Introduction to Quantum Optimization and Its Applications

Andriyan B. Suksmono, Ph.D.
Professor
School of Electrical Engineering and Informatics, ITB, Bandung

Workshop on Classical and Quantum Machine Learning for Condensed Matter Physics|(smr 3948)

An Introduction to Quantum Computing

Andriyan B. Suksmono, Ph.D.
Professor
School of Electrical Engineering and Informatics, ITB, Bandung

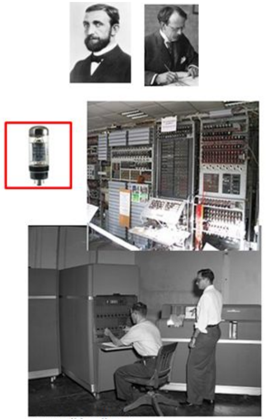
Workshop on Classical and Quantum Machine Learning for Condensed Matter Physics|(smr 3948)

Table of Contents

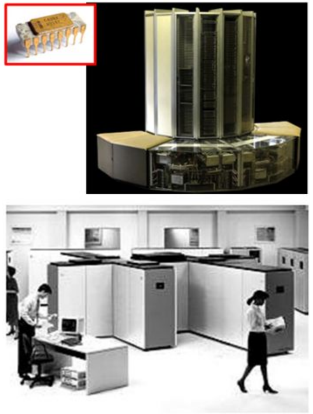
Introduction

History of Information Technology

www.wikipedia.org

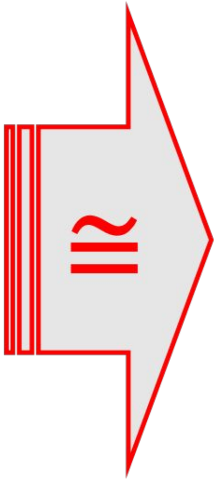


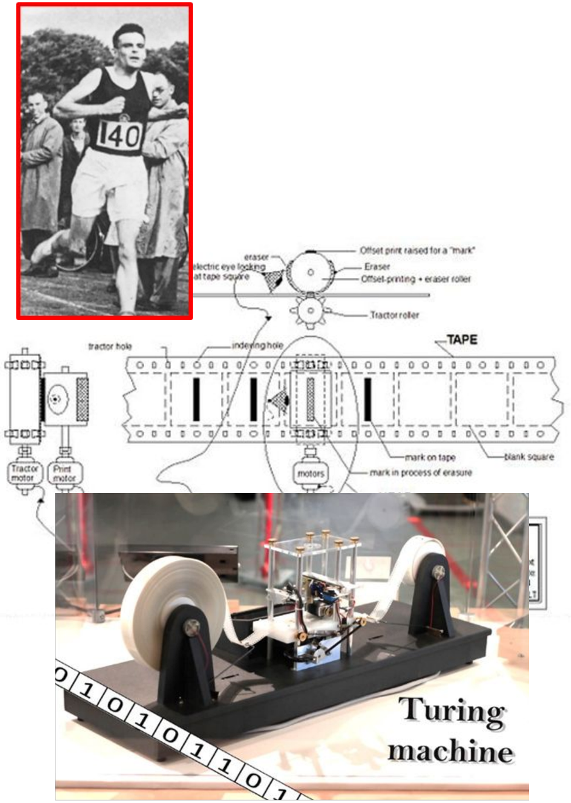
www.wikipedia.org



www.wikipedia.org

Turing Machine



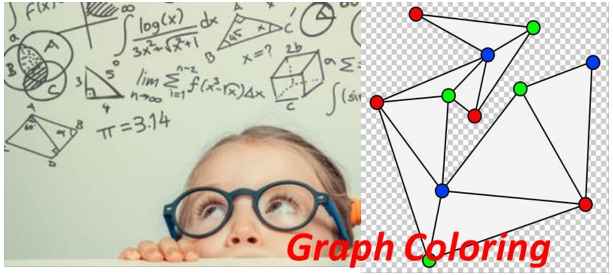


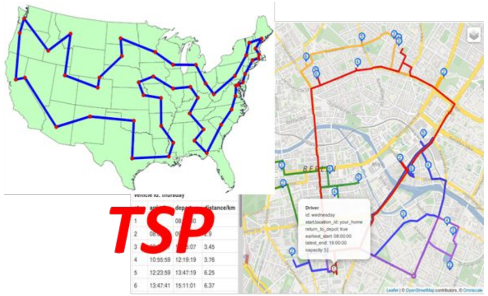
https://simple.wikipedia.org/

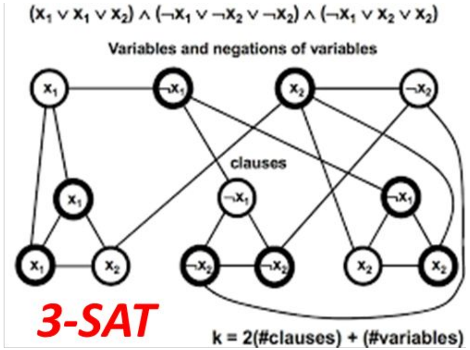
Extended Church-Turing Thesis [Bernstein-Vazirani]

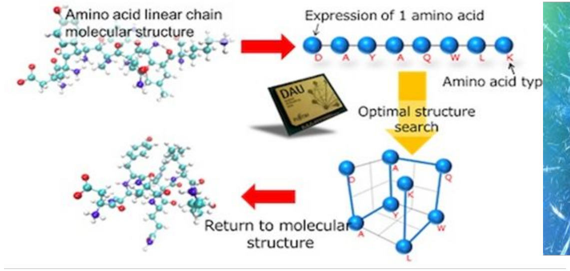
"Any realistic model of computation can be efficiently simulated by (probabilistic) Turing Machine."

Hard Problems

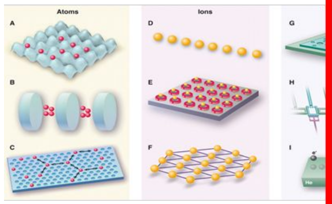




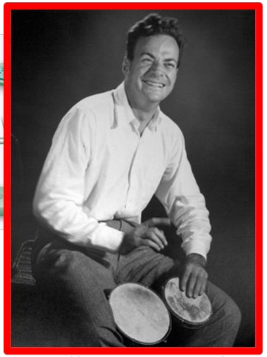




Protein Folding

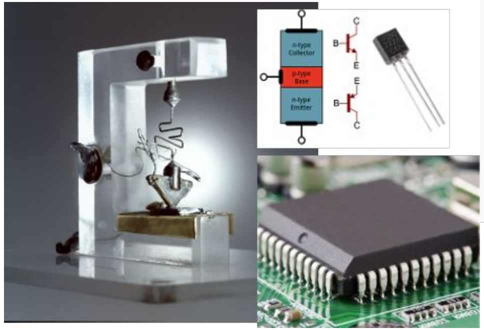


Quantum Simulation



https://cosmolearning.org/

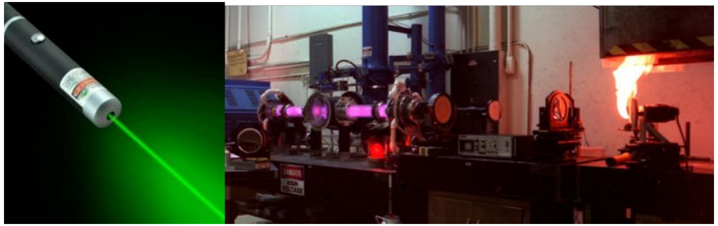
The 1st Quantum Revolution



Digital Camera

MRI

Solar Panel

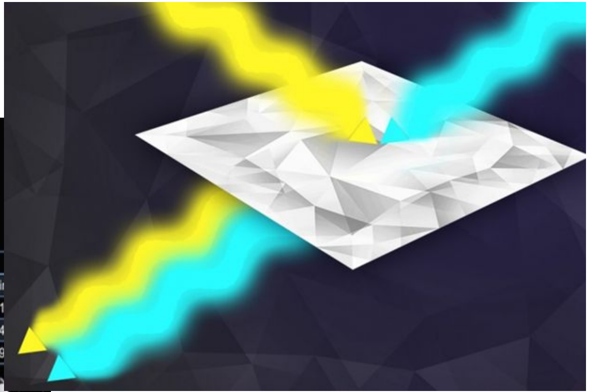


LASER

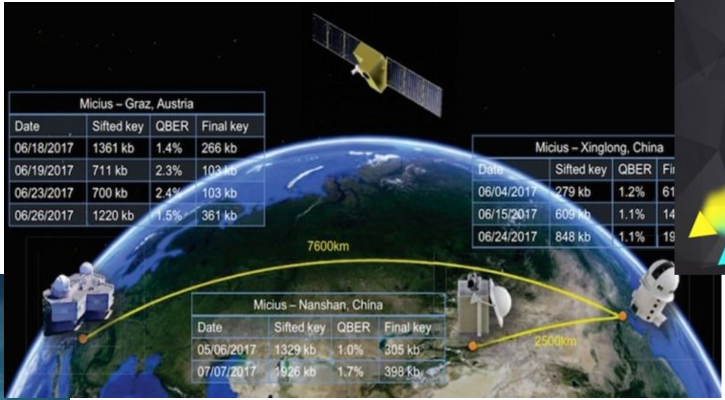
Electron Microscope

https://wikipedia.org/

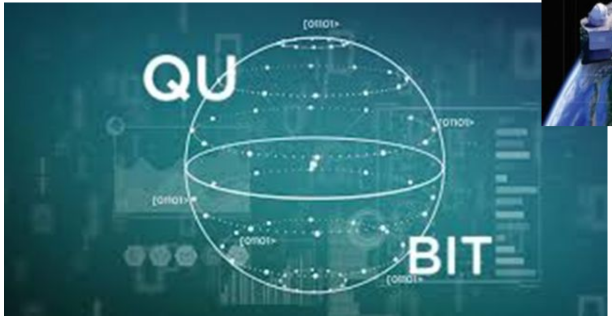
The 2nd Quantum Revolution



Quantum Sensing



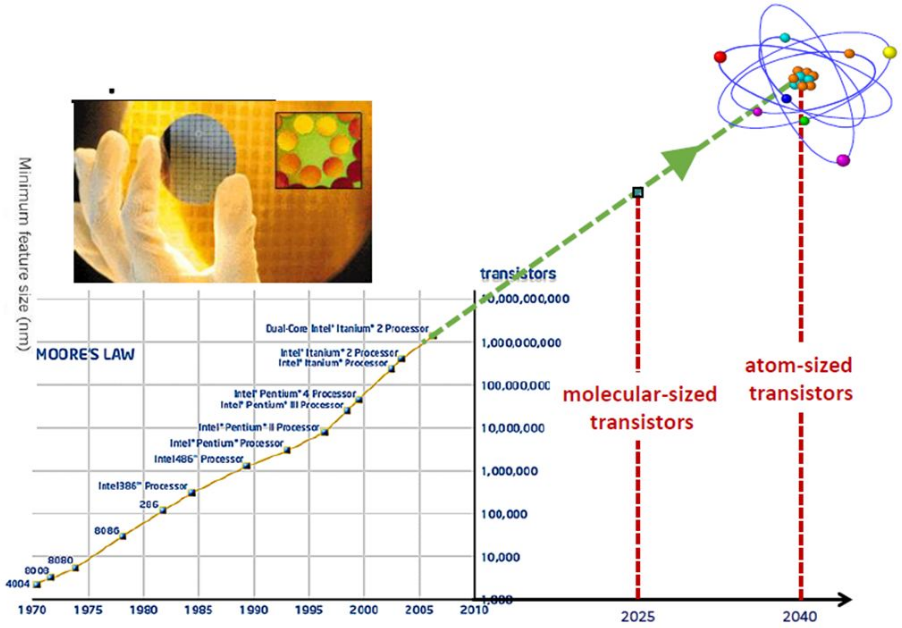
Quantum Communication



Quantum Computing

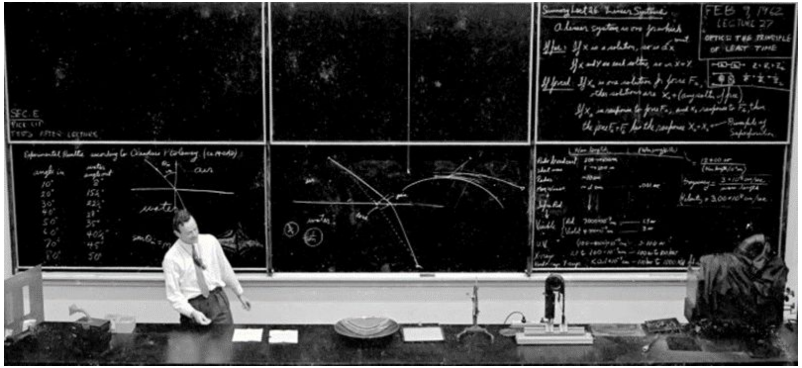
Basic Concepts of Quantum Computing

Miniaturization of Electronics and Moore Law



Moore's Law: the number of transistor in an IC doubled every 2 years [GE Moore, *Electronics,* 8, 114-117, 1965]

- **"There's Plenty of Room at the Bottom" (1959)**
- "When we get to the very, very small world –say circuits of seven atoms –we have a lot of new things that would happen that represent **completely new opportunities for design**.
- *Atoms on a small scale behave like nothing on a large scale, for they satisfy the laws of quantum mechanics…"* [Richard Feynman]



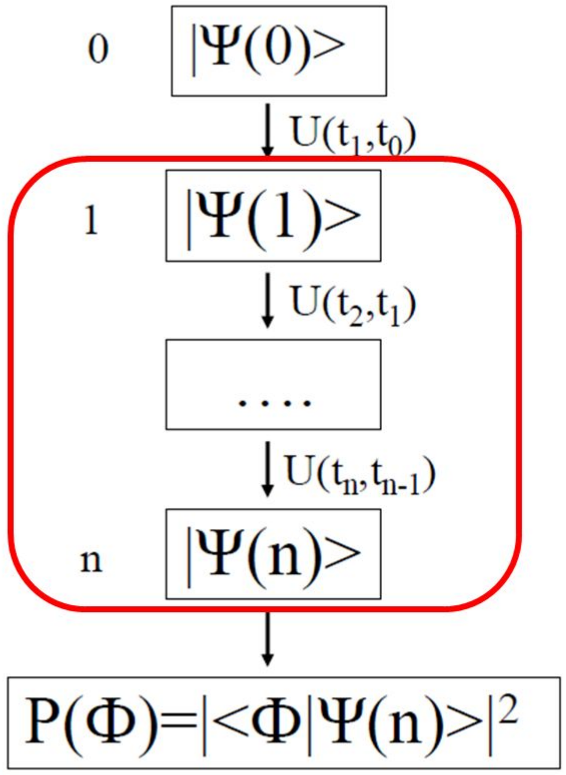
Classical vs Quantum Computers

- **Classical Computers**
  - Computer that uses electrical voltage or current that flows into circuit and logic gates.
  - It is governed by the Law of Classical Physics
- **Quantum Computer**
  - Computers that works using the principle of **Quantum Physics** to perform computation in a parallel fashion
  - Employing the principle of superposition, entanglement, and coherence.

… in essence

Computing is a Physical Process

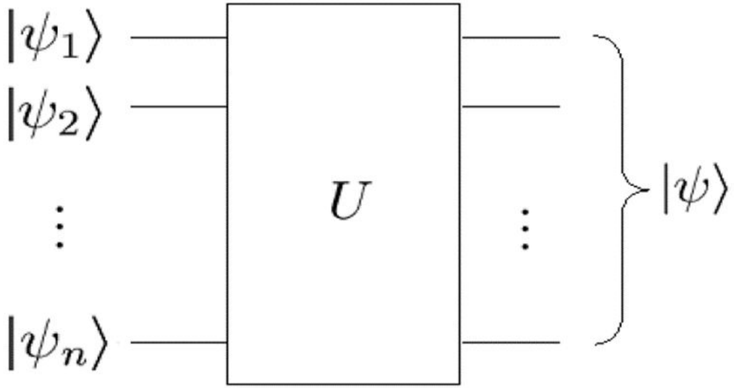
Unitary Transform as Quantum Computation



Steps in Quantum Computing

- PREPARATION
  - Preparation of initial sate of a quantum system at t=0
- EVOLUTION
  - Sequential evolution of the system by unitary transforms
- MEASUREMENT
  - The measurement make the system collapse to classical state.

… contd.



- In a Quantum Computer
  - A quantum program is executed by unitary transforms of a set of quantum states $|y_n>$. It can be state “0”, state “1” or superposition of these states.
  - Unitary transform is invertible, which means that it can be *uncomputed*

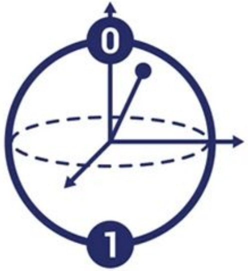
Quantum Computers

Chinese 76-qubit photon-based QC

IonQ, ion-trap-based 32-qubit QC

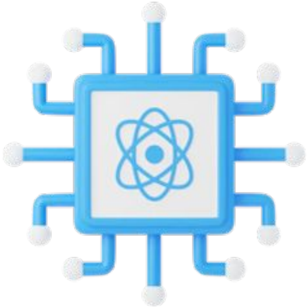
IBM 53-qubit superconductor-based QC

Qubit

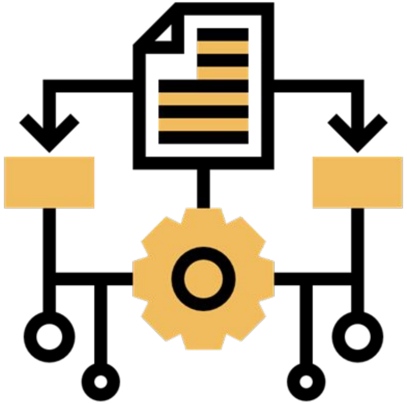


Gates

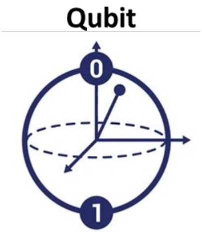
Circuit



Qubit, Circuit, Gates, and Algorithm



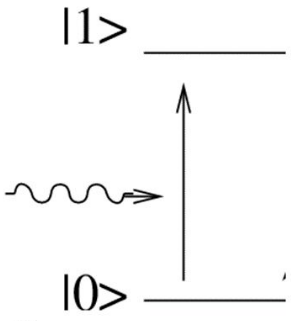
Algorithm

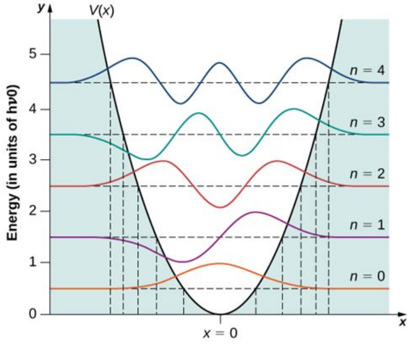


Qubit

What is qubit ?

- Smallest unit of information in quantum computing.
- The state of wave function |y> of the Schroedinger Equation.
- Can be “0”/ON, “1”/OFF, or superposition of “0” and “1”
- Qubit Realization:
  - Nuclear Spin in the NMR
  - Superconducting quantum circuit (transmon)
  - Photon in a *cavity*
  - Energy of an atom: ground state, excited state
  - Photon Polarization … etc





… contdr.

- Follows linearity principle (from Schrödinger Equation)
- General form

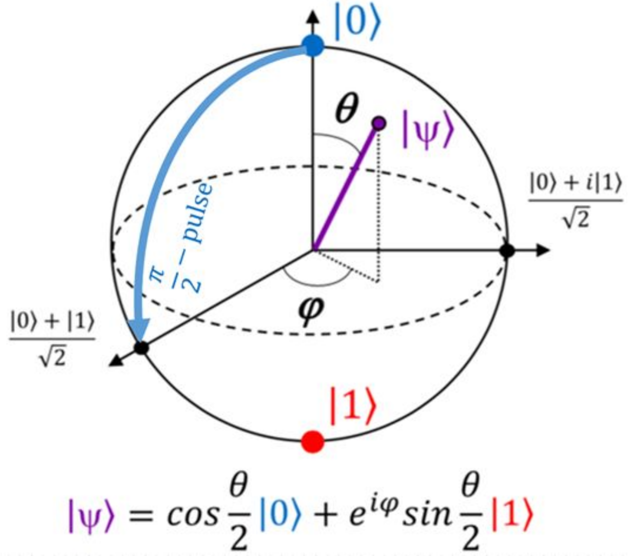
$|y\rangle = a_0|0\rangle + a_1|1\rangle$

where a_0 and a_1 are complex number, representing probability amplitude

$|a_0|^2$: probability of state $|0\rangle$

$|a_1|^2$: probability of state $|1\rangle$

Normalization: $|a_0|^2 + |a_1|^2 = 1$



Representation of qubit as A Bloch Sphere

Bit vs Qubit

- BIT (Classical)
  - Can be in two states: either “0” and “1”
  - The states can be exactly determined.
  - Measurement does not alter the states.
  - Can be copied (cloned)

- QUBIT (Quantum)
  - Can be in state |0>, |1>,or linear combination of state |0> and |1>.
  - The states can be determined with a particular probability.
  - States changed after measurement (collapse to “0” or “1”)
  - Cannot be copied (no-cloning theorem).

Advantage of qubits

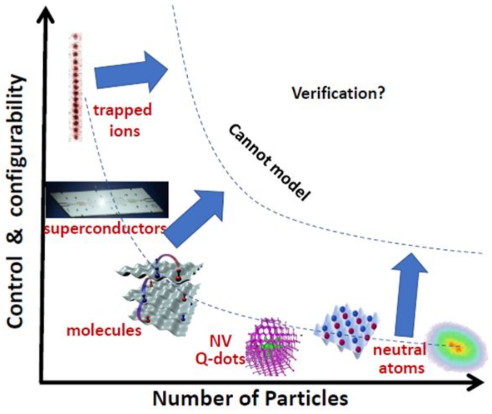
- The capacity increased exponentially
  - Increasing one qubit, the capacity is increased twice
  - Classical: the capability increased twice when 32 bits -> 64 bit
  - Quantum: capability increased when 32 qubits -> 33 qubits
- •Operation is done to all qubit superposition, likes a parallel computer
  - **Classical**: 64-bit processor can do operation to 64-bit binary number at a time
  - **Quantum**: 64-qubit processor works on 2^{64} states of binary number at a time, or

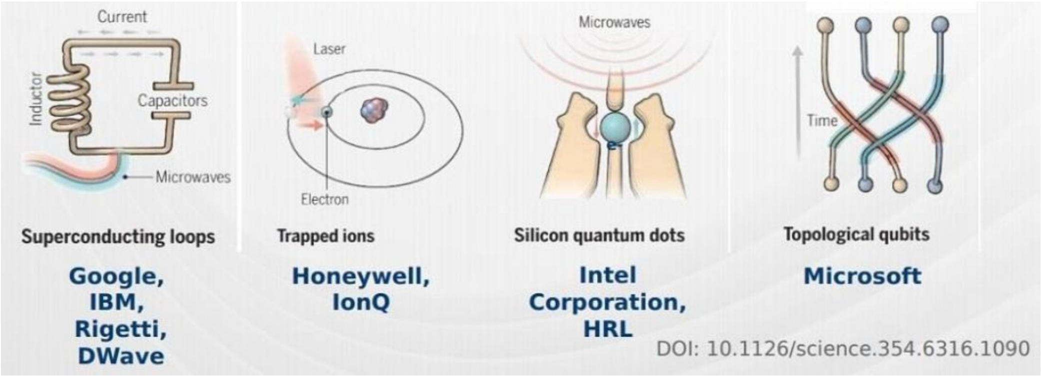
    16.000.000.000.000.000.000 = 16 x 10^{18} states

=> (a few) *hard problems* solved more easily using quantum computer

Qubit Technology

- Liquid-state NMR
- NMR spin lattices
- Linear ion-trap spectroscopy
- Neutral-atom optical lattices
- Cavity QED + atom
- Linear optics
- Nitrogen vacancies in diamond
- Electrons in liquid He
- Superconducting Josephson junctions
- charge qubits; flux qubits; phase qubits
- Quantum Hall qubits
- Coupled quantum dots
- spin, charge, excitons
- Spin spectroscopies, impurities in Semiconductor



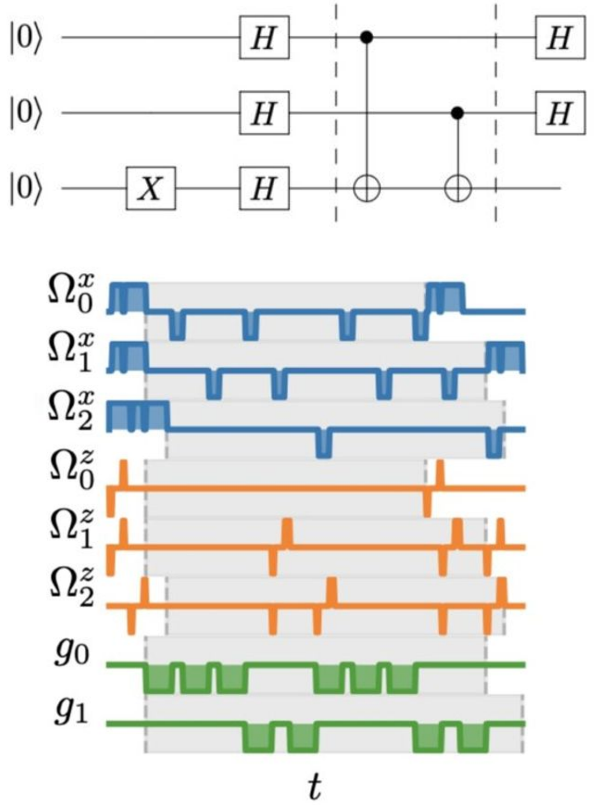


Gates

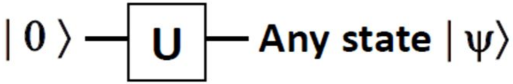
Quantum Gates

Quantum Gates

- A Quantum gate/quantum logic gate is a basic quantum circuit operating on a qubit.
- Basic building block of a quantum circuit, like a conventional logic gate of digital circuits.
- Can be expressed as a matrix U of $2^n x 2^n$ dimension. U is a unitary matrix: U*U=I.
- The number of output is equal to the number of input.
- Quantum gates are reversible.
- In reality, quantum gates realized as sequence of EM pulse at particular frequency, duration, and sequence.



Single Qubit Gates



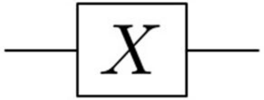
- **Pauli-X gate**

$$|0\rangle \rightarrow |1\rangle, \quad |1\rangle \rightarrow |0\rangle$$

Dirac notation

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Matrix representation

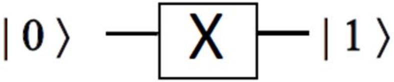


Circuit representation

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow X\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot 1 + 1 \cdot 0 \\ 1 \cdot 1 + 0 \cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow X\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \cdot 0 + 1 \cdot 1 \\ 1 \cdot 0 + 0 \cdot 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$$

- Works like the “NOT” gate



…Contd

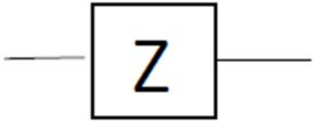
Name | Matrix Representation | Circuit Representation

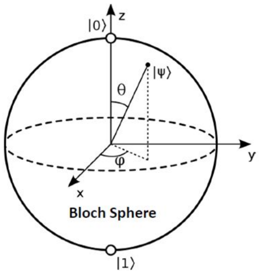
Pauli Y – Gate

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

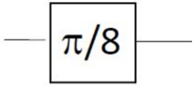
Pauli Z – Gates:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$





$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$



π/8 (T) gate

Hadamard Gate

Dirac notation

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$(|0\rangle + |1\rangle)/\sqrt{2} \rightarrow |0\rangle$$

$$(|0\rangle - |1\rangle)/\sqrt{2} \rightarrow |1\rangle$$

Unitary matrix

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

Circuit representation

no classical equivalent!

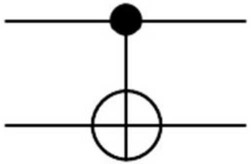
CNOT Gate

- A Controlled-NOT gate
- Act to 2 qubits
  - qubit control 0, target qubit not change
  - qubit control 1, target qubit inverted

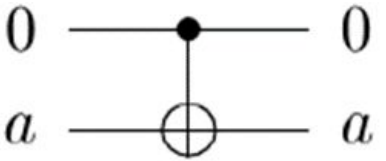
Matrix representation

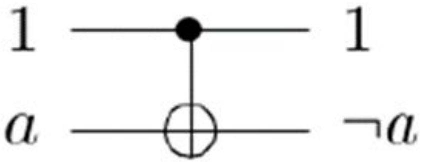
$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Circuit representation

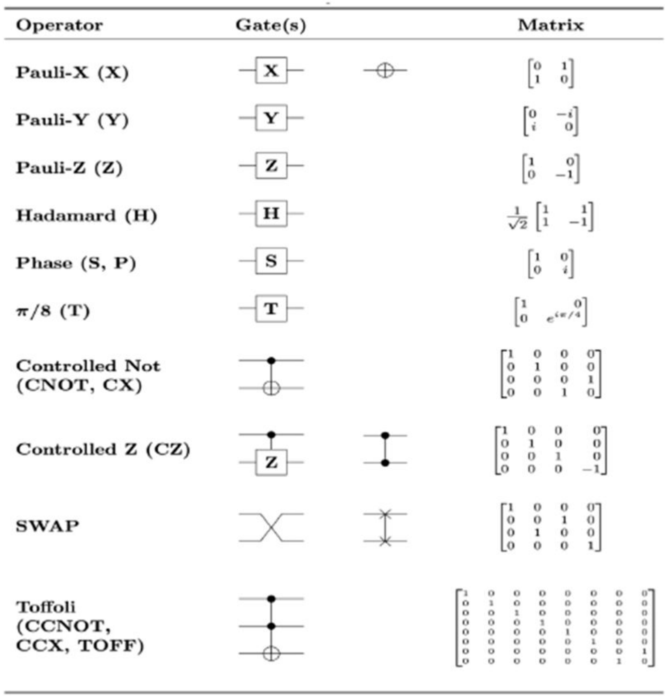


- Like XOR of a classical gate

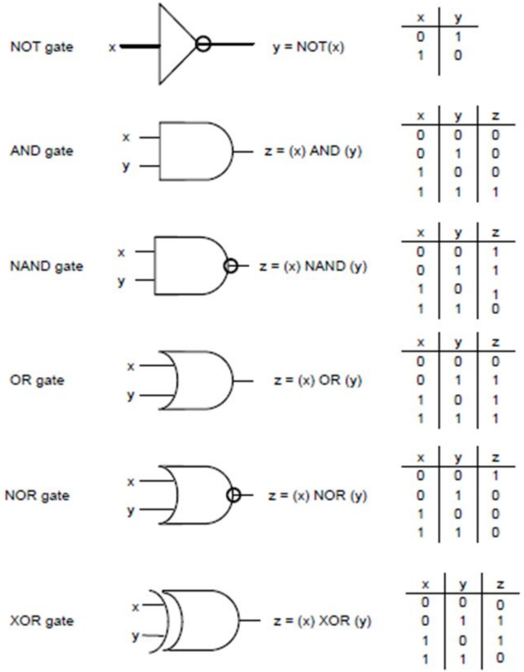




Quantum VS Classical Gates

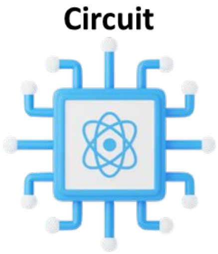


Operator	Gate(s)	Matrix
Pauli-X (X)	X	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$
Pauli-Y (Y)	Y	$\begin{bmatrix}0 & -i\\i & 0\end{bmatrix}$
Pauli-Z (Z)	Z	$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
Hadamard (H)	H	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$
Phase (S, P)	S	$\begin{bmatrix}1 & 0\\0 & i\end{bmatrix}$
$\pi/8$ (T)	T	$\begin{bmatrix}1 & 0\\0 & e^{i\pi/4}\end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$
Controlled Z (CZ)	Z	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{bmatrix}$
SWAP		$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{bmatrix}$



x	y
0	1
1	0

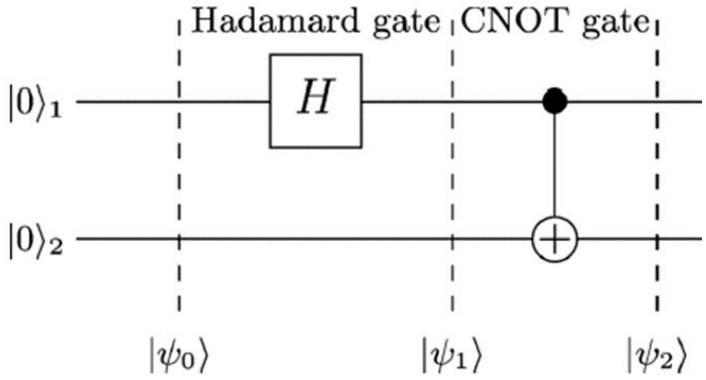
x	y	z (AND)	z (NAND)	z (OR)	z (NOR)	z (XOR)
0	0	0	1	0	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	0	1	0	0

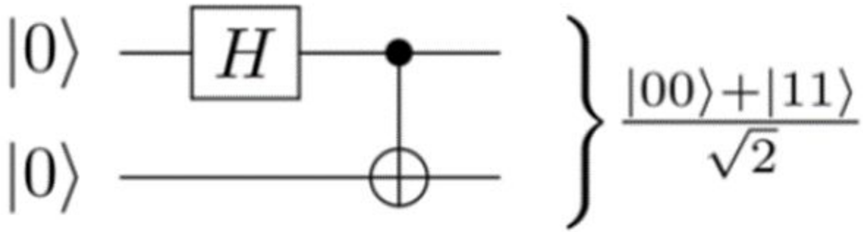


Quantum Circuits

Bell-State

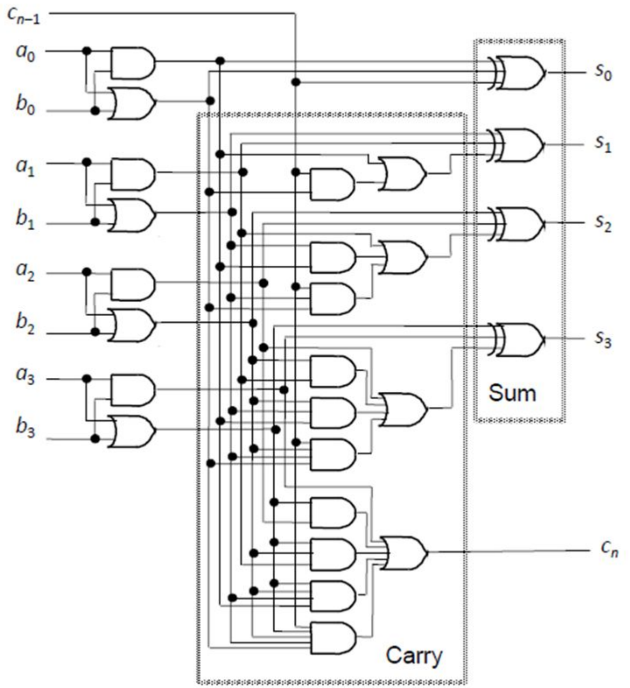
- Generate 2-qubits entangled-state
- Also called as Bell state,
- Employs a Hadamard and a CNOT gates





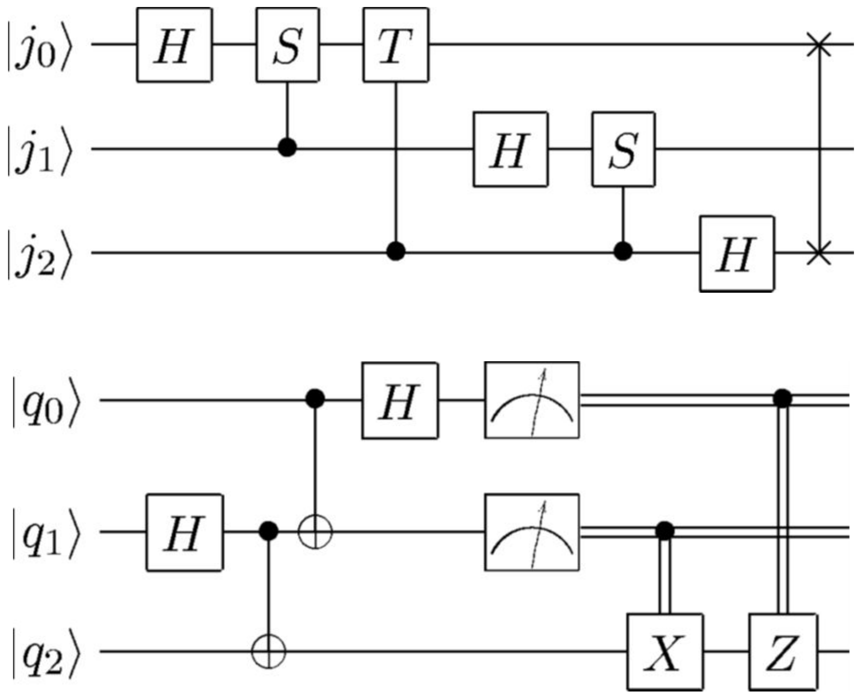
Classical VS Quantum Circuits: Classical

- Classical Logic Gates
  - Obey Classical Physics
  - State in the form of bit vectors, eg X="011101"
  - Operation defined by Boole Algebra
  - No restriction on copying and measuring the state/data



Classical VS Quantum Circuits: Quantum

- A model of quantum computing where the computation process is expressed as sequence of quantum gates with n-registers and connected with wires.
- The width of a quantum circuit is constant, equal to the number of qubits involved.

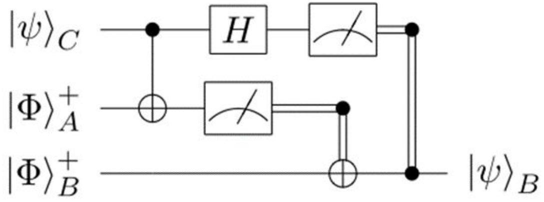


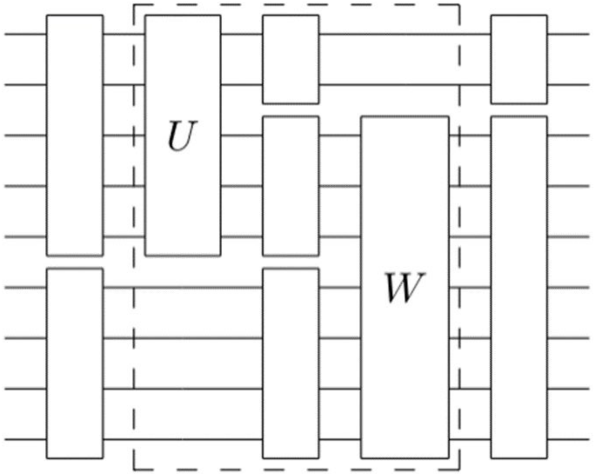
Classical VS Quantum Circuits

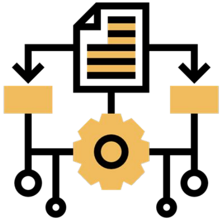
- Quantum Circuits
  - Obey quantum mechanics
  - State vector is superposition of qubits with complex coefficient

$$|\Psi\rangle = \sum_{i=0}^{2^n-1} c_i |i_{n-1} i_{n-1} \dots i_0\rangle$$

  - Operation can be defined as Linear Algebra on Hilbert Space and can be represented as unitary matrix with complex elements







Quantum Algorithms and Programming Quantum Computers

Quantum Algorithm

- Some particular problem can be more efficiently solve by quantum computing than the classical one (Turing Machine based)
- Examples: integer factorization (Shor), search in unstructured database (Grover)
- Wide impacts on cryptography and security
- Why quantum computer is faster?
  - **Parallelism:** using superposition of quantum states, algorithm execute in parallel
  - **Hilbert Space dimension:** exponentially growing
  - **Entanglement:** different qubits can be entangled, giving non-classical correlation.

Examples of Quantum Algorithms

Algorithms	Classical steps	quantum logic steps
Fourier transform e.g.: - Shor's prime factorization - discrete logarithm problem - Deutsch Jozsa algorithm	$N\log(N) = n2^n$ $N = 2^n$ - n qubits - N numbers	$\log^2(N) = n^2$ - hidden information! - Wave function collapse prevents us from directly accessing the information
Search Algorithms	N	$\sqrt{N}$
Quantum Simulation	c^N bits	kn qubits

Programming a Quantum Computer

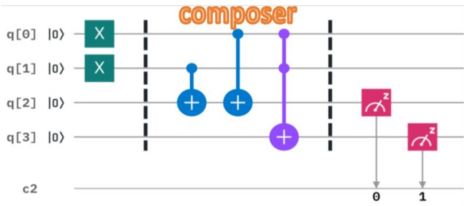
- A few languages available to implement quantum computing
  - QCL (Quantum Computation Language): syntax similar to C
  - Qiskit: Python-based, SDK developed by IBM. Some components:
    - Qiskit Terra, Qiskit Aer, Ignis, Aqua
    - Qiskit Optimization, Finance, Machine Learning, Nature, Pulse, … dst
- Tools used by researchers, not developer.

```
qureg x1[2]; // 2-qubit quantum register x1
qureg x2[2]; // 2-qubit quantum register x2
H(x1); // Hadamard operation on x1
H(x2[1]); // Hadamard operation on the first qubit of the register x2
```

```python
from qiskit import QuantumCircuit

qc = QuantumCircuit(2, 2)

qc.h(0)
qc.cx(0, 1)
qc.measure([0,1], [0,1])
```

Example: Qiskit

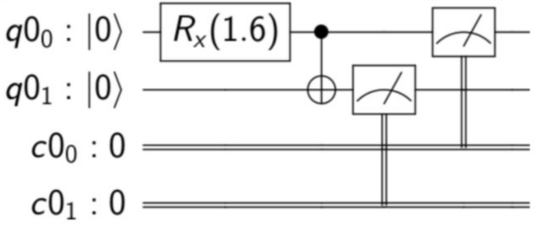
- QISKit: **Q**uantum **I**nformation **S**cience Kit
- Open Source framework for quantum computing.
- Provides tools for creating, manipulating quantum program, and run it on a (prototype) quantum processors of IBM Quantum experience over Cloud-Based Access.



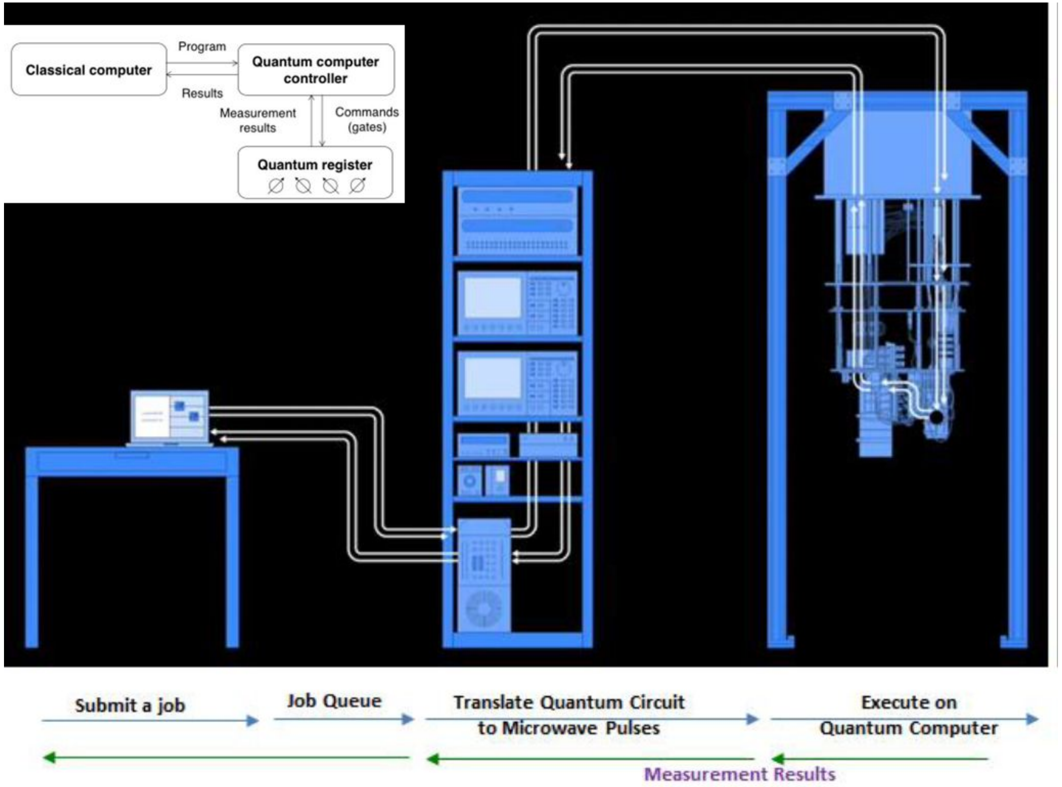

```
In [1]: from qiskit import *

In [2]: qr = QuantumRegister(2)
        cr = ClassicalRegister(2)

In [3]: c = QuantumCircuit(qr, cr)    # c = QuantumCircuit(2,2)
```

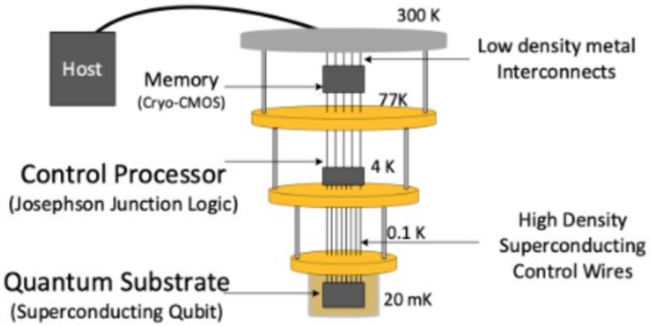
```
In [7]: from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
        from qiskit.tools.visualization import circuit_drawer
        import numpy as np

        qr = QuantumRegister(2)
        cr = ClassicalRegister(2)
        qp = QuantumCircuit(qr,cr)

        qp.rx( np.pi/2,qr[0])
        qp.cx(qr[0],qr[1])

        qp.measure(qr,cr)

        circuit_drawer(qp)
```

Out[7]:



Flow of instructions in a Quantum Computer





End of Section

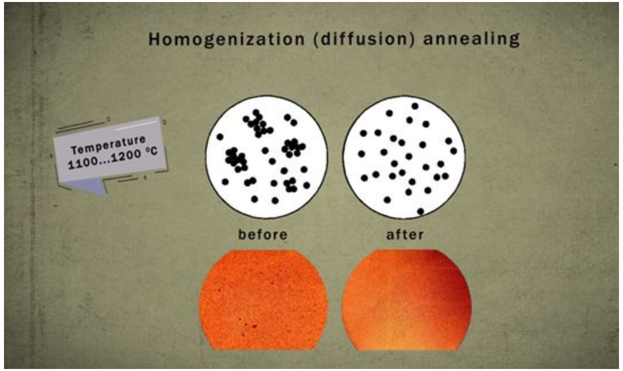
Quantum Annealing

Andriyan B. Suksmono, Ph.D.
Professor
School of Electrical Engineering and Informatics, ITB, Bandung

Workshop on Classical and Quantum Machine Learning for Condensed Matter Physics|(smr 3948)

Introduction

- **What is annealing?**
  - A terminology in metallurgy and material science
    - A **heat treatment** process that changes the physical (and sometimes also the chemical) properties of a material to **increase ductility** and **reduce the hardness** to make it more workable,
      - The annealing process requires the material **above its recrystallization** temperature for a set amount of time before cooling.
    - Or: a technique involving **heating and controlled cooling** of a material to alter its physical properties.
- **Optimization by annealing**
  - An optimization procedure/algorithm inspired by the annealing process to get the best/ optimal points/ solution of a problem.



https://en.wikipedia.org/wiki/Annealing_(materials_science)

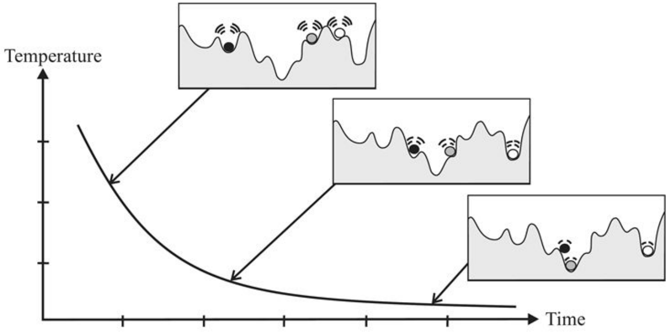
https://www.youtube.com/watch?v=rRJWuj6kauA

Classical/Thermal/ Simulated Annealing (SA)

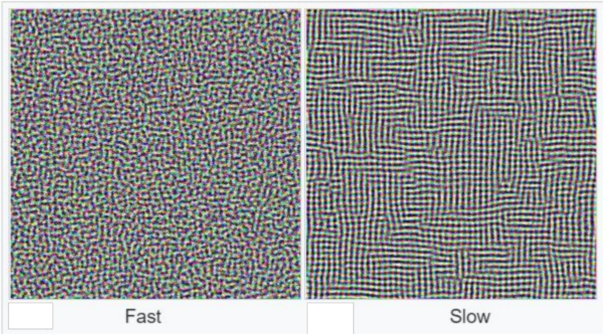
- SA is a probabilistic technique for approximating the global optimum of a given function.
    - Specifically, it is a metaheuristic to approximate global optimization in a large search space for an optimization problem.

- Let $s = s_0$
- For $k = 0$ through k_{max} (exclusive):
    - $T \leftarrow$ temperature(1 - $(k+1)/k_{max}$)
    - Pick a random neighbour, $s_{new} \leftarrow$ neighbour(s)
    - If $P(E(s), E(s_{new}), T) \geq$ random(0, 1):
        - $s \leftarrow s_{new}$
- Output: the final state s

https://en.wikipedia.org/wiki/Simulated_annealing



The annealing schedule

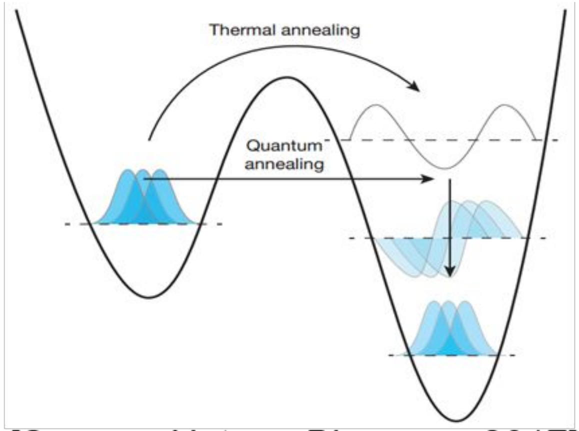


https://www.intechopen.com/books/3003

Quantum Annealing (QA)

$$\hat{H}_{pot}(\hat{\sigma}) \equiv -\sum J_{ij}\hat{\sigma}_i^z\hat{\sigma}_j^z - \sum_i h_i\hat{\sigma}_i^z$$

$$\hat{H}_{kin}(\hat{\sigma}) \equiv -\Gamma\sum_i \hat{\sigma}_i^x$$



[Source: *Nature*, Biamonte 2017]

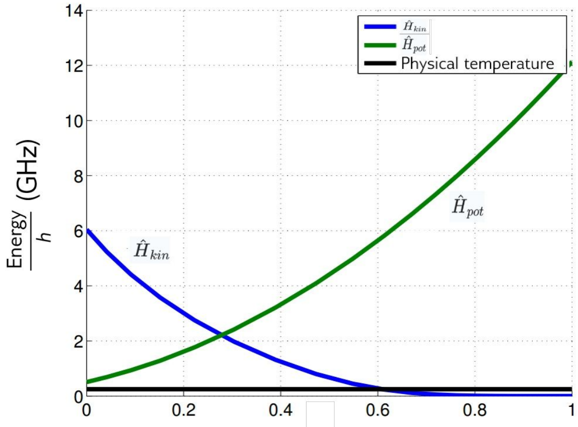
Quantum Annealer

Source: D-Wave Systems

$$\hat{H}_{QA}(\hat{\sigma}, t) = \left(1 - \frac{t}{\tau}\right)\hat{H}_{kin}(\hat{\sigma}) + \frac{t}{\tau}\hat{H}_{pot}(\hat{\sigma})$$

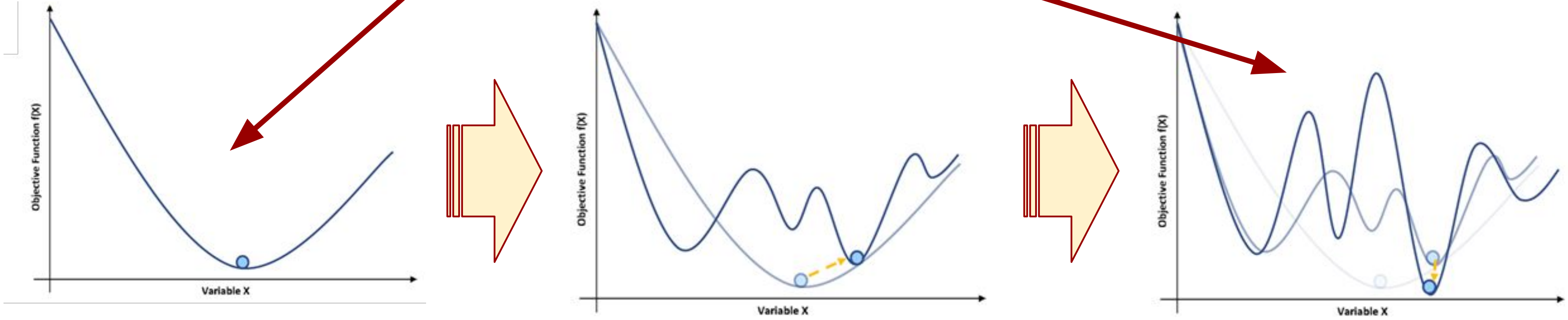
Annealing Schedule

$$\hat{H}_{QA}(\hat{\sigma}, t) = \left(1 - \frac{t}{\tau}\right)\hat{H}_{kin}(\hat{\sigma}) + \frac{t}{\tau}\hat{H}_{pot}(\hat{\sigma})$$



initial

final



D. Ottaviani, "Introduction to Quantum Annealing: Formulating and Solve QUBO Problems," PPT, *CINECA Quantum Computing Lab*.

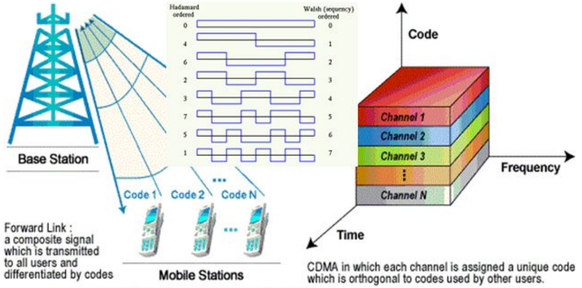
Study Case:

Finding Hadamard Matrices using Quantum Annealer

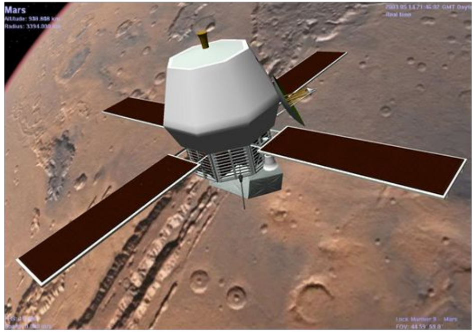
Background

- **Hadamard matrix (H-matrix)**
  Definition: an orthogonal binary {-1,1} matrix
  Applications: **orthogonal codes** used in CDMA, ECC (Error Correction Code) with maximal error correction capability, employed in Mariner-9, experiment design [Hedayat, 1973]
  Scientific/Math: H-matrix conjecture is a ~100 years old unsolved problem
- **Why finding a H-matrix is hard?**
  For an M-order matrix, there are
  $[2\text{^}(M^2)] \sim \exp(M^2)$ binary matrices
  H-matrix **conjecture** predicts, there is a H-matrix for every M=4k, k positive integer. How to find it?
  Brute force, worst-case condition: one should check all binary matrices, an $O[\exp(M^2)]$ problem --> **a hard problem**.

Proposed Solution: USE A QUANTUM COMPUTER !



CDMA Communication System employs Walsh-Hadamard Orthogonal Code



Mariner-9 employed Hadamard's ECC to protect Mars's images sent to Earth

Construction

Sylvester's Method

$$H_1 = [1],$$
$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$
$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix},$$

$$H_{2^k} = \begin{bmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{bmatrix} = H_2 \otimes H_{2^{k-1}},$$

- Sylvester's Method: only for 2^n-order H-matrices.
- Other methods for order 4k: Paley, Baumert-Hall, Williamson, …
- But, not all order (4k) can be constructed by existing methods
  - Hasn't found nor proof to exists (<2000): 668, 716, 892, 1132, 1244, 1388, 1436, 1676, 1772, 1916, 1948, and 1964.

Order of Sylvester H-matrices: 1, 2, 4, 8, 16, …. 2^k, …

Finding H-matrices by Using (Binary) Optimization

- Exploit orthogonality condition of H-matrices:

  $$H^T H = I$$

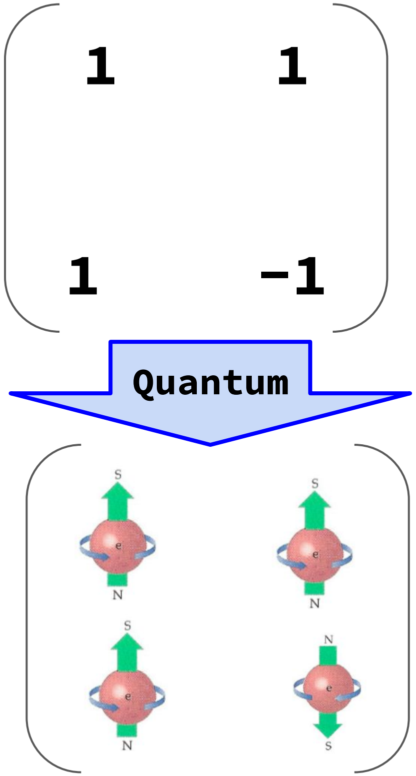
- Start with a binary matrix B, in general

  $$B^T B = D; \text{ in general } D \neq I$$

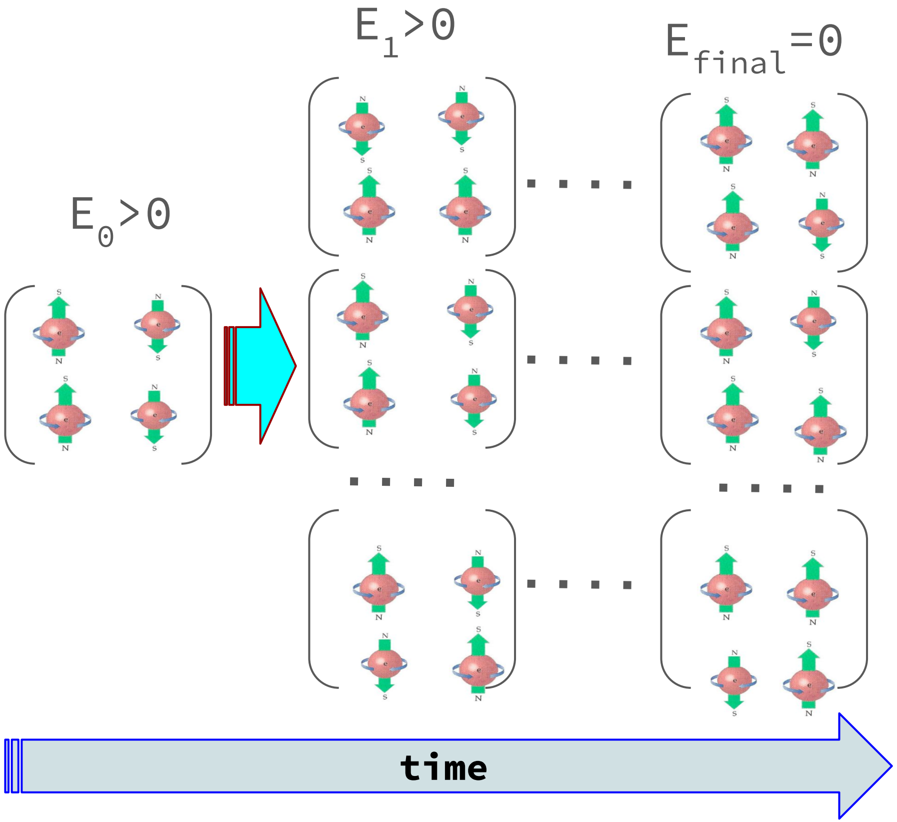
- Then, flip elements of $B=[b_{ij}]$, such that D->I.
  - Let $D=[d_{ij}]$, we define "error energy" E as the sum of the absolute values of the off-diagonal entries of D.

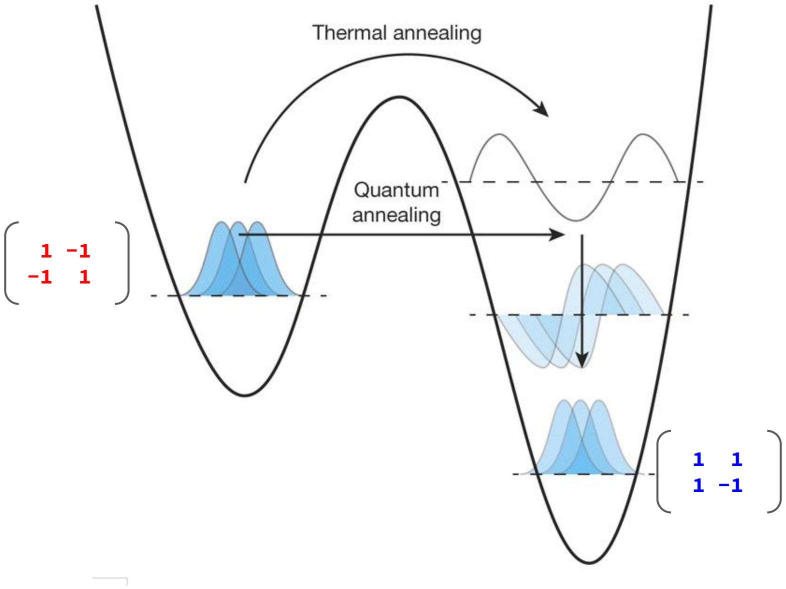
    $$E = \sum_{i \neq j} |d_{ij}|$$

  - Minimize E by flipping $[b_{ij}]$: 1 <->-1



Evolution towards the “ground states” (E=0)





Objective (Energy) Function

- Too many terms to do by hand
- Needs automatic
- => Symbolic Computing

MxM binary matrix $B=[b_{ij}]$

$b_{ij} \to s_m$ $\quad s \in \{-1, +1\}$

$$E_k(s) \rightarrow E_k(q) \rightarrow E_2(q) \rightarrow E_2(s) \rightarrow \hat{H}_2(\hat{\sigma})$$

H_{pot}

$$\begin{pmatrix} s_0 & s_M & \cdots & s_{M(M-1)} \\ s_1 & s_{M+1} & \cdots & s_{M(M-1)+1} \\ \cdots & \cdots & \cdots & \cdots \\ s_{M-1} & s_{2M-1} & \cdots & s_{M^2-1} \end{pmatrix}$$

$$C_\wedge(q_i, q_j, q_k; \delta_{i,j}) = \delta_{i,j}(3q_k + q_i q_j - 2q_i q_k - 2q_j q_k)$$

$$q_i q_j \leftarrow q_k + C_\wedge(q_i, q_j, q_k; \delta_{i,j})$$

Energy Formulation

$$E_k(s) = E_4(s) = \frac{M^2(M-1)}{2} + 2 \sum_{i<j<m<n<M^2-1} s_i s_j s_m s_n$$

Quantum Annealer (D-Wave)only takes 2-body interactions terms

Quantum Annealing

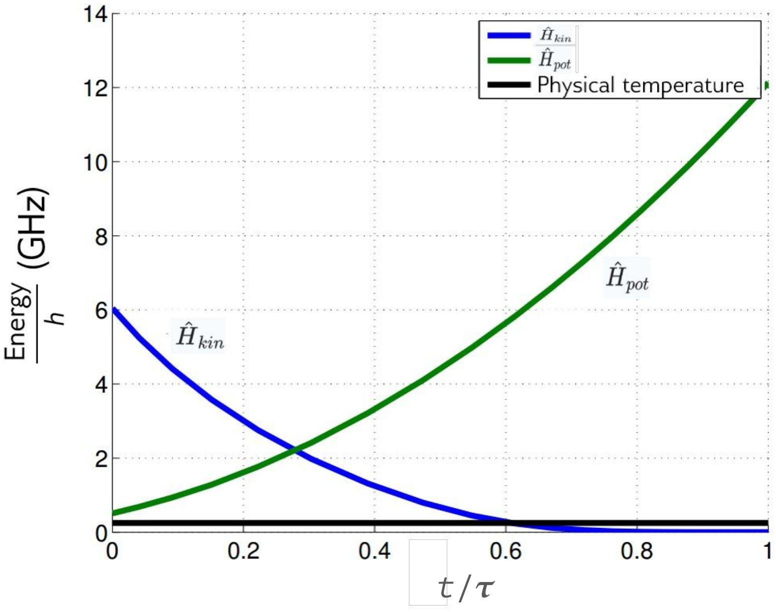
$$\hat{H}_{QA}(\hat{\sigma}, t) = (1 - \frac{t}{\tau})\hat{H}_{kin}(\hat{\sigma}) + \frac{t}{\tau}\hat{H}_{pot}(\hat{\sigma})$$

$$\hat{H}_{pot}(\hat{\sigma}) \equiv -\sum_{i \neq j} J_{ij}\hat{\sigma}_i^z\hat{\sigma}_j^z - \sum_i h_i\hat{\sigma}_i^z$$

(problem Hamiltonian)

$$\hat{H}_{kin}(\hat{\sigma}) \equiv -\Gamma \sum_i \hat{\sigma}_i^x$$

(driver Hamiltonian/
Transverse B-Field)

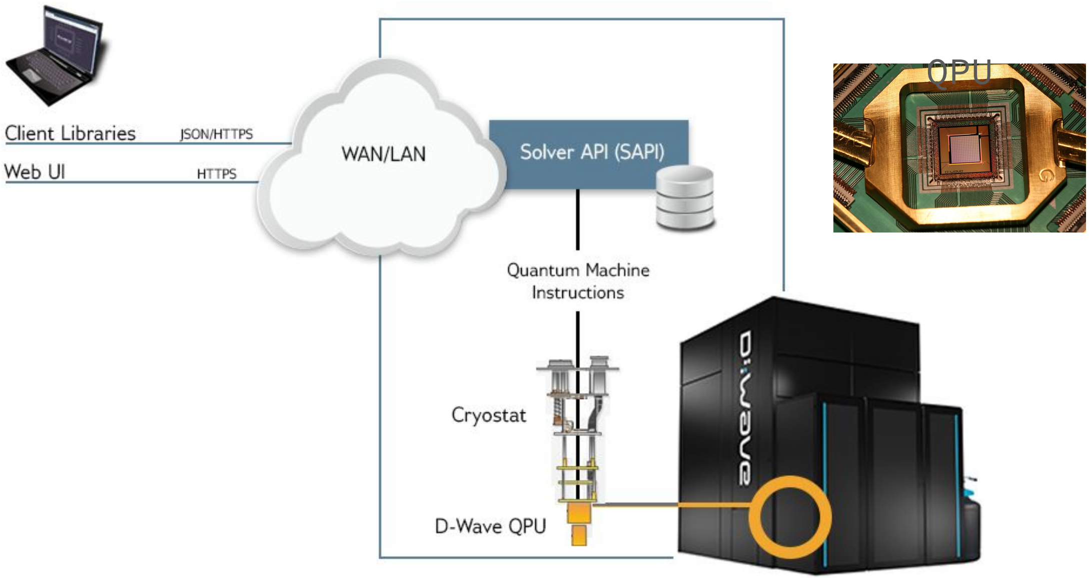


Annealing Schedule

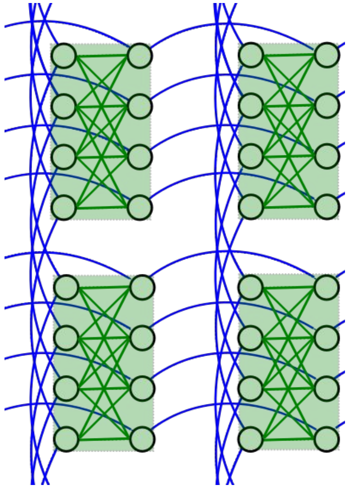
Experiments

A Brief on D-Wave 2000Q

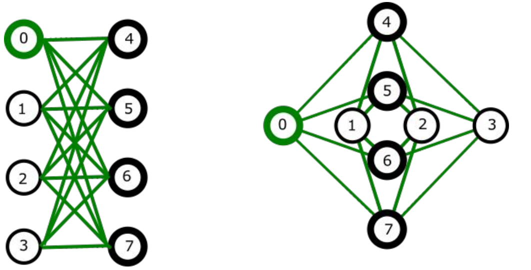
- Number of qubits: 2,048
- Number of couplers: 6,016
- Qubits connection: Chimera



QPU



Chimera Structure



Chimera Unit Cell: $K_{4,4}$
(Complete Bipartite Graph)

Experiment-1: 2-order

$$\begin{pmatrix} s_0 & s_2 \\ s_1 & s_3 \end{pmatrix}$$

$$\begin{pmatrix} s_0 & s_2 \\ s_1 & s_3 \end{pmatrix}\begin{pmatrix} s_0 & s_1 \\ s_2 & s_3 \end{pmatrix} = \begin{pmatrix} s_0s_0+s_2s_2 \\ s_0s_1+s_2s_3 \\ s_1s_0+s_3s_2 \\ s_1s_1+s_3s_3 \end{pmatrix} \Rightarrow D=\begin{pmatrix} 2 & s_0s_1+s_2s_3 \\ s_1s_0+s_3s_2 & 2 \end{pmatrix}$$

D is symmetric => take upper-part only

Energy = Σ(off-diagonal entries)²

$$\Rightarrow E_k=(s_0s_1+s_2s_3)^2=(1+2s_0s_1s_2s_3+1)$$

$$E_k(s) = 2 + 2s_0s_1s_2s_3$$

$$E_2(s) = 28 + 6s_0 + 6s_1 + 6s_2 + 6s_3 - 12s_4 - 12s_5 + 2s_0s_1 + 4s_0s_2 + 2s_0s_3 - 8s_0s_4 - 4s_0s_5 + 2s_1s_2 + 4s_1s_3 - 4s_1s_4 - 8s_1s_5 + 2s_2s_3 - 8s_2s_4 - 4s_2s_5 - 4s_3s_4 - 8s_3s_5 + 8s_4s_5$$

$$\begin{aligned}\hat{H}_2(\hat{\sigma}) = \; & 28 + 6\hat{\sigma}_0^z + 6\hat{\sigma}_1^z + 6\hat{\sigma}_2^z + 6\hat{\sigma}_3^z - 12\hat{\sigma}_4^z - 12\hat{\sigma}_5^z \\ & + 2\hat{\sigma}_0^z\hat{\sigma}_1^z + 2\hat{\sigma}_0^z\hat{\sigma}_2^z + 2\hat{\sigma}_0^z\hat{\sigma}_3^z - 8\hat{\sigma}_0^z\hat{\sigma}_4^z - 4\hat{\sigma}_0^z\hat{\sigma}_5^z \\ & + 2\hat{\sigma}_1^z\hat{\sigma}_2^z + 4\hat{\sigma}_1^z\hat{\sigma}_3^z - 4\hat{\sigma}_1^z\hat{\sigma}_4^z - 8\hat{\sigma}_1^z\hat{\sigma}_5^z + 2\hat{\sigma}_2^z\hat{\sigma}_3^z \\ & - 8\hat{\sigma}_2^z\hat{\sigma}_4^z - 4\hat{\sigma}_2^z\hat{\sigma}_5^z - 4\hat{\sigma}_3^z\hat{\sigma}_4^z - 8\hat{\sigma}_3^z\hat{\sigma}_5^z + 8\hat{\sigma}_4^z\hat{\sigma}_5^z\end{aligned}$$

Ising Coefficients

$$J = \begin{pmatrix} * & 0.167 & 0.333 & 0.167 & -0.667 & -0.333 \\ * & * & 0.167 & 0.333 & -0.333 & -0.667 \\ * & * & * & 0.167 & -0.667 & -0.333 \\ * & * & * & * & -0.333 & -0.667 \\ * & * & * & * & * & 0.667 \\ * & * & * & * & * & * \end{pmatrix}$$

$$h = (0.5, 0.5, 0.5, 0.5, -1.0, 1.0)$$

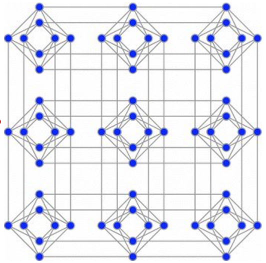
Experiment-1: 2-order H-matrix

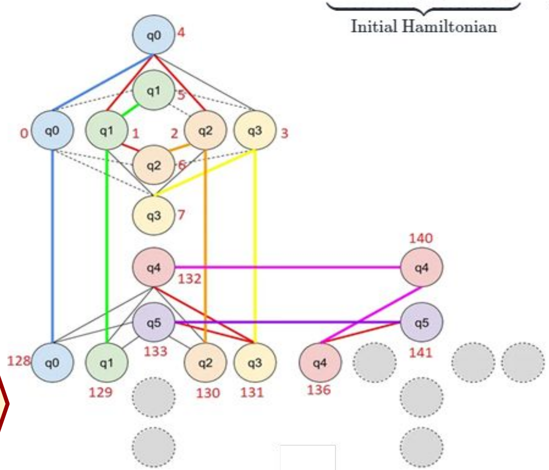
$$\mathcal{H}_{ising} = \underbrace{-\frac{A(s)}{2}\left(\sum_i \hat{\sigma}_x^{(i)}\right)}_{\text{Initial Hamiltonian}} + \underbrace{\frac{B(s)}{2}\left(\sum_i h_i \hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}\right)}_{\text{Final Hamiltonian}}$$

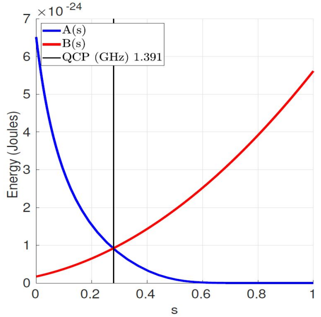
$$\begin{aligned}\hat{H}_2(\hat{\sigma}) \quad = \quad & 28 + 6\hat{\sigma}_0^z + 6\hat{\sigma}_1^z + 6\hat{\sigma}_2^z + 6\hat{\sigma}_3^z - 12\hat{\sigma}_4^z - 12\hat{\sigma}_5^z \\ & + 2\hat{\sigma}_0^z\hat{\sigma}_1^z + 2\hat{\sigma}_0^z\hat{\sigma}_2^z + 2\hat{\sigma}_0^z\hat{\sigma}_3^z - 8\hat{\sigma}_0^z\hat{\sigma}_4^z - 4\hat{\sigma}_0^z\hat{\sigma}_5^z \\ & + 2\hat{\sigma}_1^z\hat{\sigma}_2^z + 4\hat{\sigma}_1^z\hat{\sigma}_3^z - 4\hat{\sigma}_1^z\hat{\sigma}_4^z - 8\hat{\sigma}_1^z\hat{\sigma}_5^z + 2\hat{\sigma}_2^z\hat{\sigma}_3^z \\ & - 8\hat{\sigma}_2^z\hat{\sigma}_4^z - 4\hat{\sigma}_2^z\hat{\sigma}_5^z - 4\hat{\sigma}_3^z\hat{\sigma}_4^z - 8\hat{\sigma}_3^z\hat{\sigma}_5^z + 8\hat{\sigma}_4^z\hat{\sigma}_5^z\end{aligned}$$

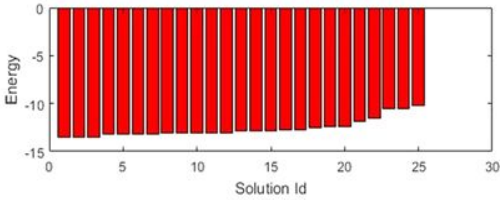
$$J = \begin{pmatrix} * & 0.167 & 0.333 & 0.167 & -0.667 & -0.333 \\ * & * & 0.167 & 0.333 & -0.333 & -0.667 \\ * & * & * & 0.167 & -0.667 & -0.333 \\ * & * & * & * & -0.333 & -0.667 \\ * & * & * & * & * & 0.667 \\ * & * & * & * & * & * \end{pmatrix}$$

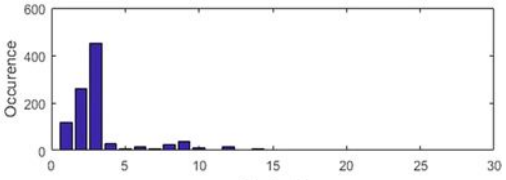
$$h = (0.5, 0.5, 0.5, 0.5, -1.0, 1.0)$$











$$\begin{pmatrix} + & - \\ - & - \end{pmatrix}$$

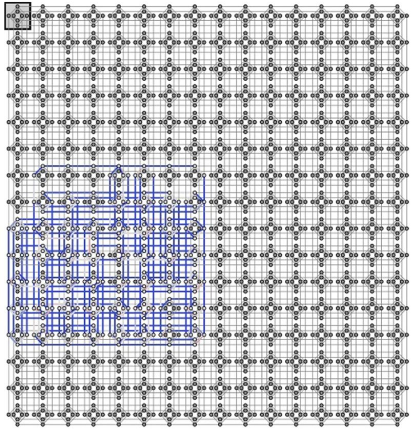
$$\begin{pmatrix} + & + \\ - & + \end{pmatrix} \quad \begin{pmatrix} + & - \\ - & - \end{pmatrix} \quad \cdots \quad \begin{pmatrix} - & + \\ + & + \end{pmatrix}$$

Successful … !

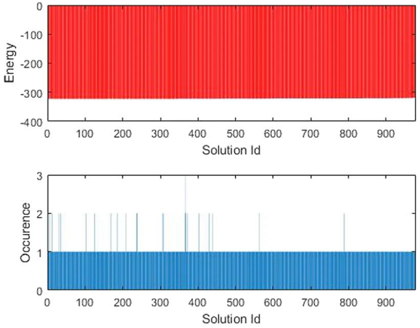
Experiment-2: Find 4-order H-matrix

$$\hat{H}_2(\hat{\sigma}) = 1,248 + 66\hat{\sigma}_0^z + \cdots - 44\hat{\sigma}_{39}^z + 6\hat{\sigma}_0^z\hat{\sigma}_1^z + \cdots + 8\hat{\sigma}_{38}^z\hat{\sigma}_{39}^z$$

389-terms



(a)



(b)

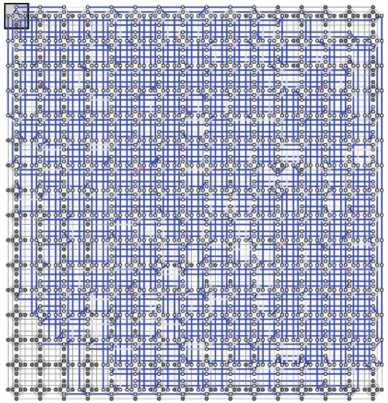
$$\begin{pmatrix} - & - & + & + \\ + & - & + & - \\ - & - & - & - \\ - & + & + & - \end{pmatrix}$$

Solution found … !

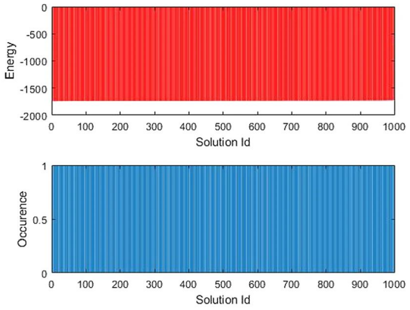
Experiment-3: finding 3-orthogonal 12-length vectors

$$\hat{H}_2(\hat{\sigma}) = 19,872 + 404\hat{\sigma}_0^z + \cdots + 404\hat{\sigma}_{71}^z + 4\hat{\sigma}_0^z\hat{\sigma}_1^z + \cdots + 8\hat{\sigma}_{70}^z\hat{\sigma}_{71}^z$$

72-variables with 7,765 terms => 1,766 qubits



(a)



(b)

$$v_0 = (+,-,+,-,-,+,+,-,+,+,-,-)^T$$
$$v_1 = (+,-,-,-,+,+,-,+,+,-,+,-)^T$$
$$v_2 = (+,+,-,+,+,+,+,+,+,+,-,+)^T$$

Solution found … !

1,000 D-Wave answer

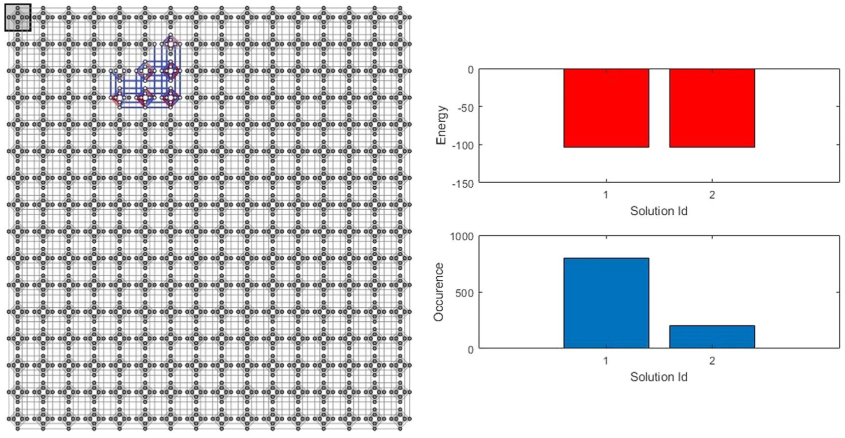
> 635: wrong

> 365: correct

Experiment-4: missing 1 vector in H-12

$$
\begin{aligned}
v_0 &= (+,+,+,+,+,+,+,+,+,+,+,+)^T \\
v_1 &= (+,-,+,-,+,+,+,-,-,-,+,-)^T \\
v_2 &= (+,-,-,+,-,+,+,+,-,-,-,+)^T \\
v_3 &= (+,+,-,-,+,-,+,+,+,-,-,-)^T \\
v_4 &= (+,-,+,-,-,+,-,+,+,+,-,-)^T \\
v_5 &= (+,-,-,+,-,-,+,-,+,+,+,-)^T \\
v_6 &= (+,-,-,-,+,-,-,+,-,+,+,+)^T \\
v_7 &= (+,+,-,-,-,+,-,-,+,-,+,+)^T \\
v_8 &= (+,+,+,+,-,-,-,+,-,-,+,-)^T \\
v_9 &= (+,+,+,-,-,-,+,-,-,+,-,+)^T \\
v_{10} &= (+,+,-,+,+,+,-,-,-,+,-,-)^T
\end{aligned}
$$

V_{11} = … ?



(a) (b)

$$v_{11} = (-,+,-,-,-,+,+,+,-,+,+,-)^T$$

Solution … !

$$\hat{H}_2(\hat{\sigma}) = 756 + 2\hat{\sigma}_0^z\hat{\sigma}_1^z + \cdots + 2\hat{\sigma}_0^z\hat{\sigma}_{27}^z + \cdots - 2\hat{\sigma}_{26}^z\hat{\sigma}_{27}^z$$

27 variables, 379 terms > 50 qubits (after embedding)

Summary

- We introduced the concept of quantum annealing
- Study Case: Finding H-matrix problem using Quantum Computing
    - Formulation of Hamiltonian/Energy Function
        - Conversion into 2-body interactions
    - Implementation on Quantum Annealer (D-Wave)
        - D-Wave 2000Q, max 2048 qubits
    - Capable to only up-to 4-order
    - Solving related QUBO problems
        - FInding n-set of m-order orthogonal vectors (n<m)
        - Finding n-missing vector of m-order H-matrix

End of Section

Appendix

Ising Hamiltonian

- D-Wave solve a problem when it is expressed in Ising Hamiltonian

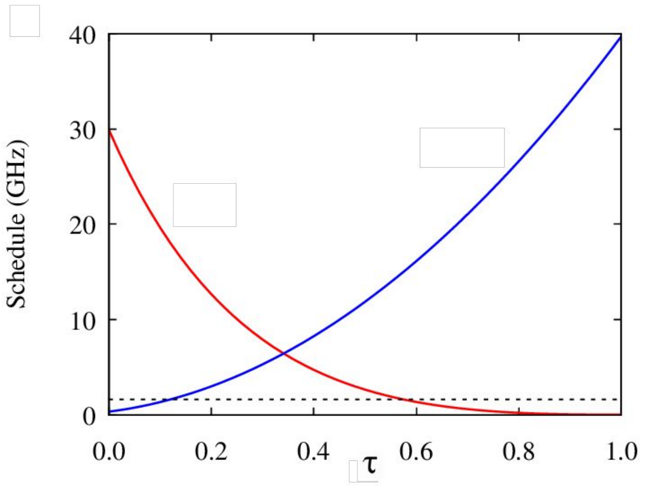
$$H_P = \Sigma_i \Sigma_j J_{ij} \sigma_i^z \sigma_j^z + \Sigma_i h_i \sigma_i^z$$

Sometimes, it is also called H_{pot}

- The user translate his/her problem into H_P, then he/she specifies J_{ij} and h_i into a quantum annealer D-Wave).

- D-Wave provides driver/kinetic Hamiltonian H_{kin}, which is a transfer (magnetic field) The annealing is performed as follows

$$\hat{H}_{QA}(\hat{\sigma}, t) = \left(1 - \frac{t}{\tau}\right) \hat{H}_{kin}(\hat{\sigma}) + \frac{t}{\tau} \hat{H}_{pot}(\hat{\sigma})$$



Introduction to Quantum Optimization and Its Applications

Andriyan B. Suksmono, Ph.D.
Professor
School of Electrical Engineering and Informatics, ITB, Bandung

Workshop on Classical and Quantum Machine Learning for Condensed Matter Physics|(smr 3948)

VQE
(Variational Quantum Eigensolver)

Andriyan B. Suksmono, Ph.D.
Professor
School of Electrical Engineering and Informatics, ITB, Bandung

Workshop on Classical and Quantum Machine Learning for Condensed Matter Physics|(smr 3948)

Introduction

- A hybrid classical-quantum algorithm
- Find a minimum eigenvalue of an Hermitian matrix H: the ground state
  - Ideally phase estimation algorithm: need a “deep” circuit, an issue for existing NISQ device
  - VQE: need only shallow circuits
- Problem: given H, find minimum eigenvalue λ_{min} with associated $|\psi_{min}>$
- VQE: provides an estimate λ_{θ} bounding : λ_{min}

$$\lambda_{min} \leq \lambda_{\theta} \equiv \langle\psi(\theta)|H|\psi(\theta)\rangle$$

- Where $|\psi(\theta)>$ is eigenstate associated with λ_{θ}
- How?
  - Apply parameterized circuit $U(\theta)$ to starting state $|\psi(\theta)>$
  - It will yield $U(\theta)|\psi(\theta)>=|\psi(\theta)>$
  - Change parameter θ to minimized $<\psi(\theta)H|\psi(\theta)>$

https://github.com/Qiskit/textbook/blob/main/notebooks/ch-applications/vqe-molecules.ipynb

Variational Method in Quantum Mechanics

Spectral theorem: eigenvalue of an hermitian matrix $H=H^\dagger$ is real, i.e. $\lambda_i = \lambda^*_i$.
Moreover, H can be expressed as

$$H = \Sigma_{i=1}^{N} \lambda_i |\psi_i\rangle\langle\psi_i|$$

The expectation of observable of H on a quantum state $|\psi\rangle$ is given by

$$\langle H\rangle_\psi = \langle\psi|H|\psi\rangle$$
$$= \langle\psi| \left(\Sigma_{i=1}^{N} \lambda_i |\psi_i\rangle\langle\psi_i|\right)|\psi\rangle \quad = \Sigma_{i=1}^{N} \lambda_i \langle\psi|\psi_i\rangle \langle\psi_i|\psi\rangle$$
$$= \Sigma_{i=1}^{N} \lambda_i |\langle\psi_i|\psi\rangle|^2$$

=> the expectation value of an observable on any state can be expressed as a linear combination using the eigenvalues associated with H as the weights. The weights are non-negatives.

$$\lambda_{min} \leq \langle H\rangle_\psi = \langle\psi|H|\psi\rangle = \Sigma_{i=1}^{N} \lambda_i |\langle\psi_i|\psi\rangle|^2$$

… the variational method

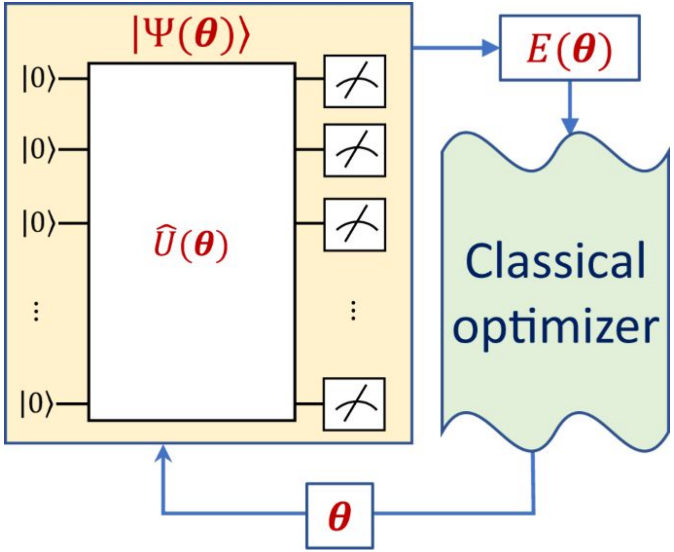
The VQE

- Need ansatz to implement the variational form.
- VQE employs parameterized circuit with a fix form: the *variational form*.
    - Its action is represented by linear transform $U(\Theta)$
    - $U(\Theta)$ is applied to starting state $|\psi\rangle$, such as the vacuum state $|0\rangle$ or the Hartree Fock state and generates an output state $U(\Theta)|\psi\rangle \equiv |\psi(\Theta)\rangle$
    - Iterative optimization over $|\psi(\Theta)\rangle$ aims to yield an expectation value

$$\langle\psi(\Theta)|H|\psi(\Theta)\rangle \cong E_{gs} \equiv \lambda_{min}$$

VQE: the algorithm

1. Encode problem into a qubit Hamiltonian (sum of Pauli operators and their (tensor) products)
2. Choose/update an **ansatz** for state preparation on the quantum computer and build the quantum circuit
3. Measure the basis of the qubit Hamiltonian to get expectation values for the states
4. Send the result to classical optimizer to update gate/wave parameters
5. Repeat 2-4 until convergence



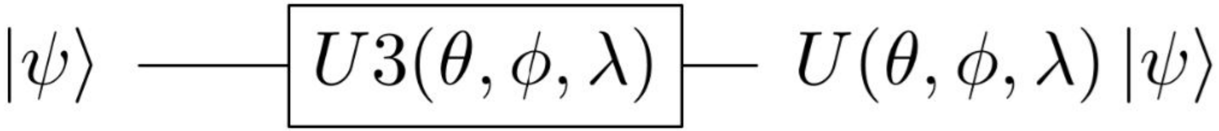
Eigen-value calculation: Conventional vs VQE

- Basically, VQE is a method to calculate eigenvalue of a matrix H, which represents the Hamiltonian of a quantum system.
- Conventional/Numerical Methods
  - **Mature Techniques**: These methods are well-studied, optimized, and widely implemented in various software libraries.
  - **Precision**: They can achieve high precision, limited mainly by the numerical precision of the computer.
  - **Deterministic**: These methods typically provide deterministic results for eigenvalue computations.
- VQE
  - **Scalability**: VQE is potentially more scalable for certain problems where the Hamiltonian has an exponentially large state space, as it uses quantum resources to represent and manipulate quantum states.
  - **Quantum Advantage**: VQE can exploit quantum parallelism and entanglement, potentially providing advantages for problems that are hard for classical computers.
  - **Flexibility**: It is well-suited for Noisy Intermediate-Scale Quantum (NISQ) devices, as it can work with the noise and errors inherent in current quantum hardware through error mitigation techniques.

Example-1: simple 1 qubit

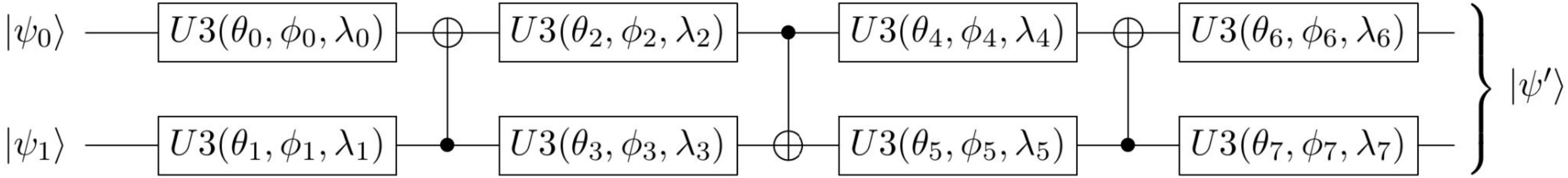
- The *n*-qubit variational form able to generate any |**ψ**>, where |**ψ**>$\in \mathbb{C}^N$, and $N=2^n$.
- Consider *n*=1, U3 gate with parameters θ, ϕ, and λ represents

$$U3(\theta, \phi, \lambda) = \begin{pmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i\lambda+i\phi}\cos\frac{\theta}{2} \end{pmatrix}$$

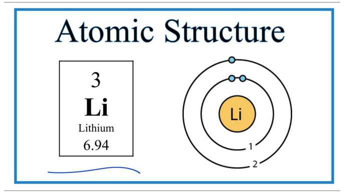


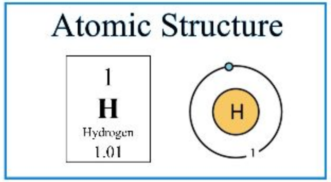
Example-2: 2 qubits

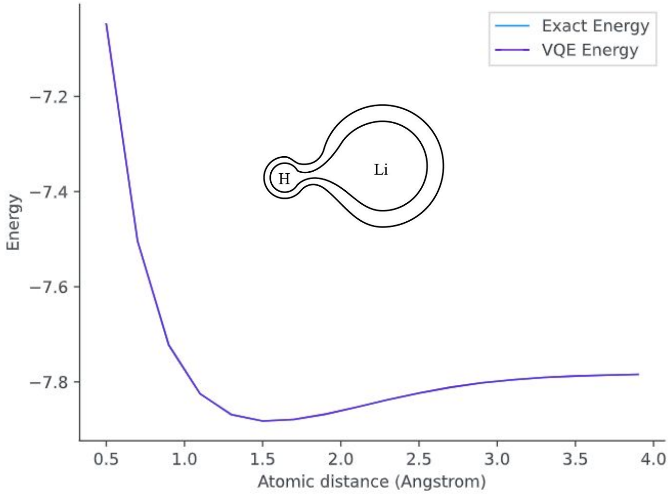
- Universal 2 qubit circuit



Application Example: Ground State of Li-H

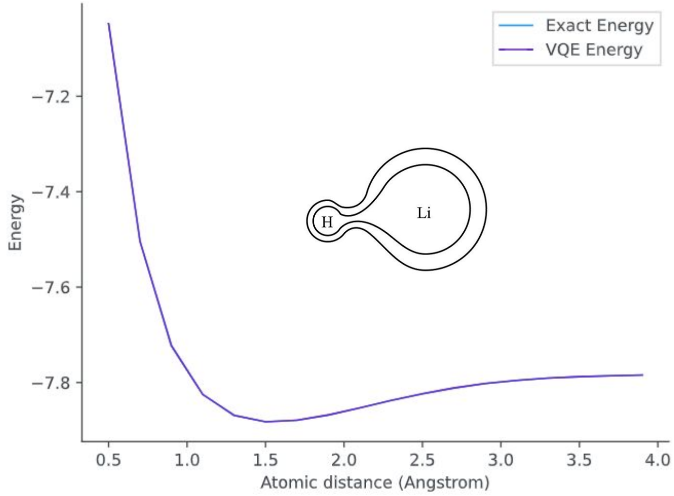






- **Calculation Process**: For each atomic distance R, the VQE algorithm calculates the ground state energy by finding the minimum eigenvalue of the Hamiltonian H(R).
- **The Graph**:The curve shows how the energy changes with varying bond length, with the lowest point indicating the optimal bond length and bond strength of the LiH molecule.
- **Physical Meaning**:
  - The minimum point on the curve represents the most stable configuration of the LiH molecule.
  - The energy at this point is the ground state energy of the molecule at equilibrium bond length, which is a crucial property in understanding the molecule's behavior and interactions.

Performance Comparison



- **VQE**
  - Equilibrium Bond Length: ~ 1.5 to 1.6. Å.
  - Ground State Energy: −7.88 Hartree.
- **Experimental**
  - Equilibrium Bond Length: 1.595 Å.

Hartree-Fock (HF) Method:

- Equilibrium Bond Length: Around 1.6 Å.
- Ground State Energy: Approximately −7.83 Hartree.

Configuration Interaction (CI) Method:

- Equilibrium Bond Length: Around 1.6 Å.
- Ground State Energy: More accurate than HF, closer to the exact solution.

Coupled Cluster with Single, Double, and Perturbative Triple Excitations (CCSD(T)):

- Equilibrium Bond Length: Around 1.6 Å.
- Ground State Energy: Approximately −7.88 Hartree, which is considered highly accurate and close to the experimental value.

Density Functional Theory (DFT):

- Equilibrium Bond Length: Around 1.6 Å, but can vary slightly depending on the functional used.
- Ground State Energy: Generally lower than HF but depends on the specific functional.

End of Section

Appendix

A brief on ***ansatz***

- A parameterized quantum circuit or function that represent a trial wave function for approximating the solution
  - A structured quantum circuit composed of quantum gates that depend on a set of adjustable parameters
  - These parameters are tuned during optimization
- Common type of ansatzes
  - UCC (Unitary Coupled Cluster)
  - Hardware-type ansatz
  - Problem specific ansatz
- Construction
  - Built using a sequence of quantum gates that applied to qubits
  - Gates parameters are variable that are optimized during the algorithm

Variational Method

- A method for estimating the ground state and its energy of a quantum state H.
- **Basic Principle**: The ground state of a quantum system (H) is less than or equal to the expectation value of the energy for any trial wave function ψ
  - $E_0 \leq <\psi|H|\psi>$
- Procedure:
  - **Choose Trial Wave-Function**: normalized parameter $\{\alpha_1, \alpha_2, ..\}$ dependent $\psi(\alpha_1, \alpha_2, ..)$
  - **Calculate the Expectation Value of the Energy**

$$E(\alpha_1, \alpha_2, \cdots) = \frac{<\psi(\alpha_1,\alpha_2,\cdots)|H|<\psi(\alpha_1,\alpha_2,\cdots)}{<\psi(\alpha_1,\alpha_2,\cdots)|\psi(\alpha_1,\alpha_2,\cdots)>}$$

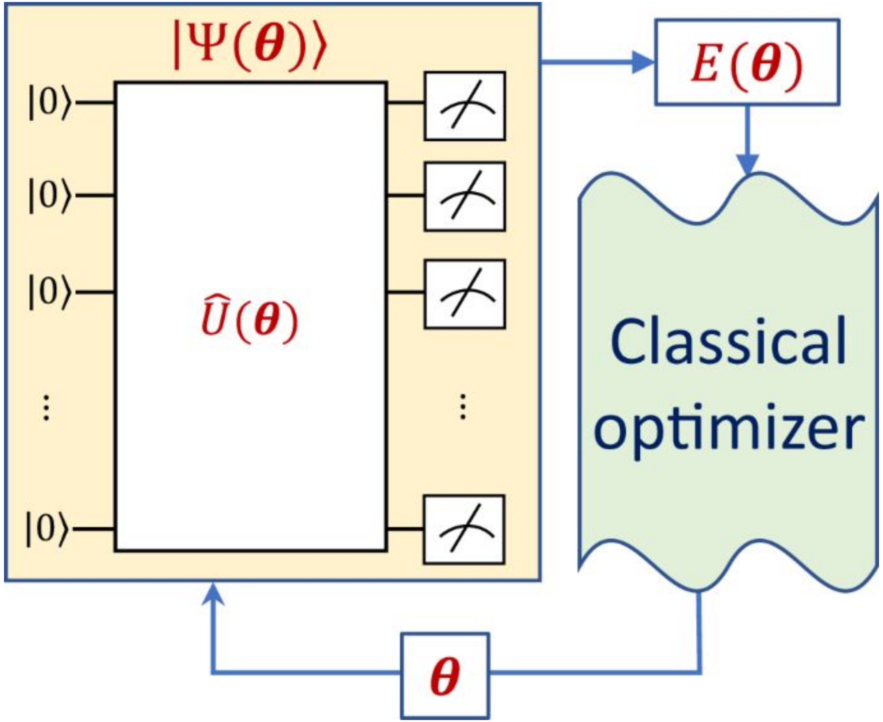
  - **Minimize the Energy:** by adjusting parameters: $\alpha_1, \alpha_2, ..$
    - The minimum is the ground state E0.
- **Usage:** estimate ground state energy and wave function without requiring exact solution of Schrodinger Equation

REF Examples

- Simple eigenvalue problem
  - https://medium.com/@beef_and_rice/get-started-vqe-with-blueqat-2ef6a73bbaee
- Molecule problem
  - https://medium.com/mdr-inc/vqe-and-quantum-chemistry-on-blueqat-acd0e91b4d24

VQE: Basic Concept

- A hybrid quantum-classical method to minimize expectation value of the energy of a Hamiltonian H.
  - The Quantum Computer: prepares and measures quantum states
  - The Classical Computer: optimizes parameters to minimize energy



VQE: Procedure

1. **Choose a Parameterized Quantum State:**
   select trial wave function $|\psi(\theta)>$
2. **Prepare quantum state:**
   use quantum computer to prepare $|\psi(\theta)>$
3. **Measure Energy:**
   $E(\theta) = <\psi(\theta)|H|\psi(\theta)>$
4. **Classical Optimization:**
   use classical computer to adjust θ to minimize $E(\theta)$
5. **Iterate:**
   repeat the preparation, measurement, and optimization steps until energy is minimized

Example: Finding Ground State of Lithium Hydride (LiH)

1. Construction of Molecular Hamiltonian:
   a. Quantum Chemistry Methods
   b. Second Quantization and Mapping
2. Parameterized Quantum State Preparation
   a. Ansatz Selection: $|\psi(\theta)>$
   b. Initialized the parameters: θ
3. Execute Quantum Circuit
4. Measurement of Expectation Values
   a. Hamiltonian Decomposition
   b. Measurements
5. Classical Optimization
   a. Calculate Energy
   b. Update/optimize Parameters
6. Repeat 2-5 (until convergence)

The Abdus Salam International Centre for Theoretical Physics

QAOA

(Quantum Approximate Optimization Algorithm)

Andriyan B. Suksmono, Ph.D.
Professor
School of Electrical Engineering and Informatics, ITB, Bandung

Workshop on Classical and Quantum Machine Learning for Condensed Matter Physics|(smr 3948)

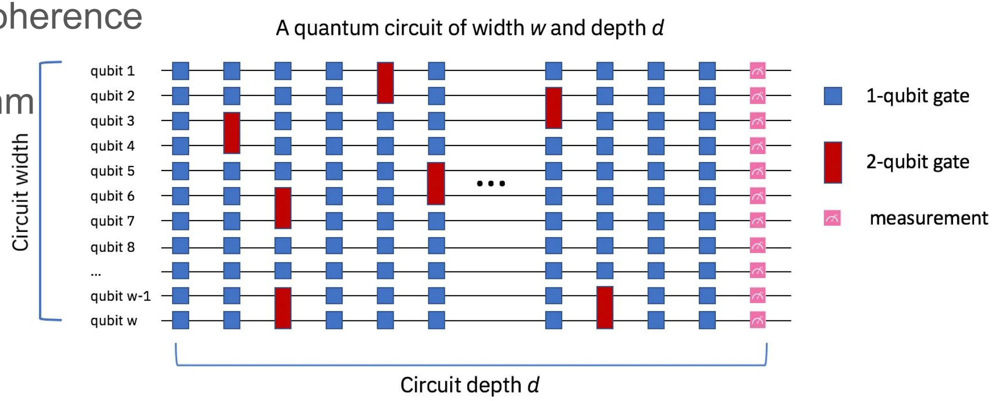
Background

- Quantum computers can solve problems faster than classical computers
  - Shor's factoring algorithm
  - Grover Search
  - Quantum Simulations
- Resource needed in Shor's Algorithm (with significant number of digits)
  - Without ECC : ~ 1,000 qubits
  - With ECC : ~ 1,000,000 qubits, ~1,000,000 gates
- Existing quantum device
  - NISQ (Noisy Intermediate Scale Quantum)
  - Cannot handle such problem(s)

REF: Peter Shor's "QAOA Talk"

What do current quantum computers good for?

- Algorithm that can run on small number of qubits, gates, and shallow circuit
  - Useful enough (to solve real-world problems)
  - Doesn't need (extensive) error correction
  - Can be implemented on small number of qubits (<1,000)
- Solution: QAOA
  - Low-depth: need not too much coherence
  - Robust to error
  - Hybrid classical-quantum algorithm

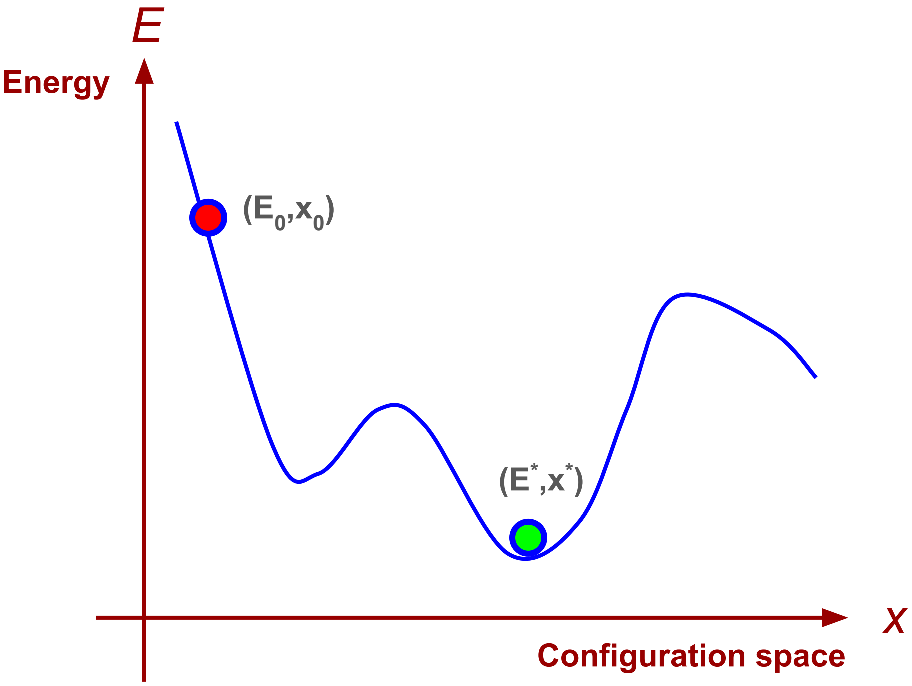


REFs:

- Peter Shor's "QAOA Talk"
- https://uwaterloo.ca/institute-for-quantum-computing/news/quantum-advantage-shallow-circuits

Optimization: general idea

- **Objective**: bring the ***initial*** state x_0 (with initial energy E_0) to the ***final*** optimum state x* (with optimum energy E*)
  - x_0 is the initial guess
  - x* is the solution
- **Constraint**
  - Limited computing resource (space/#qubits, time)
  - Best way to manage (local minimum) traps: classical, quantum



problem → $\hat{H} = \hat{H}_1 + \hat{H}_2 + \cdots + \hat{H}_N$

unitary

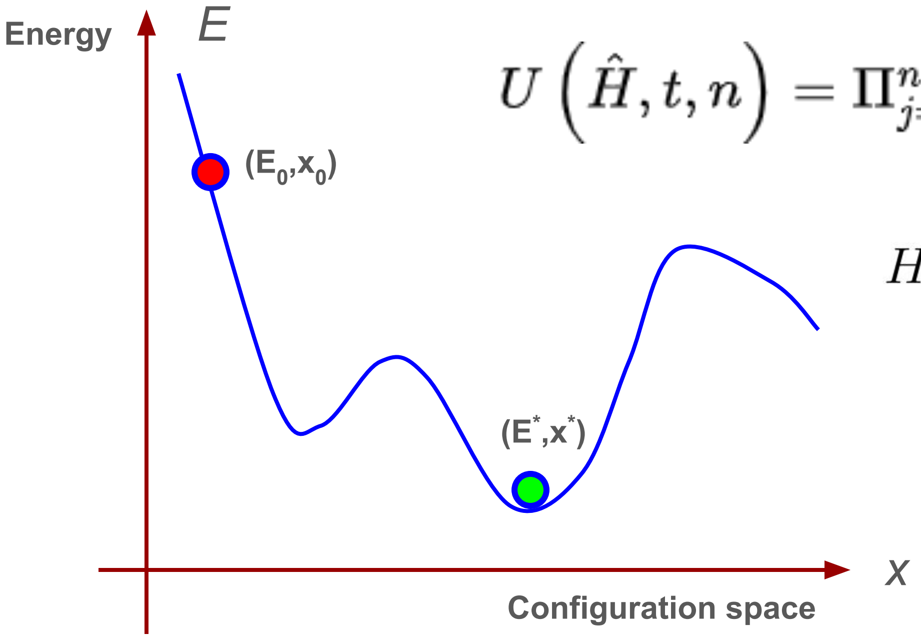
time-evolution

$$U\left(\hat{H}, t\right) = e^{-i\hat{H}t/\hbar}$$

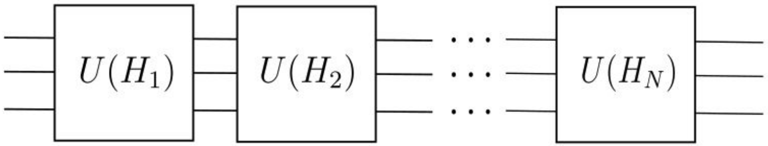
Trotter-Suzuki theorem

$$e^{\hat{A}+\hat{B}} \approx \left(e^{\hat{A}/n} e^{\hat{B}/n}\right)^n$$

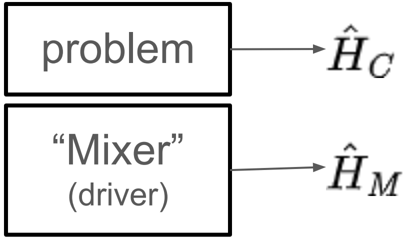
$$U\left(\hat{H}, t, n\right) = \Pi_{j=1}^{n} \Pi_k e^{-i\hat{H}_k t/n}$$



$$H = H_1 + H_2 + \cdots + H_N$$

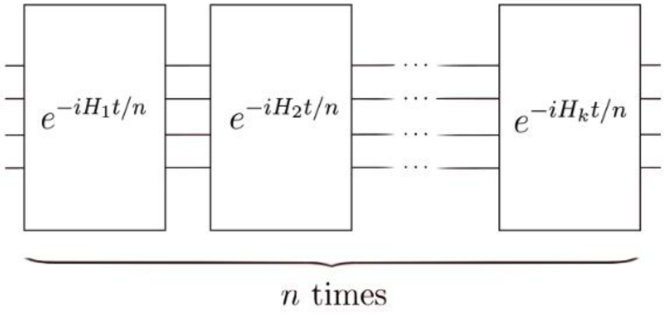


Unitary Evolution

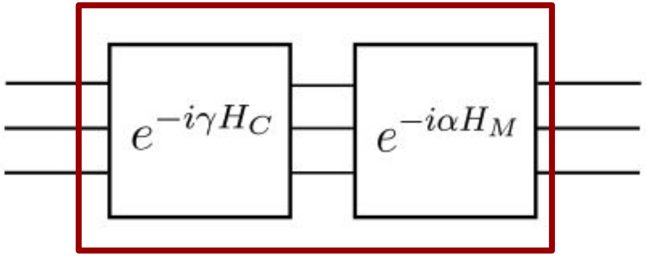


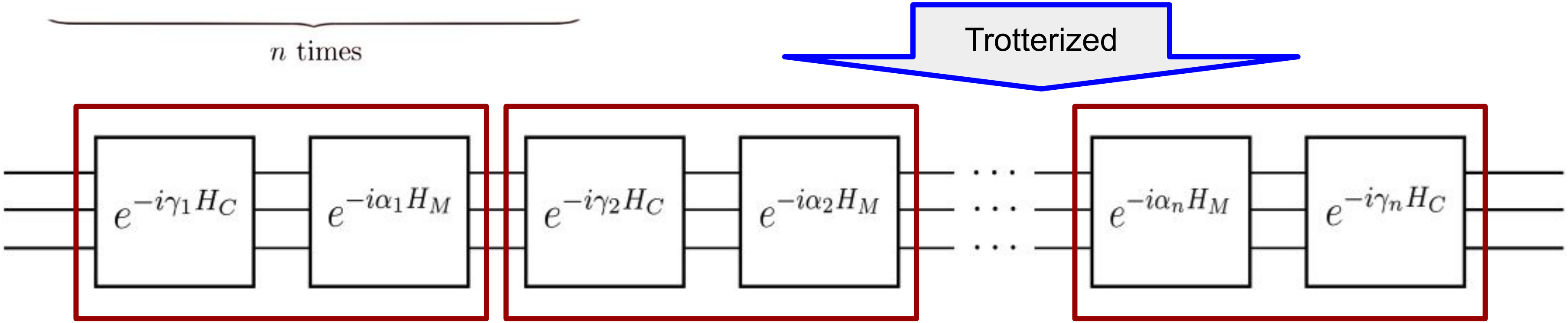
$$\hat{H} = \gamma \hat{H}_C + \alpha \hat{H}_M$$

Trotter-Suzuki decomposition



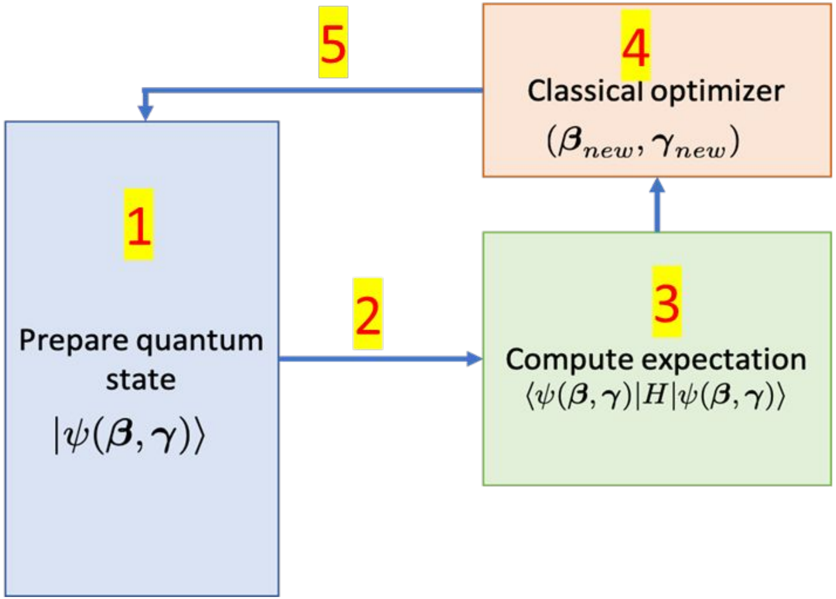
parameterized circuit





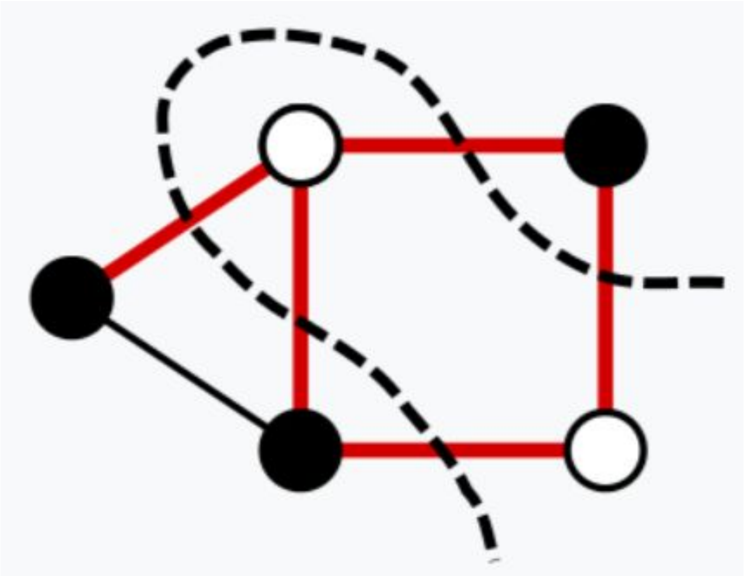
QAOA: the Algorithm

1. Initialize β and γ
2. Repeat until convergence criteria are satisfied:
   1. Prepare the state $|\psi(\beta,\gamma)\rangle$ using QAOA circuit
   2. Measure the state in standard basis
   3. Compute $\langle\psi(\beta,\gamma)|H_P|\psi(\beta,\gamma)\rangle$
   4. Find new set of parameters $(\beta_{new},\gamma_{new})$ using a classical optimization algorithm
   5. Set current parameters (β,γ) equal to the new parameters $(\beta_{new},\gamma_{new})$



Example-1
Max-Cut problems

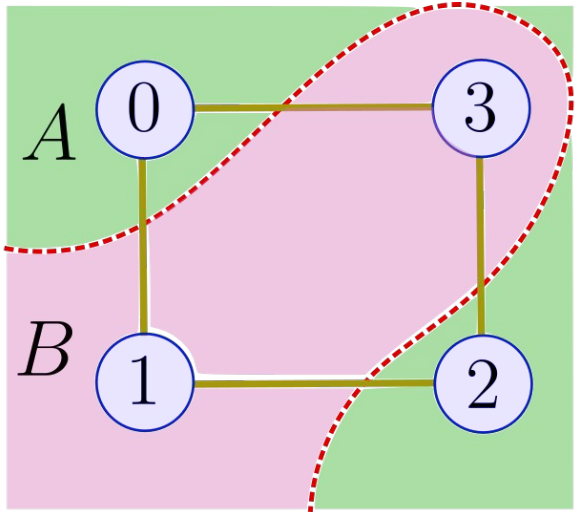
Solving MAX-CUT Problem Using QAOA



- A **maximum cut** is a cut whose size is at least the size of any other cut.
- That is, it is a **partition of the graph's vertices** into two complementary sets **S** and **T**, such that the **number of edges between S and T is as large as possible.**

https://en.wikipedia.org/wiki/Maximum_cut

Example



z="0101"

- The brute-force method:
  - exhaustively try all the binary assignments.
- Quantum computing
  - Translate into Ising model
  - Solve the problem
- We seek the partition z of vertex into two sets, A and B, that **maximize** C(z)

$$C(z) = \Sigma_{\alpha=1}^{m} C_\alpha(z)$$

- z_i=0 if vertex-i in A, z_i=1 if vertex-i in B

https://pennylane.ai/qml/demos/tutorial_qaoa_maxcut/

QAOA Circuit

- Denoting partitions using computational basis states |z>, we represent the objective function as operator

$$C_\alpha = \frac{1}{2}\left(1 - \sigma_z^j \sigma_z^k\right)$$

- QAOA is started in uniform superposition over n bitstring basis states

$$|+_n\rangle = \frac{1}{\sqrt{2}^n} \sum_{z \in \{0,1\}^n} |z\rangle$$

- We perform a sequence of operation,

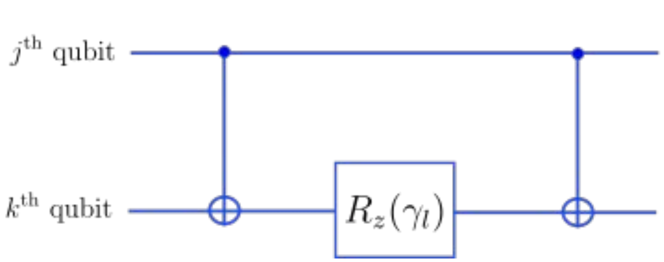
$$|\gamma, \beta\rangle = U_{B_p} U_{C_p} U_{B_{p-1}} U_{C_{p-1}} \cdots U_{B_1} U_{C_1} |+_n\rangle$$

$$U_{B_i} = e^{-i\beta_i B} = \Pi_{j=1}^{n} e^{-i\beta_i \sigma_x^j}$$

$$U_{C_i} = e^{-i\beta_i C} = \Pi_{j=1}^{n} e^{-i\gamma_i (1 - \sigma_z^j \sigma_z^k)/2}$$

Hamiltonian and Quantum Circuits

$$e^{-i\gamma_l(1-\sigma_z^j\sigma_z^k)/2}$$



Problem Hamiltonian

$$e^{-i\beta_l\sigma_x^j}$$

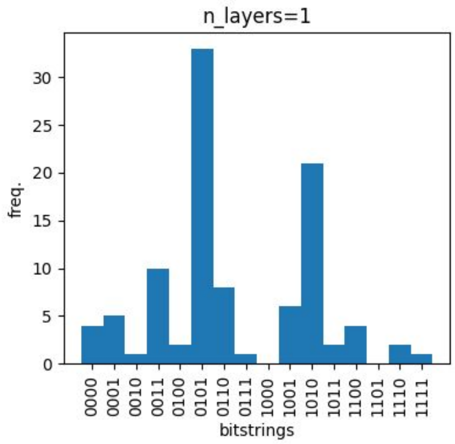
Mixer/Driver Hamiltonian

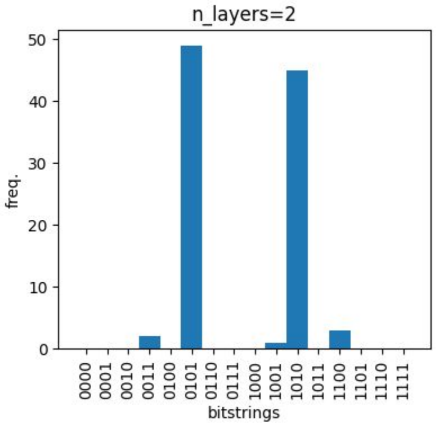
Solution

```
p=1
Objective after step     5:  4.0000000
Objective after step    10:  2.0000000
Objective after step    15:  2.0000000
Objective after step    20:  3.0000000
Objective after step    25:  2.0000000
Objective after step    30:  3.0000000
Optimized (gamma, beta) vectors:
[[-0.79877491]
 [ 0.42271535]]
Most frequently sampled bit string is: 0101
```

```
p=2
Objective after step     5:  4.0000000
Objective after step    10:  4.0000000
Objective after step    15:  4.0000000
Objective after step    20:  4.0000000
Objective after step    25:  4.0000000
Objective after step    30:  4.0000000
Optimized (gamma, beta) vectors:
[[-1.01801414 -0.96385261]
 [ 0.60409681  0.46537939]]
Most frequently sampled bit string is: 0101
```

C=4



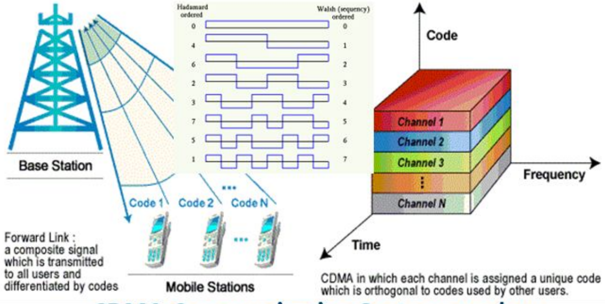


Example-2
Finding Hadamard matrices

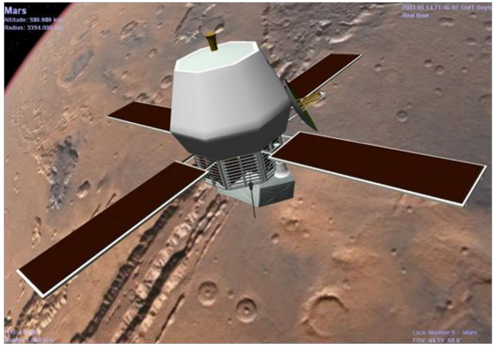
Background

- **Hadamard matrix (H-matrix)**
  Definition: an orthogonal binary {-1,1} matrix
  Applications: **orthogonal codes** used in CDMA, ECC (Error Correction Code) with maximal error correction capability, employed in Mariner-9, experiment design [Hedayat, 1973]
  Scientific/Math: H-matrix conjecture is a ~100 years old unsolved problem
- **Why finding a H-matrix is hard?**
  For an M-order matrix, there are
  $[2^{(M^2)}] \sim \exp(M^2)$ binary matrices
  H-matrix **conjecture** predicts, there is a H-matrix for every M=4k, k positive integer. How to find it?
  Brute force, worst-case condition: one should check all binary matrices, an $O[\exp(M^2)]$ problem --> **a hard problem**.

Proposed Solution: USE A QUANTUM COMPUTER !



CDMA Communication System employs Walsh-Hadamard Orthogonal Code



Mariner-9 employed Hadamard's ECC to protect Mars's images sent to Earth

Examples of the Hadamard matrices

$$(1)$$

$$\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \quad \begin{pmatrix}1 & -1\\ 1 & 1\end{pmatrix}$$

$$\begin{pmatrix}1 & 1\\ -1 & 1\end{pmatrix} \quad \cdots\cdots$$

$$\cdots\cdots \quad \begin{pmatrix}-1 & -1\\ -1 & 1\end{pmatrix}$$

$$\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{pmatrix}$$

Simplest non-trivial case

$$H_2 = \begin{pmatrix}1 & 1\\ 1 & s_1\end{pmatrix}$$

$$H_2 = \begin{pmatrix}1 & s_1\\ 1 & s_2\end{pmatrix}$$

$$H_2 = \begin{pmatrix}1 & s_1\\ s_2 & s_3\end{pmatrix}$$

$$H_2 = \begin{pmatrix}s_1 & s_3\\ s_2 & s_4\end{pmatrix}$$

Define Energy Function

- Deviation from the orthogonality condition
- Non-negative value
- E: squared sum of the off-diagonal values of D, where

  $D = H^T H$

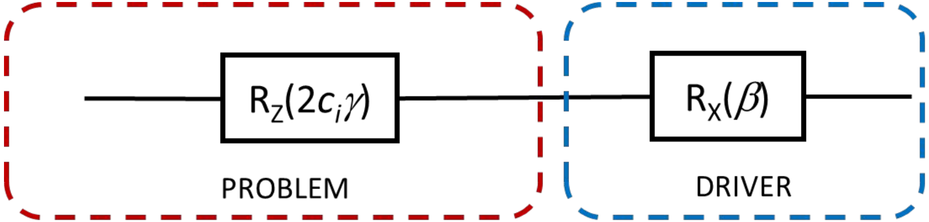
- Since $H=H(s_i)$, then $E=E(s_i)$

Convert $E(s_i)$ to Hamiltonian

1-body term

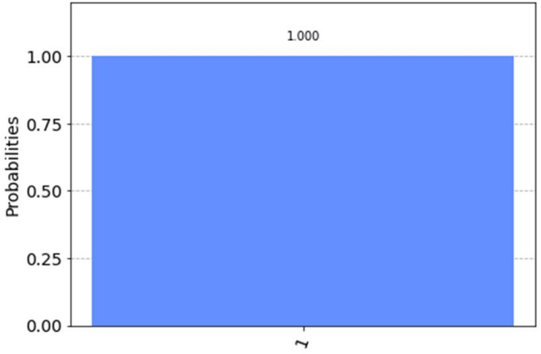
$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & s_1 \end{pmatrix}$$

$E(H_2) = (1+s_1)^2$
$= 1 + 1 + 2s_1$
$= 2(1 + s_1)$
$\Rightarrow s_1 = -1$



$$U(C,\gamma) = e^{-icj\gamma C}$$

$$U(B,\beta) = e^{-i\beta B}$$



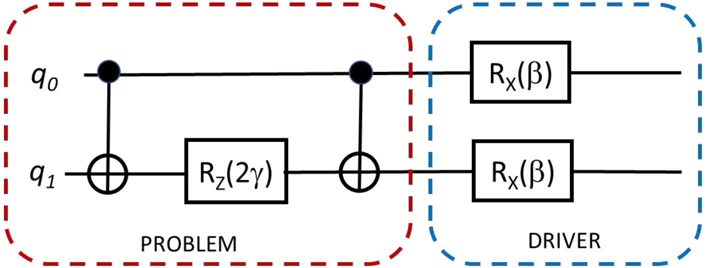
Variables conversion: 0 ⇔1; 1⇔ -1

2-body term

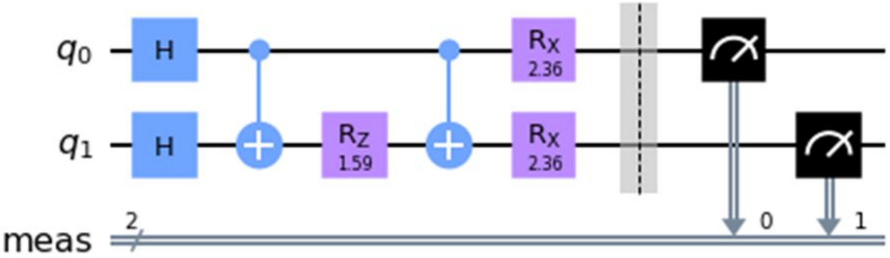
$$H_2 = \begin{pmatrix} 1 & s_1 \\ 1 & s_2 \end{pmatrix}$$

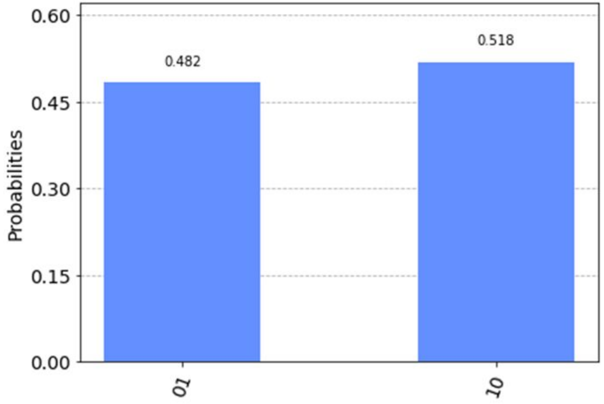
$E(H_2) = (s_1+s_2)^2$
$= 1 + 1 + 2s_1s_2$
$= 2(1 + s_1 s_2)$
$\Rightarrow s_1 = -1, s_2 = +1$, or
$\Rightarrow s_1 = +1, s_2 = -1$,

$H_{problem} = \sigma_0\sigma_1$ $H_{driver} = (\sigma^x_0+\sigma^x_1)$



$U(C,\gamma) = e^{-i\gamma C}$ $U(B,\beta) = e^{-i\beta B}$

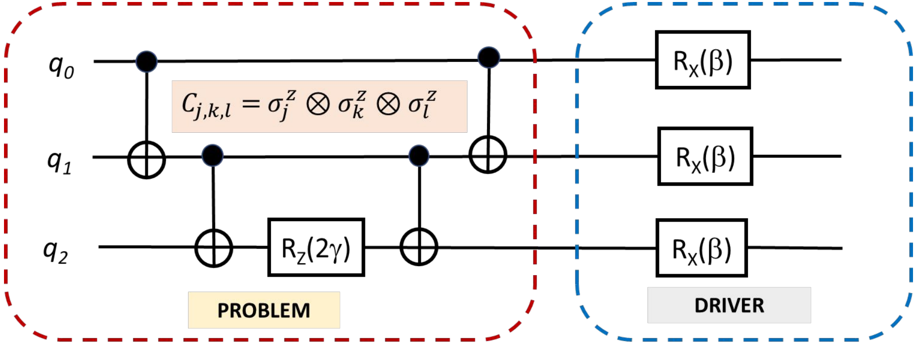




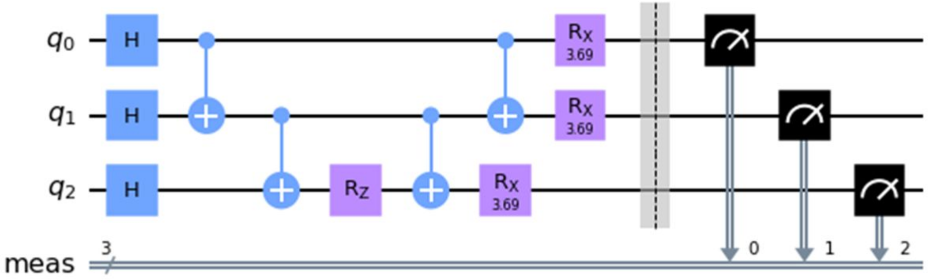
3-body term

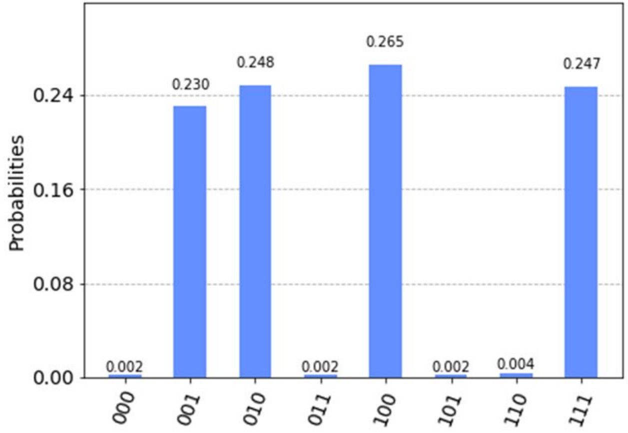
$$H_2 = \begin{pmatrix} 1 & s_1 \\ s_2 & s_3 \end{pmatrix}$$

$E(H_2) = (s_1+s_2s_3)^2$
$= 1 + 1 + 2s_1s_2s_3$
$= 2(\ 1 + \mathbf{s_1\, s_2 s_3})$
Þ$s_1 = +1$, $s_2 = +1$, $s_3 = -1$ or
Þ$s_1 = +1$, $s_2 = -1$, $s_3 = +1$ or
Þ$s_1 = -1$, $s_2 = +1$, $s_3 = +1$ or



$U(C,\gamma) = e^{-i\gamma C}$ $U(C,\gamma) = e^{-i\gamma C}$





4-body term

$$H_2 = \begin{pmatrix} s_1 & s_3 \\ s_2 & s_4 \end{pmatrix}$$

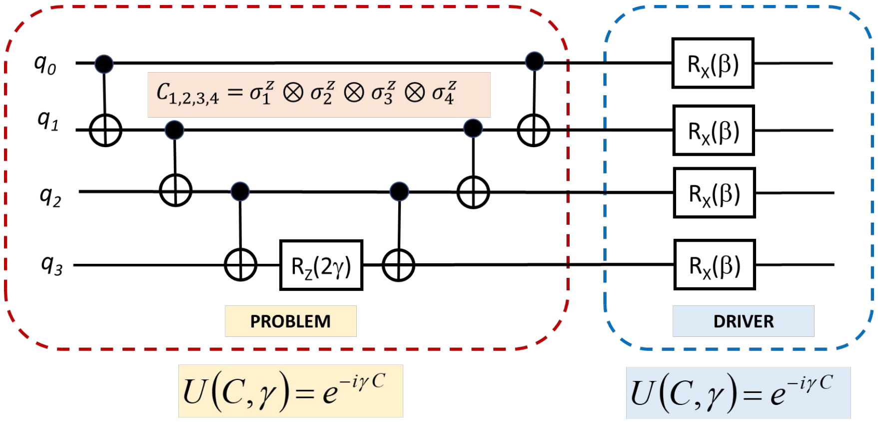
$E(H_2) = (s_1s_3+s_2s_4)^2$

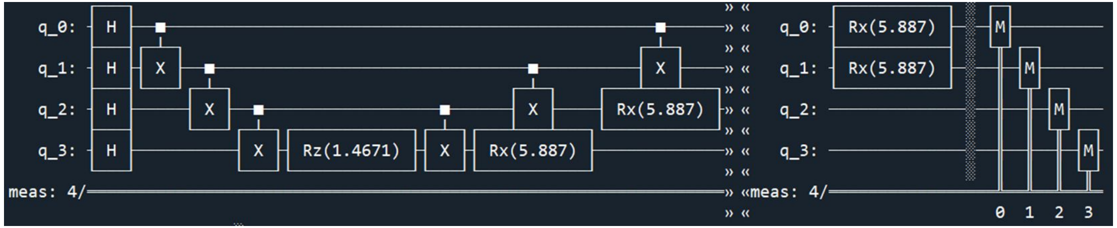
$= s_1^2\, s_3^2 + s_2^2\, s_4^2 + 2s_1s_2s_3s_4$

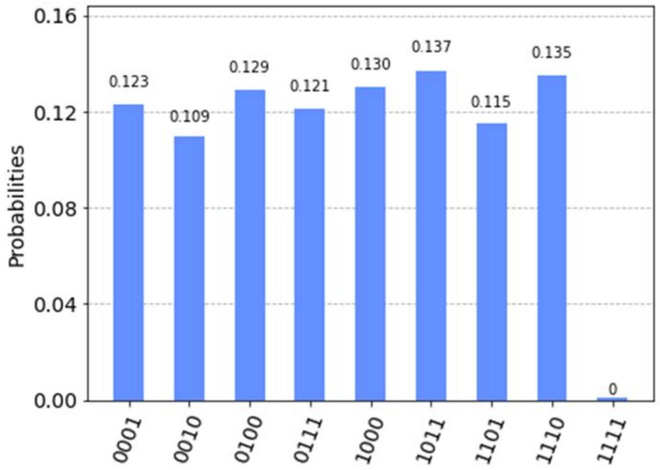
$= 1 + 1 + 2s_1s_2s_3s_4$

$= 2(\, 1 + \mathbf{s_1s_2s_3s_4})$

$\Rightarrow s_1s_2s_3s_4 = -1$

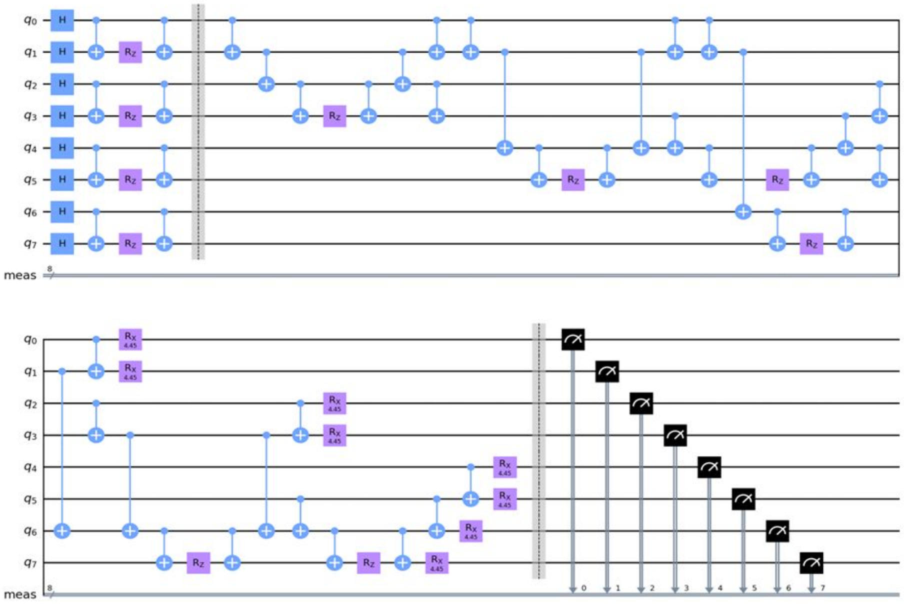


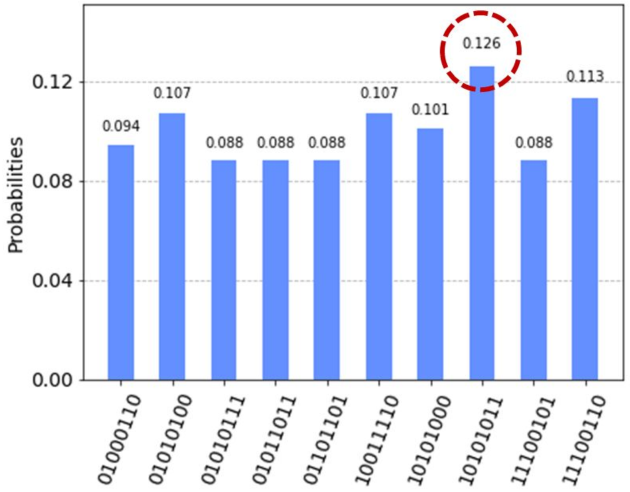




Finding Order-12/36: number of qubits=8

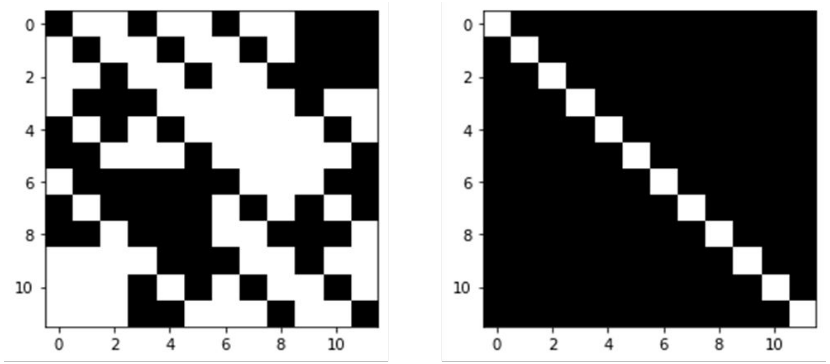
$$\hat{H}\left(\hat{\sigma}\right) = 2\hat{\sigma}_0^z\hat{\sigma}_1^z + 2\hat{\sigma}_2^z\hat{\sigma}_1^z 3 + 2\hat{\sigma}_4^z\hat{\sigma}_5^z + 2\hat{\sigma}_6^z\hat{\sigma}_7^z + \hat{\sigma}_0^z\hat{\sigma}_1^z\hat{\sigma}_2^z\hat{\sigma}_3^z + \hat{\sigma}_0^z\hat{\sigma}_1^z\hat{\sigma}_4^z\hat{\sigma}_5^z + \hat{\sigma}_0^z\hat{\sigma}_1^z\hat{\sigma}_6^z\hat{\sigma}_7^z + \hat{\sigma}_2^z\hat{\sigma}_3^z\hat{\sigma}_4^z\hat{\sigma}_5^z + \hat{\sigma}_2^z\hat{\sigma}_3^z\hat{\sigma}_6^z\hat{\sigma}_7^z + \hat{\sigma}_4^z\hat{\sigma}_5^z\hat{\sigma}_6^z\hat{\sigma}_7^z + 4$$



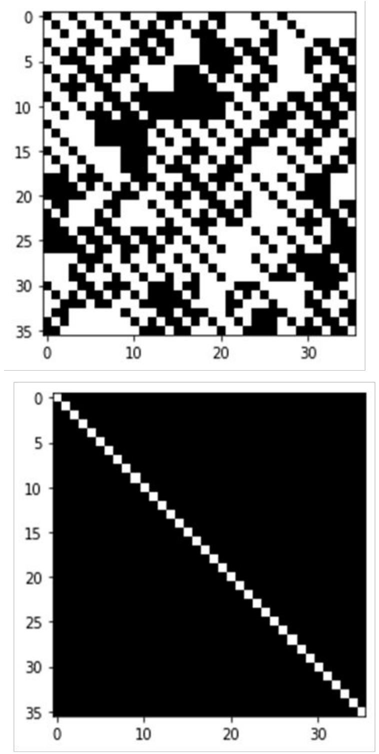


- 10 most common solutions
- Most frequent='10101011'

Results: Williamson 12 and Baumert-Hall 36



Williamson 12

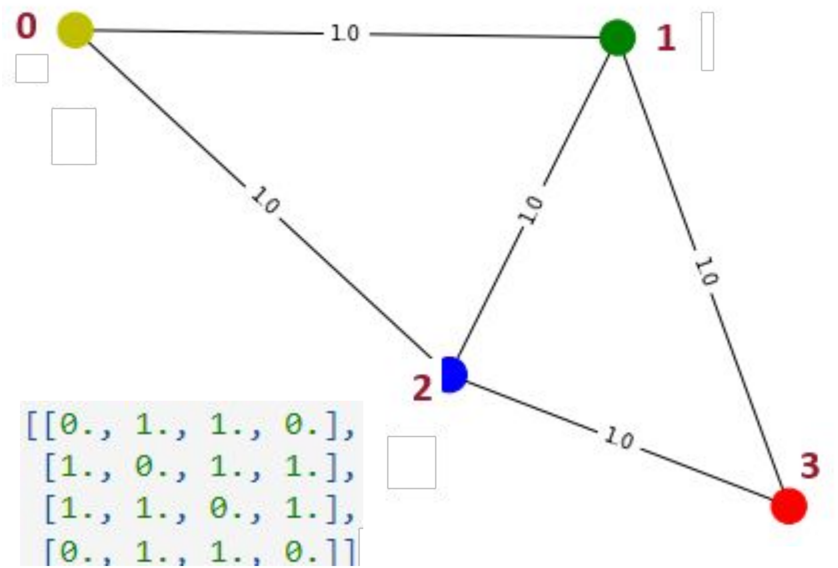


Baumert-Hall 36

End of Section

Appendix-1

Example Maxcut-2:



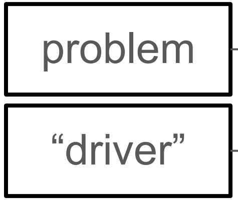
- The brute-force method:
  - exhaustively try all the binary assignments.
- Quantum computing
  - Translate into Ising model
  - Solve the problem

Qiskit Output

- Objective value computed by the brute-force method is 3
- QAOA-Sol: [1 1 0 0]
- Objective value computed by QAOA is 3

https://github.com/SophiaZhyrovetska/qaoa-maxcut

https://github.com/Qiskit/qiskit-tutorials/blob/master/tutorials/algorithms/05_qaoa.ipynb



$$\hat{H}_C$$
$$\hat{H}_B$$

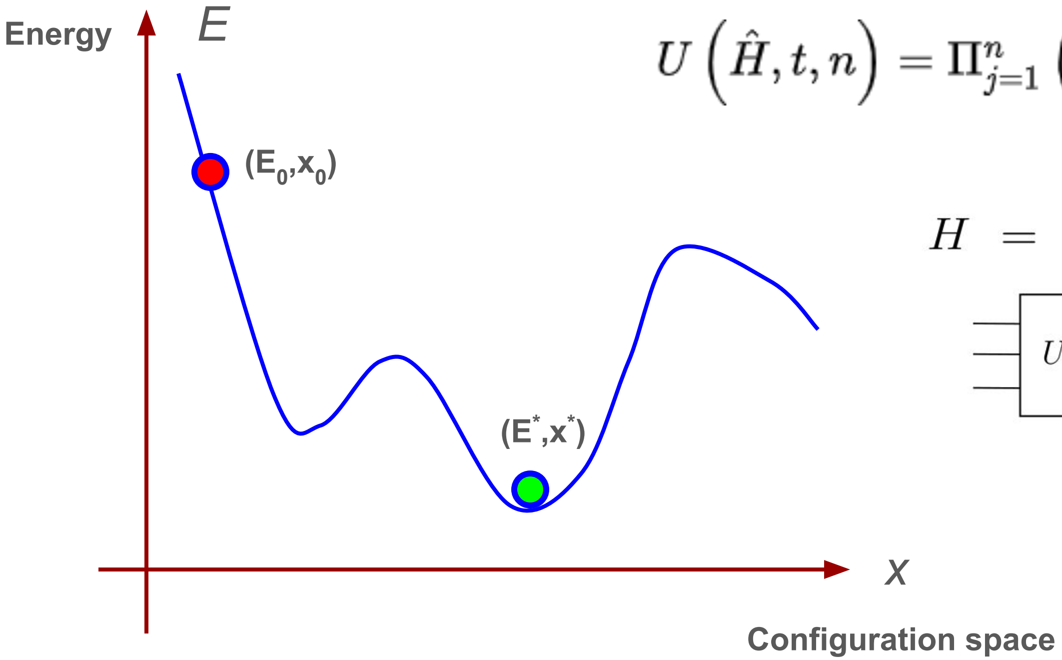
$$\hat{H} = \hat{H}_C + \hat{H}_B$$

$$U\left(\hat{H}, t\right) = e^{-i\hat{H}t/\hbar}$$

Trotter-Suzuki theorem

$$e^{\hat{A}+\hat{B}} \approx \left(e^{\hat{A}/n} e^{\hat{B}/n}\right)^n$$

$$U\left(\hat{H}, t, n\right) = \Pi_{j=1}^{n} \left(e^{-i\hat{H}_C t/n} e^{-i\hat{H}_B t/n}\right)$$



$$H = H_1 + H_2 + \cdots + H_N$$