Gate-model Quantum Computing.

Software Stack

A famous NP (non-deterministic) hard problem

↑

Problem Definition e.g. traveling salesman problem

↓

Quantum Algorithm e.g. QAOA — Quantum Approximate optimization algorithm

↓

Quantum circuit gates & unitary operators

↓

Quantum Compiler → actual set of gates

→ connectivity (2 qubits are not physically connected, but have some interactions between)

↙ QPU ↘ Simulator

on laptop, we can simulate 20 to 22 qubits.

on super computer, around 50.

Then we will run out of the classical compute power.

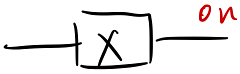
Solovay-Kitaev theorem ⇒ Finite set of gates can approximate any unitary operation.

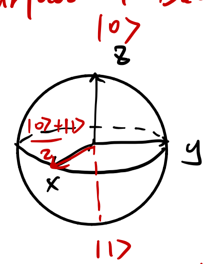
The gate model is universal, because it can transform any quantum states/qubits into any other quantum gates/qubits.

Quantum Circuits

A single-qubit gate moves a point on the surface of Bloch Sphere.

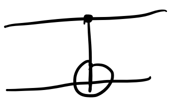
① X-gate $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$





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

③ CNOT gate $\begin{bmatrix} \boxed{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}} & 0 \\ 0 & \boxed{\begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix}} \end{bmatrix}$ → when control qubit is 1, then applies NOT.

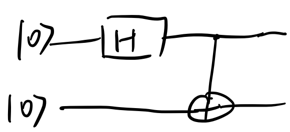


when control qubit is 0, do nothing

$CNOT|00\rangle = |00\rangle$

$CNOT|01\rangle = |11\rangle$

Create $|\phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$:



$\Big\} \frac{|00\rangle + |11\rangle}{\sqrt{2}}$

Quantum Annealing for Optimization.

1. Adiabatic Quantum Computing

绝热的

Unitary evaluation and the Hamiltonian

The Hamiltonian is always an Hermitian operator (it is equal to its own conjugate transpose)

Classical Ising model: $H = -\sum_{\langle ij \rangle} J_{ij} \sigma_i^z \sigma_j^z - \sum_i h_i \sigma_i^z$ → Hermitian

Energy expectation value: $\langle H \rangle = \langle \psi | H \psi \rangle$

Evolution: $U|\psi\rangle$

Schrödinger equation: $i\hbar \frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle$

$\hbar$: Plank Constant

The temperal evolution of the system is described by the Hamiltonian applied to the time-dependent state

The fact that Hamiltonian is Hermitian implies that operater U is unitary

⇓

Every Hamiltonian implies unitary operator.

Solution for time-independent H: $U = \exp(iHt/\hbar)$ → unitary

Every gate has an underlying Hamiltonian.

($\bar{A}$: conjugate of A. A^T transpose of A)

summary:
- Hermitian matrix: $\bar{A}^T = A$ ⇐ the Hamiltonians (H)
- Unitary matrix: $\bar{A}^T = A^{-1}$ ⇐ the time-evolution operater U.

The Adiabatic Theorem

2 Hamiltonians
$$\begin{cases} H_0 = \sum_i \sigma_i^x \\ H_1 = -\sum_{\langle i,j \rangle} J_{ij} \sigma_i^z \sigma_j^z - \sum_i h_i \sigma_i^z \end{cases}$$

H_0 → transverse field → The ground state, the lowest energy state of this, is equal superposition

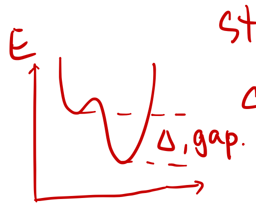
$H(t) = (1-t)H_0 + tH_1, \quad t \in [0,1]$

If we change the time t slowly, and start from the ground state of H_0, end at the ground state of H_1.

In classical Ising model, we can easily get stuck at local optimum.

↳ Solution: Adiabatic transition

Stay in ground state (lowest energy) throughout the change.



Speed limit: $\sim \frac{1}{\min(\Delta(t))^2}$, Δ: gap, diffence between the ground state and the first excited state.

For different t, we have different gap $\Delta(t)$

However, if the gap is very small, the speed limit will be very bad.

∴ It is not true to say we can solve a NP-hard problem faster or exponentially faster, because those problems have very small gap.

Adiabatic Quantum Computing

$$H = \underbrace{-\sum_{\langle ij \rangle} J_{ij}\sigma_i^z\sigma_j^z - \sum_i h_i\sigma_i^z}_{\text{classical Ising model}} \underbrace{- \sum_{\langle ij \rangle} g_{ij}\sigma_i^x\sigma_j^x}_{\text{interaction between transverse field.}}$$

(not transverse field ising model, which don't have interactions)

This is universal!

(if it's able to implement a specific Hamiltonian

2. Quantum Annealing

Adiabatic quantum Computing → idealized

Quantum Annealing → less idealized.

→ Minimum gate is hard to calculate ⇒ Anneal repeatedly
(probably violating adiabatic conditions)

pick the one with lowest energy.
(this isn't guranteed to be the ground state and the global minimum, but will give you a better result.

→ Solve classical Ising model.

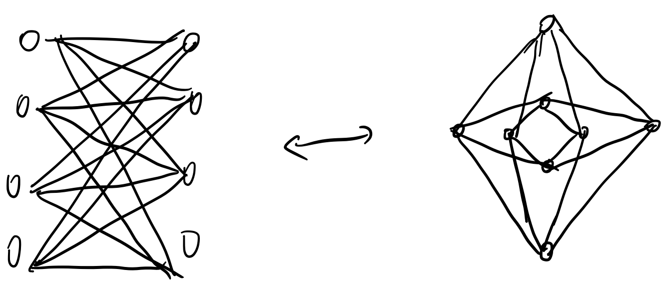
→ Experimental conditions (finite temperature, electromagnetic interference).

Software Stack

Problem Definition e.g. TSP
↓
Map to Ising model
↓
Minor Embedding → Similar to the compilation in the gate model architecture.
It maps the connectivity structure of the qubits to the hardware
↓
QPU

$\boxed{\text{Not all qubits are fully connected.}}$

Chimera graph



Problem to solve: $\underbrace{\sigma_1\sigma_2+\sigma_2\sigma_3+\sigma_1\sigma_3}_{K_3}$

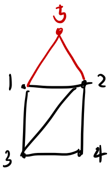
Implementations how to build a quantum computer?

Superconducting architecture

- Silicon-based
- Cooled to around 10mK
- Microwave pulses control the system → fast control speed

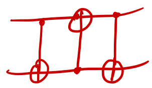
Disadvantages:

- 2D topology



good for 4 qubits.
but for a 5th qubit, it's difficult to establish between every qubit.
If we want to interact 3 and 5, a SWAP must be implemented.
and then SWAP gate.

SWAP:



∴ 7 gates for 1 gate.

- cooling requirements
- short coherence time.
  time ⇔ limited circuit depth. 10-20 gates

Trapped ions

→ operate at room temperature
→ have fully connectivity

- Individual ions (charged atomic particles) confined using electromagnetic fields.
- lasers induce coupling
- long coherence time

Disadvantages:

- Scalability unclear [unclear how well it will scale to larger systems]
- speed of control is much slower.

Photonic systems

- e.g. photon polarization
- Room temperature

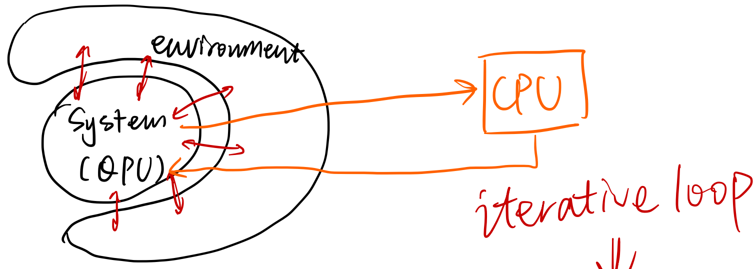
Disadvantages:

- photon loss
- photons can't be stored.

Variational Circuit

Quantum Approximate Optimization Algorithm.

Variational circuits

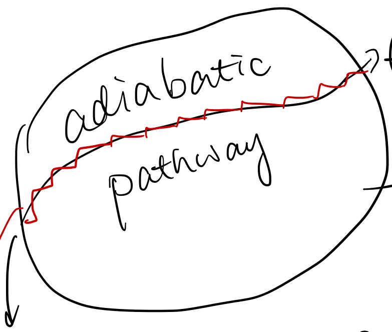


We have limited ability to control our system, and this limits our circuit depth, for instance. So what we want is to run a short burst of calculation on the quantum processing unit, extract the result to a classical CPU. and then the circuit that we ran on the quantum computer is parametrized.

By understanding the result of a calculation, we can go back, adjust the parameters of the quantum processor. and again run a short burst of calculation.

QAOA: one of the most famous variational circuits.

draws inspiration from quantum annealing, approximate the adiabatic pathway



Ground state of $H_1 = \sum_{\langle ij \rangle} J_{ij} \sigma_i^z \sigma_j^z - \sum_i h_i \sigma_i$

State space

Ground state of $H_0 = -\sum_i \sigma_i^x$

break the pathway up into discrete chunks and parameterize the circuit to have a more and more accurate approximation.

just the same as the quantum annealer, at the end we can read out the ground state

$H(t) = (1-t) H_0 + t H_1$

Discretize up to time t_0:

$$U(t_0) \approx U(H_0, \beta_0) U(H_1, \gamma_0)$$

Then: subsequent time steps.

$$U \approx U(H_0, \beta_0) U(H_1, \gamma_0) \cdots U(H_0, \beta_p) U(H_1, \gamma_p)$$

Trotterization