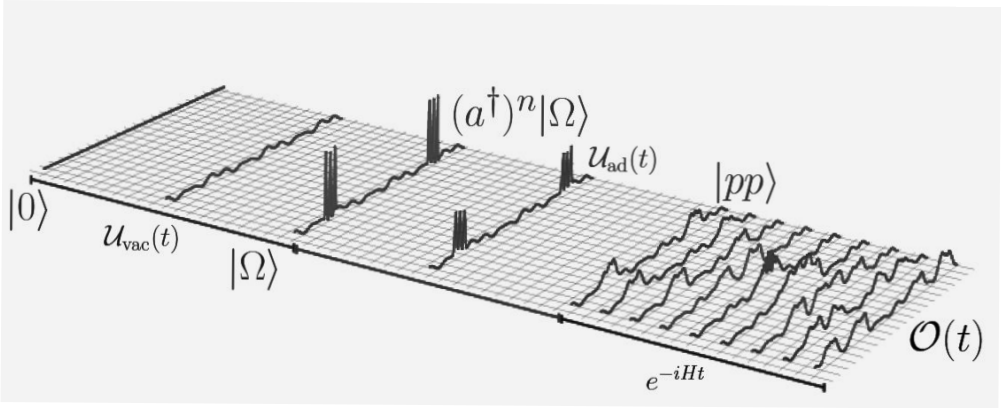
Hybrid approach:
Complementary Advantage for QC and CC

Different variants: VQE, SSVQE, QAOA, QNN, etc

Quantum simulation I

Wiesner, 9603028 (1996); Zalka, 9603026 (1996)

1. **Define problem of interest**
2. Encode onto qubits
3. Prepare initial states
4. Variational ansatz
5. Measurement protocol

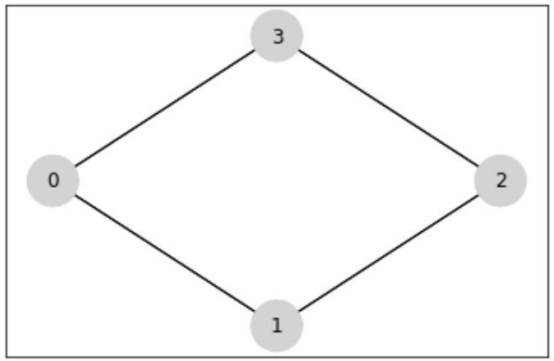


Lamm's talk at Fermilab (2021)

Okay, let us find an interesting problem?

Maxcut problem (maximum cut)

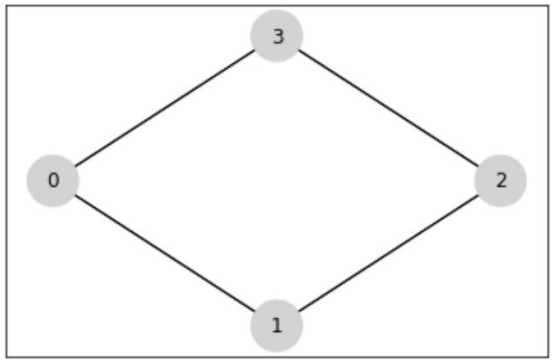
Given a graph of n nodes, how do we cut the graph into two partitions such that the number of edges between the two partitions are maximized?



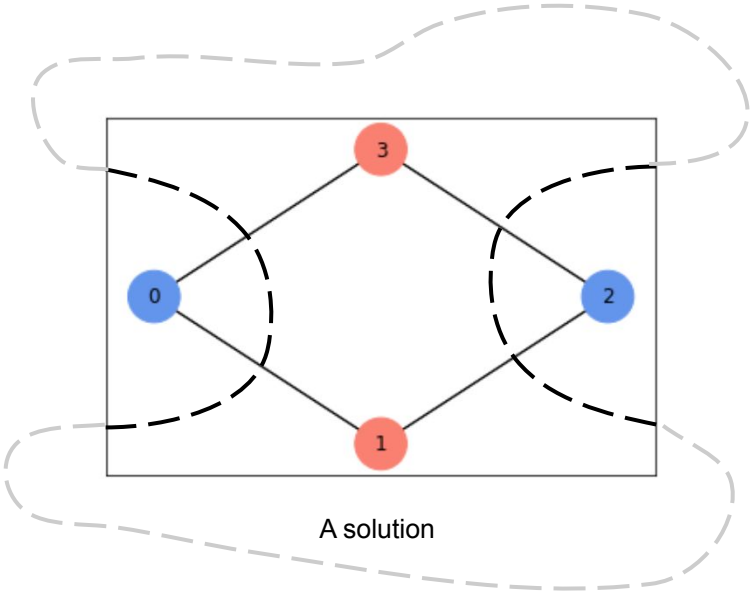
Initial Problem

Maxcut problem (maximum cut)

Given a graph of n nodes, how do we cut the graph into two partitions such that the number of edges between the two partitions are maximized?



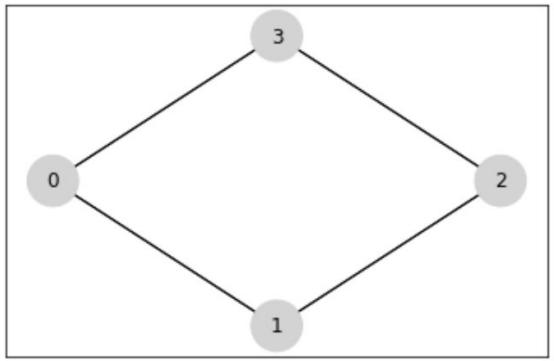
Initial Problem



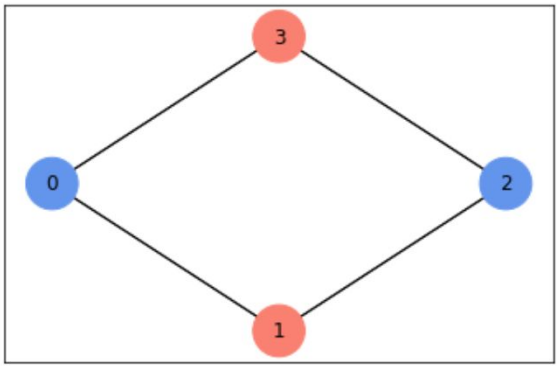
A solution

Maxcut problem (maximum cut)

Given a graph of n nodes, how do we cut the graph into two partitions such that the number of edges between the two partitions are maximized?



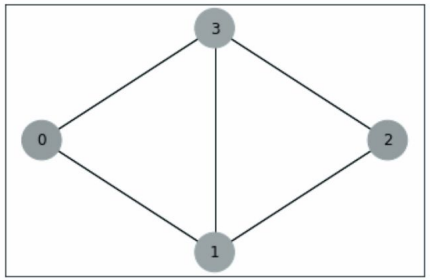
Initial Problem

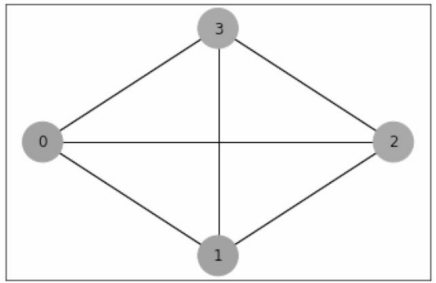


A solution

Equivalent to a **coloring problem** to partition the nodes into Red or Blue

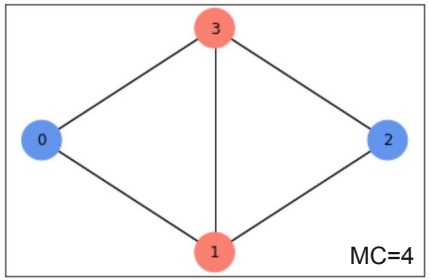
Maxcut problem is NP-Hard!

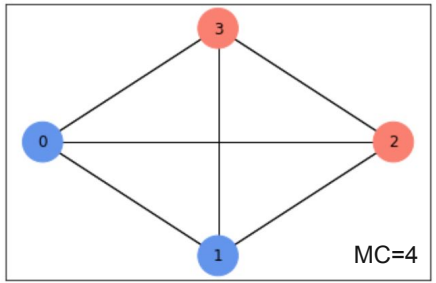




...

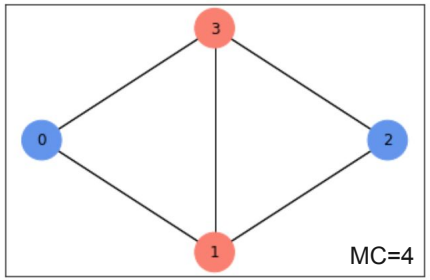
Maxcut problem is NP-Hard!

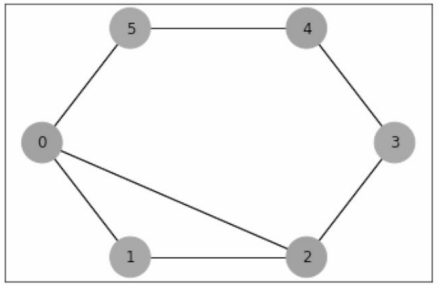




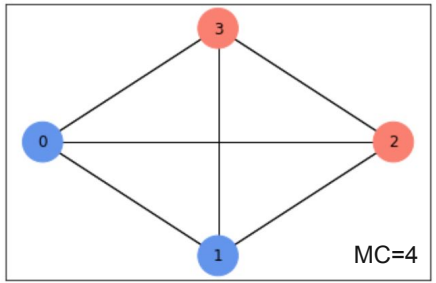
...

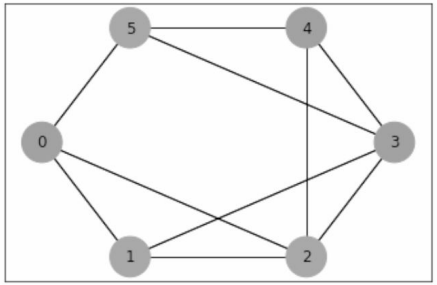
Maxcut problem is NP-Hard!



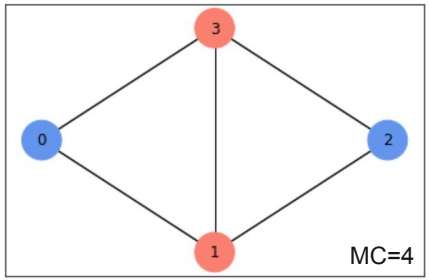


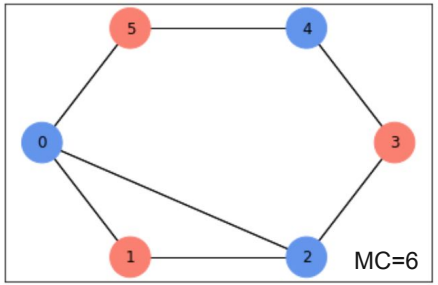
...



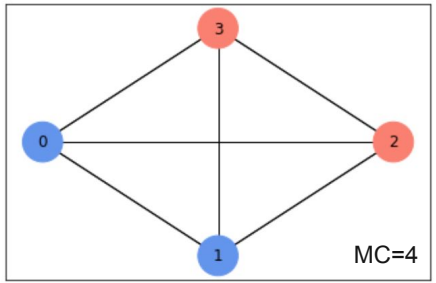


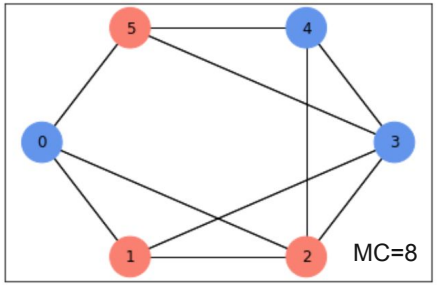
Maxcut problem is NP-Hard!



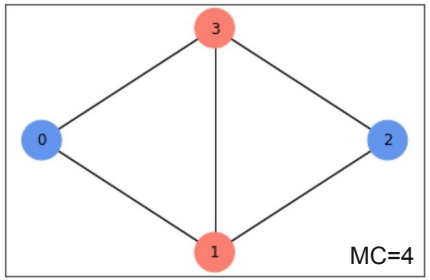


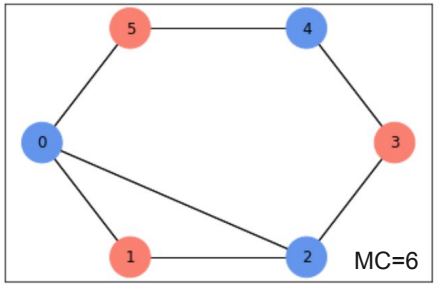
...

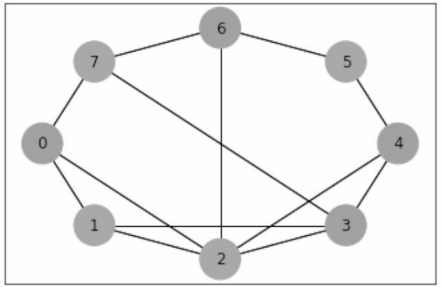




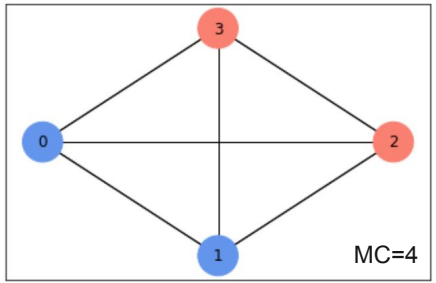
Maxcut problem is NP-Hard!

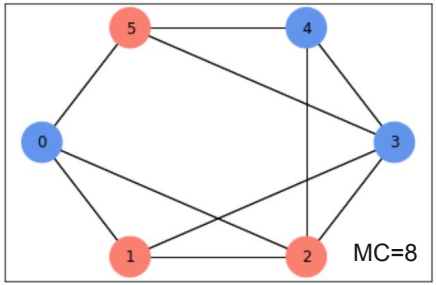


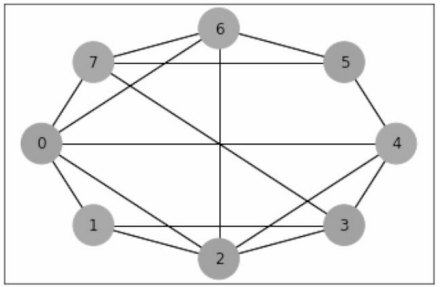




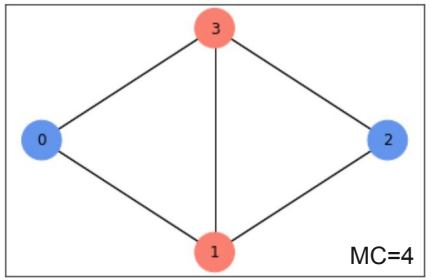
...

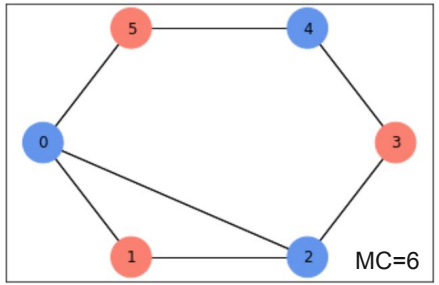


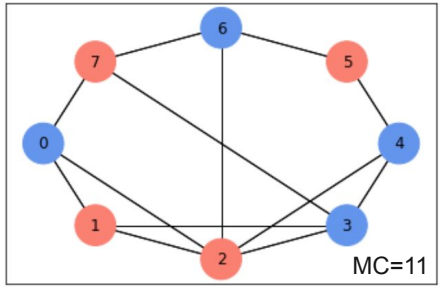




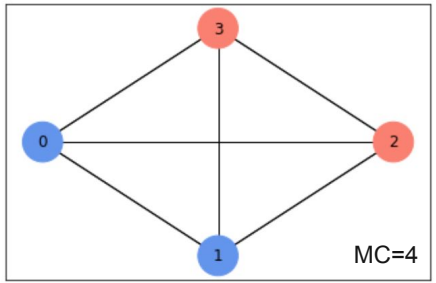
Maxcut problem is NP-Hard!

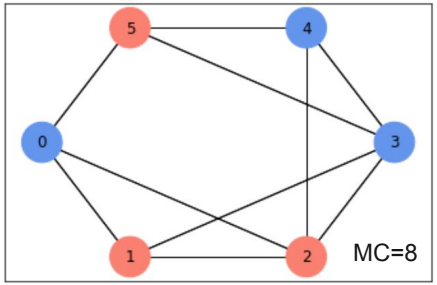


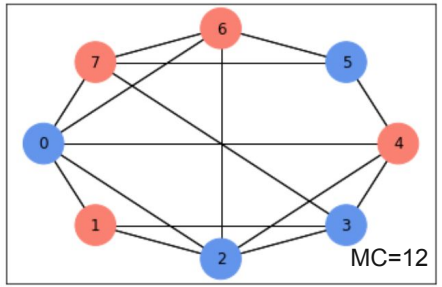




...

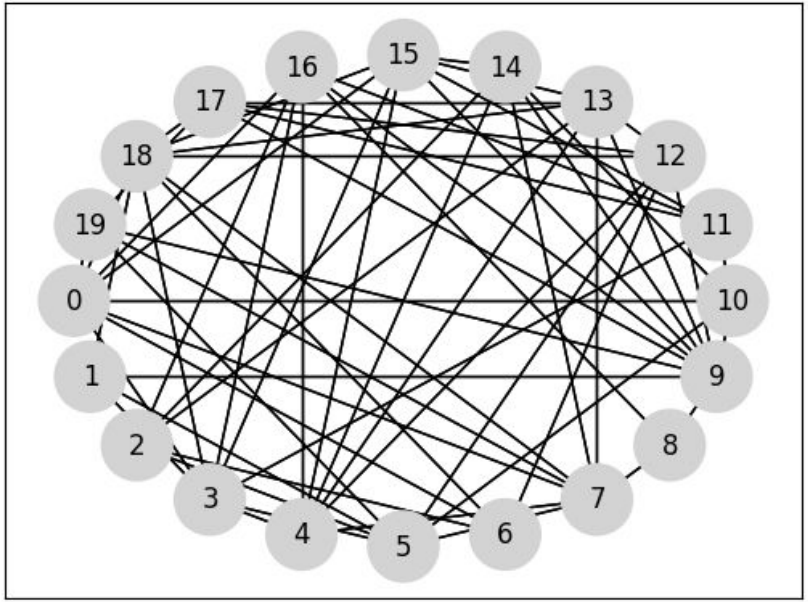






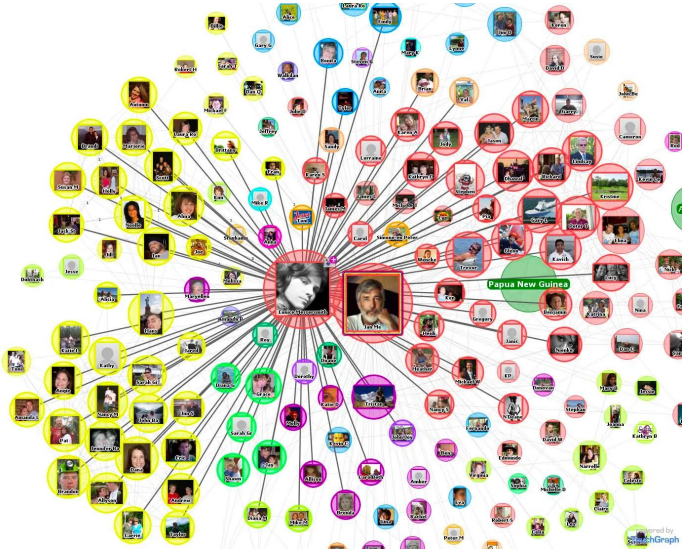
Maxcut problem is NP-Hard!

Eventually, bruteforce solutions $\mathcal{O}(n!)$ for large graphs becomes impractical…



n = 20 is already so slow to compute…

Maxcut problem is universal & important



Social network:
recommending new Friends

Vehicle routing:
finding optimal path

Images are from Google Search

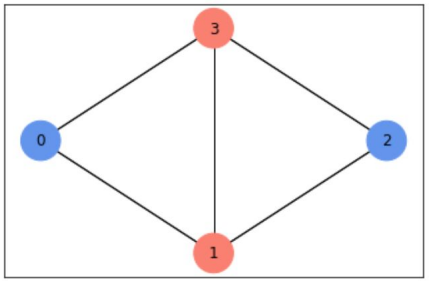
Maxcut problem state representation

How to leverage quantum computing to solve Maxcut problems?

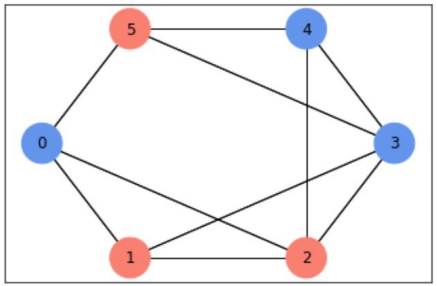
By representing a solution to the n-node Maxcut using binary sequence!

where $x_i = 0$ indicates node-i in Partition 0 (red), and $x_i = 1$ indicates node-i in Partition 1 (blue).

$$\boldsymbol{x} = x_0 x_1 x_2 \cdots x_{n-1} = |x_0\rangle\,|x_1\rangle\,|x_2\rangle \cdots |x_{n-1}\rangle$$



$$\boldsymbol{x} = 1010 = |1\rangle_0\,|0\rangle_1\,|1\rangle_2\,|0\rangle_3$$



$$\boldsymbol{x} = 100110 = |100110\rangle$$

Maxcut problem cost function

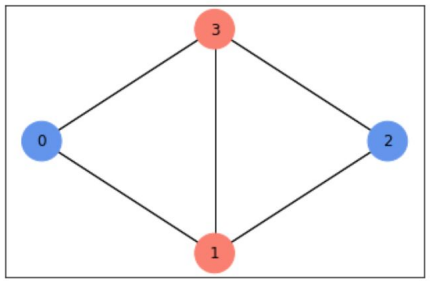
With these binary variables, we can build a cost function for any Maxcut problem

$$H(\boldsymbol{x}) = \sum_{i,j} w_{ij} x_i (1 - x_j)$$

$$w_{ij} = \begin{cases} 1, & \text{if edge}(i,j) \in G \\ 0, & \text{if edge}(i,j) \notin G \end{cases}$$

connectivity (adjacency matrix)

We claim a solution to the maxcut problem must **maximize this Hamiltonian**!



$$\boldsymbol{x} = 1010 = |1\rangle_0 |0\rangle_1 |1\rangle_2 |0\rangle_3$$

$$w = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

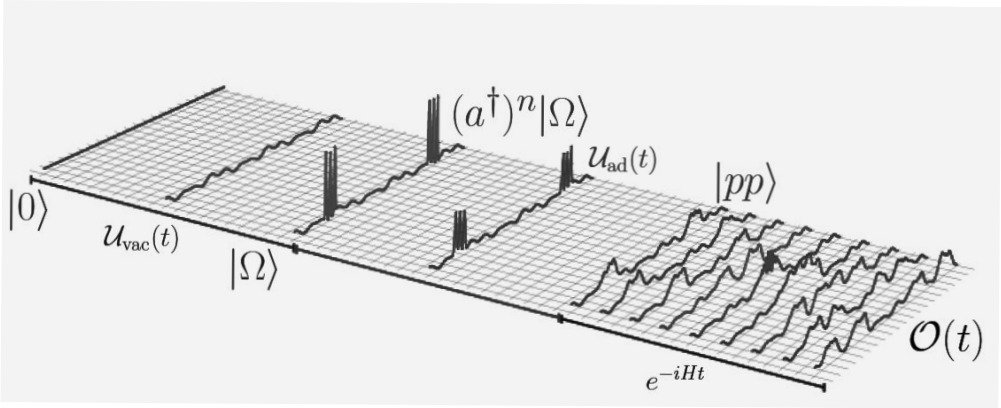
$$w_{01}x_0(1-x_1) = w_{01} \cdot 1 \cdot (1-0) = 1$$
$$w_{03}x_0(1-x_3) = w_{03} \cdot 1 \cdot (1-0) = 1$$
$$w_{10}x_1(1-x_0) = w_{10} \cdot 0 \cdot (1-1) = 0$$
$$w_{12}x_1(1-x_2) = 0 \qquad w_{30}x_3(1-x_0) = 0$$
$$w_{13}x_1(1-x_3) = 0 \qquad w_{31}x_3(1-x_1) = 0$$
$$w_{21}x_2(1-x_1) = 1 \qquad w_{32}x_3(1-x_2) = 0$$
$$w_{23}x_2(1-x_3) = 1$$

$$H(\boldsymbol{x}) = 4$$

Quantum simulation II

Wiesner, 9603028 (1996); Zalka, 9603026 (1996)

1. Define problem of interest
2. **Encode onto qubits**
3. Prepare initial states
4. Variational ansatz
5. Measurement protocol



Lamm's talk at Fermilab (2021)

How do we efficiently encode the problem to qubits?

Pauli matrices

The problem of interest can be expressed in terms of Pauli matrices, I, X, Y, Z

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \vec{\sigma} = (\sigma^0, \sigma^1, \sigma^2, \sigma^3) = (I, X, Y, Z)$$

such that we end up with a linear combinations of products of these Pauli matrices

$$H = \sum_i c_i P_i$$

For example:

$$\begin{aligned} H_{ex} &= I_0 I_1 + X_0 X_1 + Y_0 Y_1 + Z_0 Z_1 \\ &= II + XX + YY + ZZ \end{aligned} \qquad \exp(-iH_{ex}t) = \exp(-i\big[II + XX + YY + ZZ\big]t)$$

Quantum state in Pauli-Z basis

Most quantum devices are in Z-basis or computational basis $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$

In fact any operations can be represented in this way:

$$\sigma_0 \equiv I \equiv \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} = |0\rangle\langle 0| + |1\rangle\langle 1|, \qquad \sigma_2 \equiv Y \equiv \begin{pmatrix}0 & -i\\i & 0\end{pmatrix} = i|1\rangle\langle 0| - i|0\rangle\langle 1|,$$

$$\sigma_1 \equiv X \equiv \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} = |1\rangle\langle 0| + |0\rangle\langle 1|, \qquad \sigma_3 \equiv Z \equiv \begin{pmatrix}1 & 0\\0 & -1\end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|$$

Including the Control-Not gate!

$$\mathrm{CNOT} = \left(|0\rangle\langle 0|\right)_0 \otimes I_1 + \left(|1\rangle\langle 1|\right)_0 \otimes X_1$$

Maxcut problem qubit encoding

To map the maxcut problem to quantum circuit, we use

$$x_i \to \frac{I - Z}{2} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Exercise: Why? check eigenvalue and eigenvector

$$Z|0\rangle = +1|0\rangle$$
$$Z|1\rangle = -1|1\rangle$$

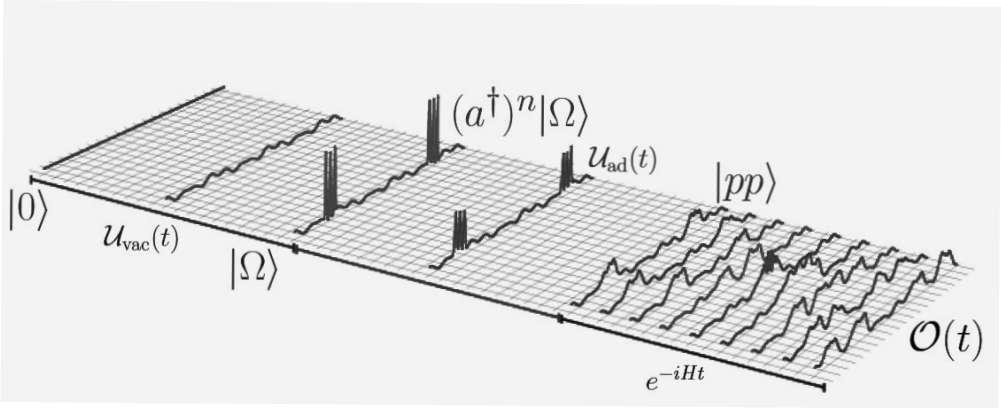
Then, Maxcut cost function becomes a sum of Pauli-Z and Pauli-ZZ operators (i.e., Ising Hamiltonian, or sometimes known as the **QUBO** problem)

$$H \to \sum_{ij} w_{ij} \frac{I_i - Z_i}{2} \left(I - \frac{I_j - Z_j}{2} \right) = \sum_i c_i Z_i + \sum_{ij} d_{ij} Z_i Z_j$$

Quantum simulation III

Wiesner, 9603028 (1996); Zalka, 9603026 (1996)

1. Define problem of interest
2. Encode onto basis
3. **Prepare initial states**
4. Variational Ansatz
5. Measurement protocol



Lamm's talk at Fermilab (2021)

What initial state should we use?

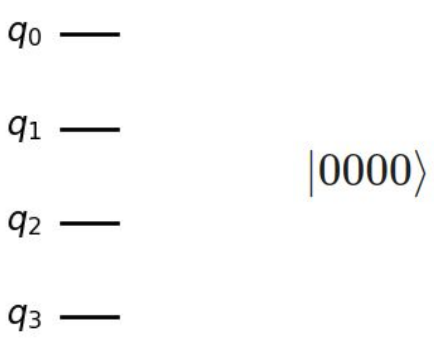
State preparation

State preparation (initialization) is the First step of quantum computing

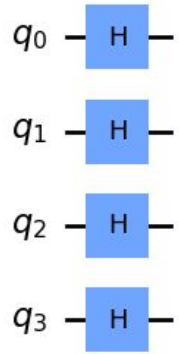
Unlike classical computation, one cannot simply prepare an arbitrary state in QC. Often it involves careful analysis of the problem and genius engineering…

Here, we look at two possible initializations

Hadamard gate: $H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$



$|0000\rangle$



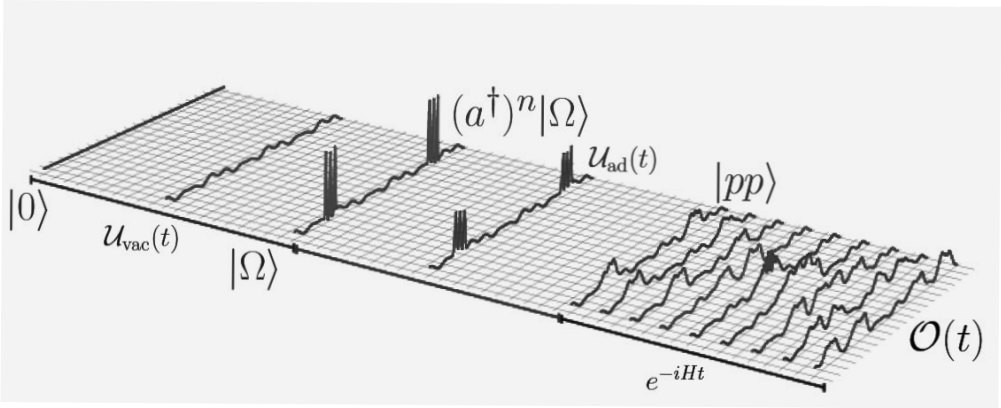
$H_0 H_1 H_2 H_3 |0000\rangle$

Exercise:
What does Hadamard initialization give you?

Quantum simulation IV

Wiesner, 9603028 (1996); Zalka, 9603026 (1996)

1. Define problem of interests
2. Encode onto basis
3. Prepare initial states
4. **Variational Ansatz**
5. Measurement protocol

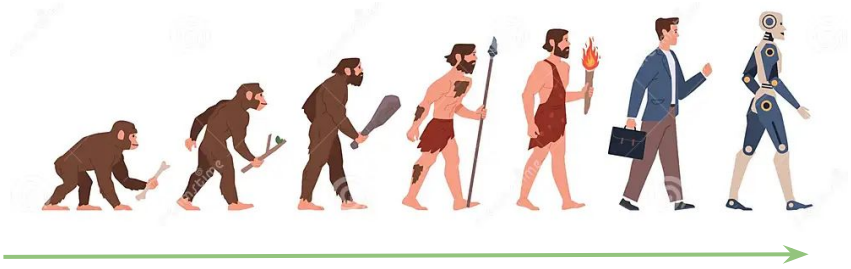


Lamm's talk at Fermilab (2021)

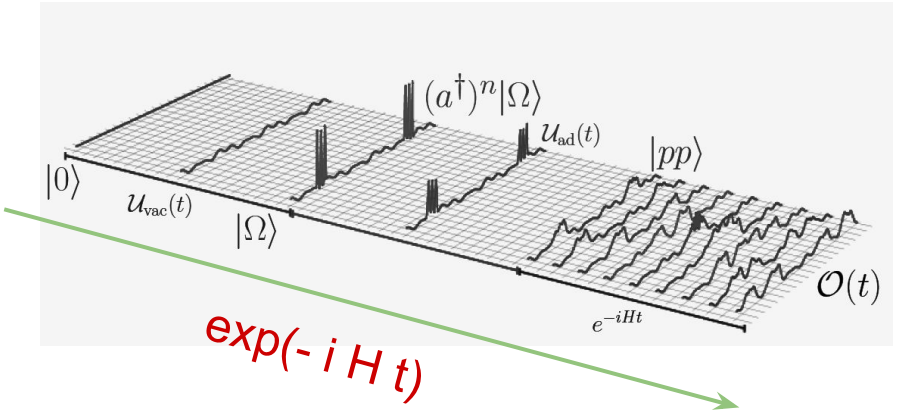
What does variational ansatz (quantum evolution) mean?

Variational ansatz = Unitary quantum evolution

Literally means evolving some quantum state by applying an operator H for a parameter t.



exp(- i H t)



exp(- i H t)

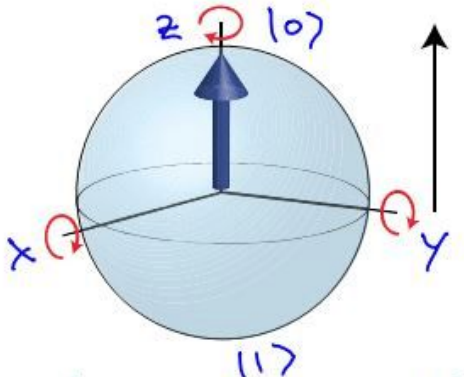
Single qubit simulation

Rotational gates take care single qubit Pauli terms (Taylor expansion)

$$e^{-iXt} = \cos(t)I - i\,\sin(t)X = \begin{pmatrix} \cos(t) & -i\sin(t) \\ -i\sin(t) & \cos(t) \end{pmatrix} \qquad R_x(2t)$$

$$e^{-iYt} = \cos(t)I - i\,\sin(t)Y = \begin{pmatrix} \cos(t) & -\sin(t) \\ \sin(t) & \cos(t) \end{pmatrix} \qquad R_y(2t)$$

$$e^{-iZt} = \cos(t)I - i\,\sin(t)Z = \begin{pmatrix} e^{-it/2} & 0 \\ 0 & e^{it/2} \end{pmatrix} \qquad R_z(2t)$$

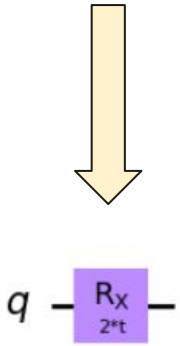


These are just
Rotational gates on Bloch sphere!

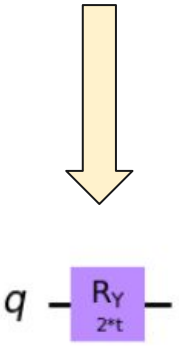
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Single qubit simulation

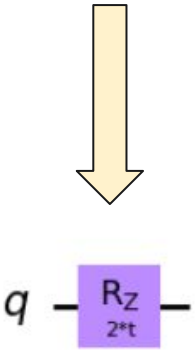
e^{-iXt}



e^{-iYt}

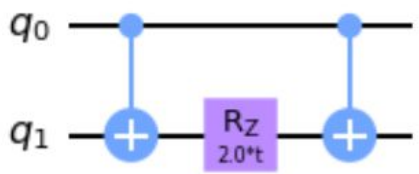


e^{-iZt}



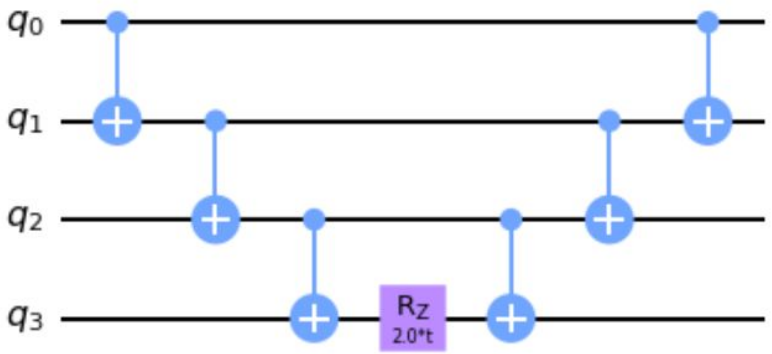
Multiple qubit simulation

$$\exp(-iZ_0Z_1t) = \mathrm{CNOT}_{0,1}\exp(-iZ_1t)\mathrm{CNOT}_{0,1}$$



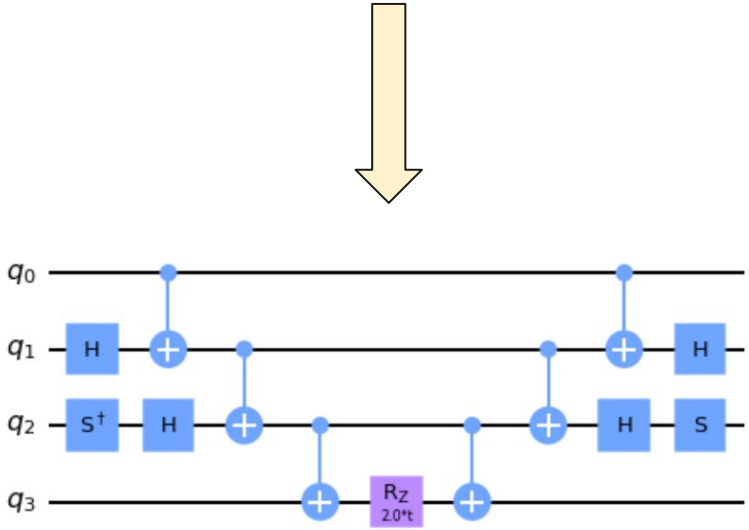
Multiple qubit simulation

The results can be generalized to exp(-i ZZ….Z t), where the circuit is a tower of control gates
For example, exp(-i ZZZZ t) becomes

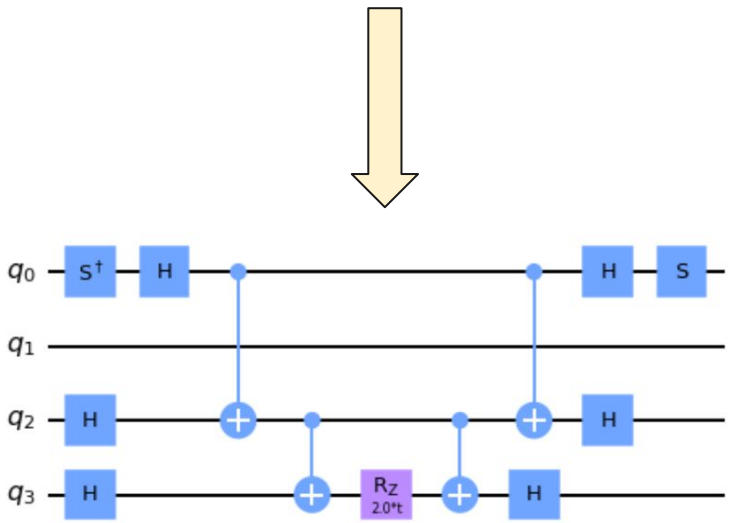


Multiple qubit simulation

$\exp(-iZ_3Y_2X_1Z_0t)$



$\exp(-iX_3X_2I_1Y_0t)$



$X = HZH^\dagger, \quad Y = (SH)Z(SH)^\dagger$

$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$S = \sqrt{Z} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$

Two types of VQA

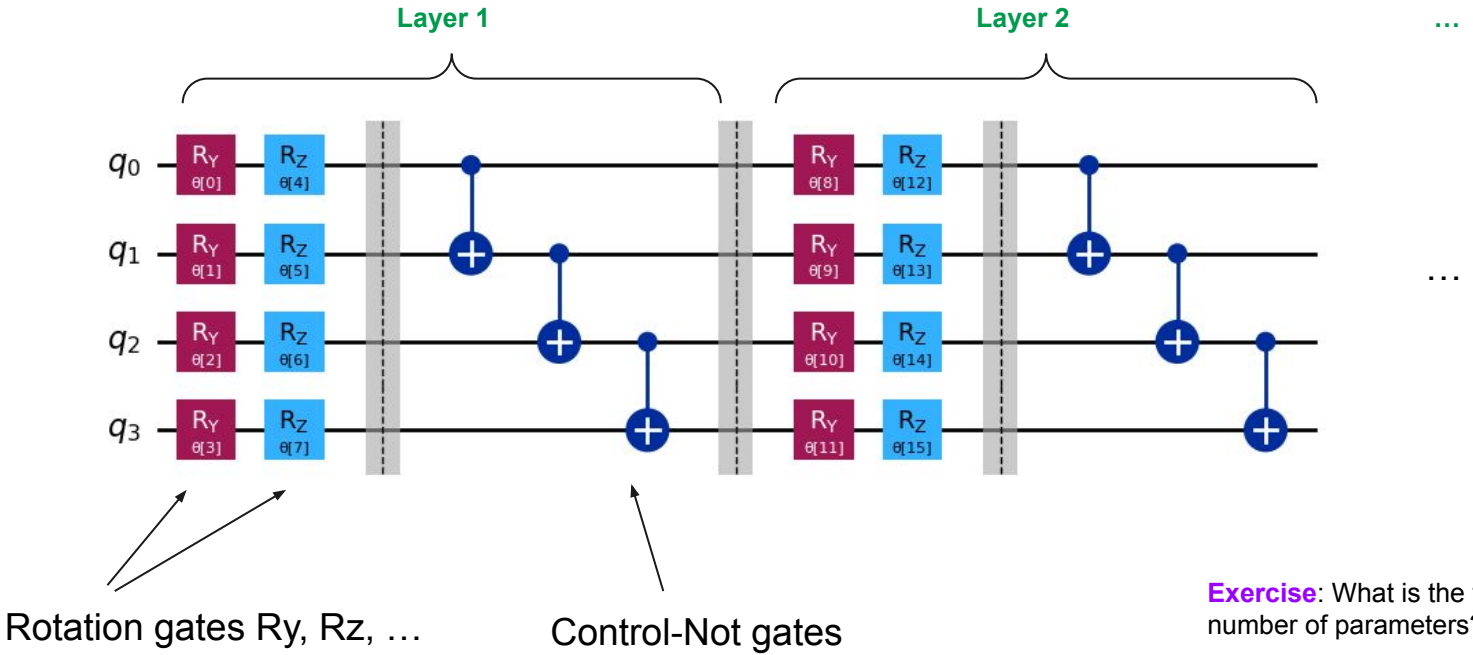
Here we focus on two types of VQAs in this lecture:

- Variational Quantum Eigensolver (VQE)
- Quantum Approximation Optimization Algorithm (QAOA)

Specifically, we use **parameterized** ansatz to perform unitary evolution.

Variational Quantum Eigensolver

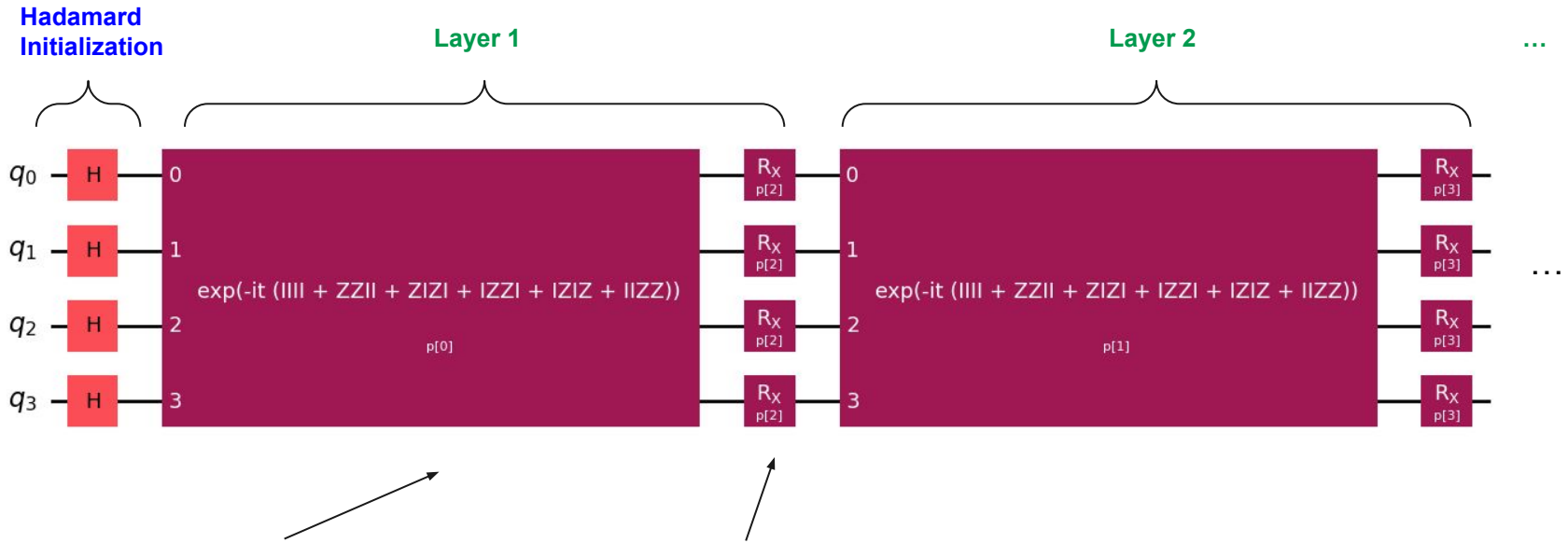
VQE is the simplest variational algorithm that typically uses heuristic, hardware-efficient ansatz that consists of **many layers** of **Rotation gates** and **control-Not gates**.



Exercise: What is the total number of parameters?

Quantum Approximation Optimization Algorithm

Unlike VQE, QAOA is a problem-inspired ansatz that consists of **many layers** of **cost operators** and **mixer operators**. It usually requires a Hadamard initialization

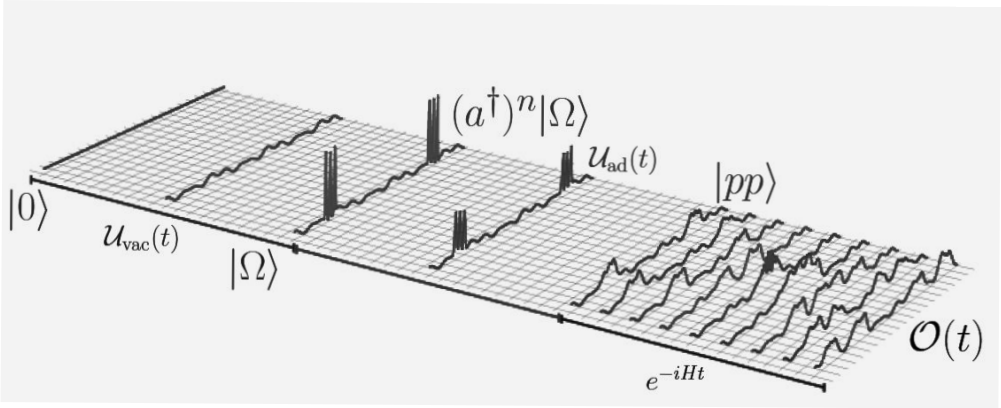


Exercise: What is the total number of parameters?

Quantum simulation V

Wiesner, 9603028 (1996); Zalka, 9603026 (1996)

1. Define problem of interest
2. Encode onto basis
3. Prepare initial states
4. Variational ansatz
5. **Measurement protocol**



Lamm's talk at Fermilab (2021)

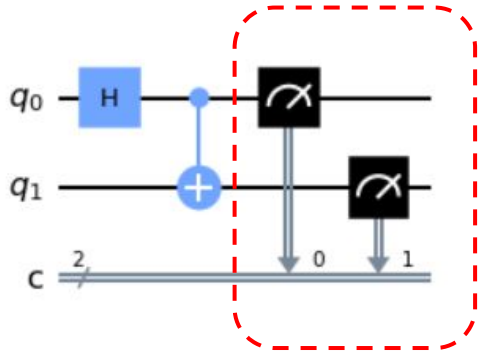
What is measurement and how to compute observable expectation?

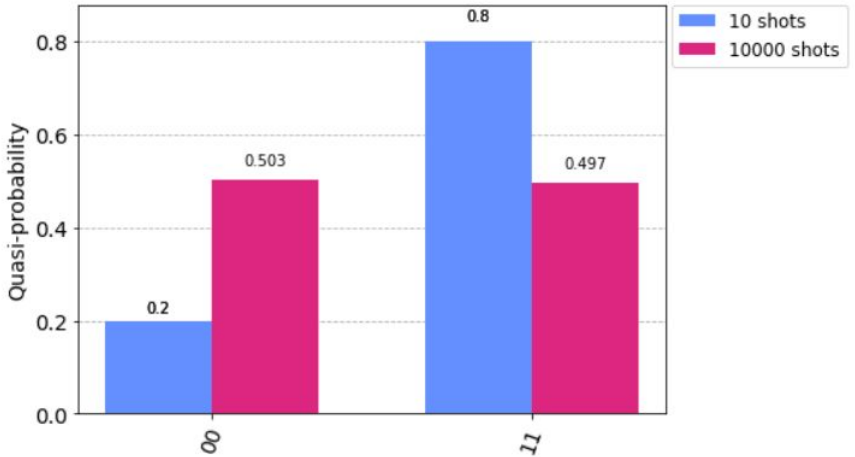
What is measurement?

Measurement is the result of quantum simulation

From quantum mechanics, measurement collapse the wavefunction (in a sense, we lost the full information of the state)

Therefore, we must measure/sample the circuit many times (shots) to partially reconstruct the state





Observable

- What is observable?

  Quantity of interests. For example, energy of the state.

- How are observables constructed?

  Formally, it is the expectation value of an operator $\langle O \rangle = \langle \psi | \hat{O} | \psi \rangle$

  On the quantum circuit, it is computed as a certain weighted sum of probabilities in the computational basis.

Expectation of Observable

For example, $\hat{O}_1 = \frac{1}{2} I_0 Z_1$ $\quad |\psi\rangle = \sum_i c_i |i\rangle = c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle$

Then $\hat{O}_1 |\psi\rangle = \frac{1}{2}\Big(c_{00} IZ|00\rangle + c_{01} IZ|01\rangle + c_{10} IZ|10\rangle + c_{11} IZ|11\rangle\Big)$

$= \frac{1}{2}\Big(c_{00}|00\rangle - c_{01}|01\rangle + c_{10}|10\rangle - c_{11}|11\rangle\Big)$

$Z|0\rangle = |0\rangle$

$Z|1\rangle = -|1\rangle$

Obviously, this generalizes to any linear combinations of Pauli-Z observables

$\langle\psi|\hat{O}_1|\psi\rangle = \frac{1}{2}\big(|c_{00}|^2 - |c_{01}|^2 + |c_{10}|^2 - |c_{11}|^2\big)$

$= \frac{1}{2}\big(\mathrm{P}(00) - \mathrm{P}(01) + \mathrm{P}(10) - \mathrm{P}(11)\big)$

linear combo of probabilities

Expectation of Observable

What about Pauli-X, Pauli-Y based observables?

Just like the simulation problem, the trick is to append quantum gates that move the calculations to computational basis!

$$X = H^\dagger Z H, \quad Y = H_y^\dagger Z H_y \qquad H_y = HS^\dagger$$

For example,

$$\begin{aligned}
\langle\psi|\, I_0 X_1 Y_2 Z_3 \,|\psi\rangle &= \langle\psi|\, I_0 (H^\dagger Z H)_1 (H_y^\dagger Z H_y)_2 Z_3 \,|\psi\rangle \\
&= \langle\psi| H_1 (H_y)_2 |\, I_0 Z_1 Z_2 Z_3 \,| H_1 (H_y)_2 |\psi\rangle \\
&= \langle\psi'|\, I_0 Z_1 Z_2 Z_3 \,|\psi'\rangle
\end{aligned}$$

We see that any Pauli observables are just the same as Pauli-Z based observables by appending basis transformation gates. Luckily, Qiskit will do the work automatically for you.

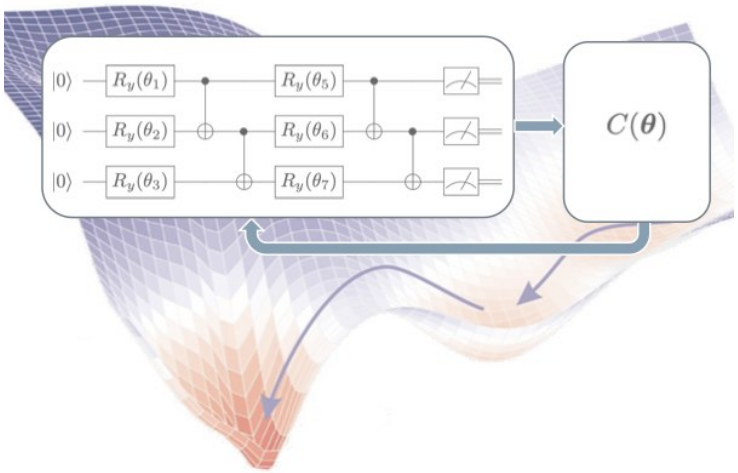
Classical Optimization

Once the quantum simulation is completed, the classical optimization kicks in

The classical tasks are:

1. Evaluate the cost function (in our case is trivially the Hamiltonian expectation)

2. Optimize the parameters with classical optimizer.



Classical Optimization

In general, optimizers can be divided into two categories depend on whether they use gradients

1. Gradient inspired optimizers

examples:
- Limited-memory BFGS Bound optimizer (BFGS)
- Sequential Least SQuares Programming optimizer (SLSQP)

Faster Convergence
Efficient in Noiseless setting

Sensitive to Noise
Require Gradients

2. Gradient-free optimizers

examples:
- Simultaneous Perturbation Stochastic Approximation (SPSA)
- Constrained Optimization BY Linear Approximations (COBYLA)

Easy to Implement
Robust to Noise

Slower Convergence

Both can be subject to Barren Plateaus with High Dimension. Quantum Optimization is still open question!

Summary

- Today we learned about Variational Quantum Algorithm is efficient tool to solve real-world problems
- Specifically we focus on two algorithms: VQE and QAOA
- They are Hybrid algorithms that use quantum computer for unitary evolution and classical computer for parameter optimization.
- In the lab, we will develop these algorithms to solve Maxcut problems!

Thanks for your attention!