Combinatorial Optimization on Quantum Computers

Yuri Alexeev, Argonne National Laboratory
Ilya Safro, Clemson University
Ruslan Shaydulin, Clemson University

CLEMSON UNIVERSITY

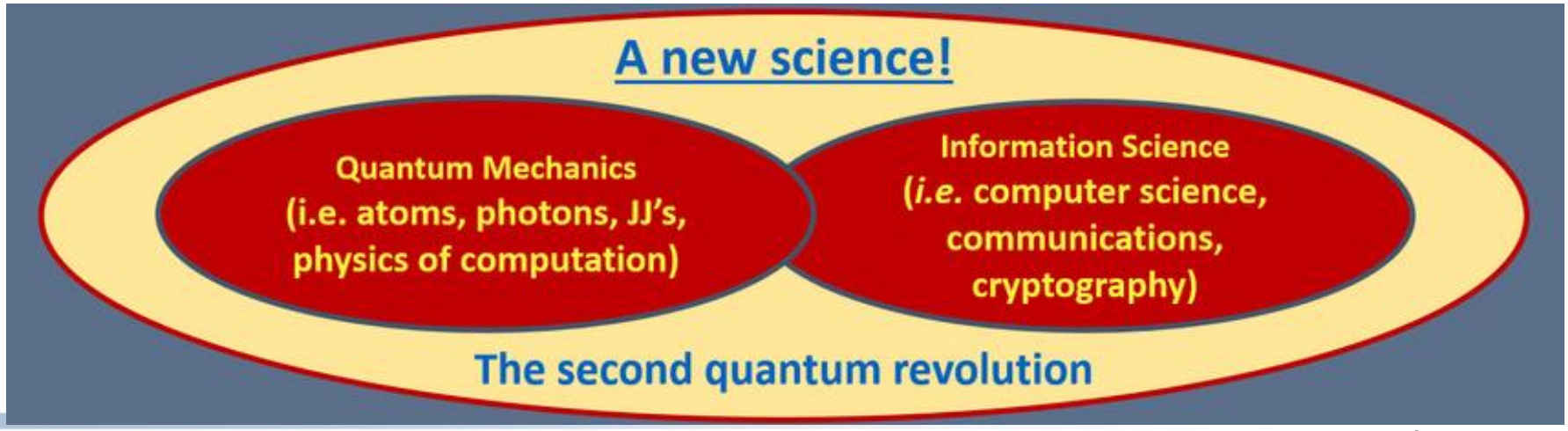
Slides and notebooks

- To download ZIP archive, go to http://bit.ly/PP20quantum
- If you prefer Git: https://github.com/rsln-s/SIAM_PP_20_minitutorial

- If you do not have an IBM account, go to quantum-computing.ibm.com to create one

What is Quantum Information Science?

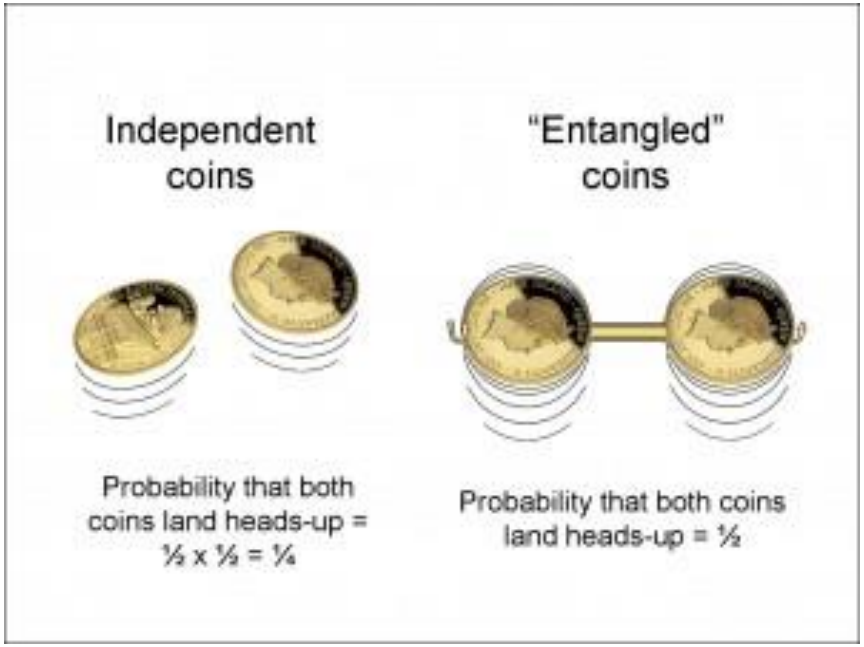
- Quantum mechanics explains how world works at microscopic level, which governs behavior of all physical systems, regardless of their size. Arguably one of the greatest scientific discoveries on 20th century, leading to fundamental discoveries of how nature works
- Information science revolutionized how information is collected, stored, analyzed, manipulated, moved, protected, and moved. It ushered new age of information in 20th century, with major repercussions in economic, social, and political spheres
- We see convergence of two 20th century greatest revolutions in the form of Quantum Information Science (QIS)



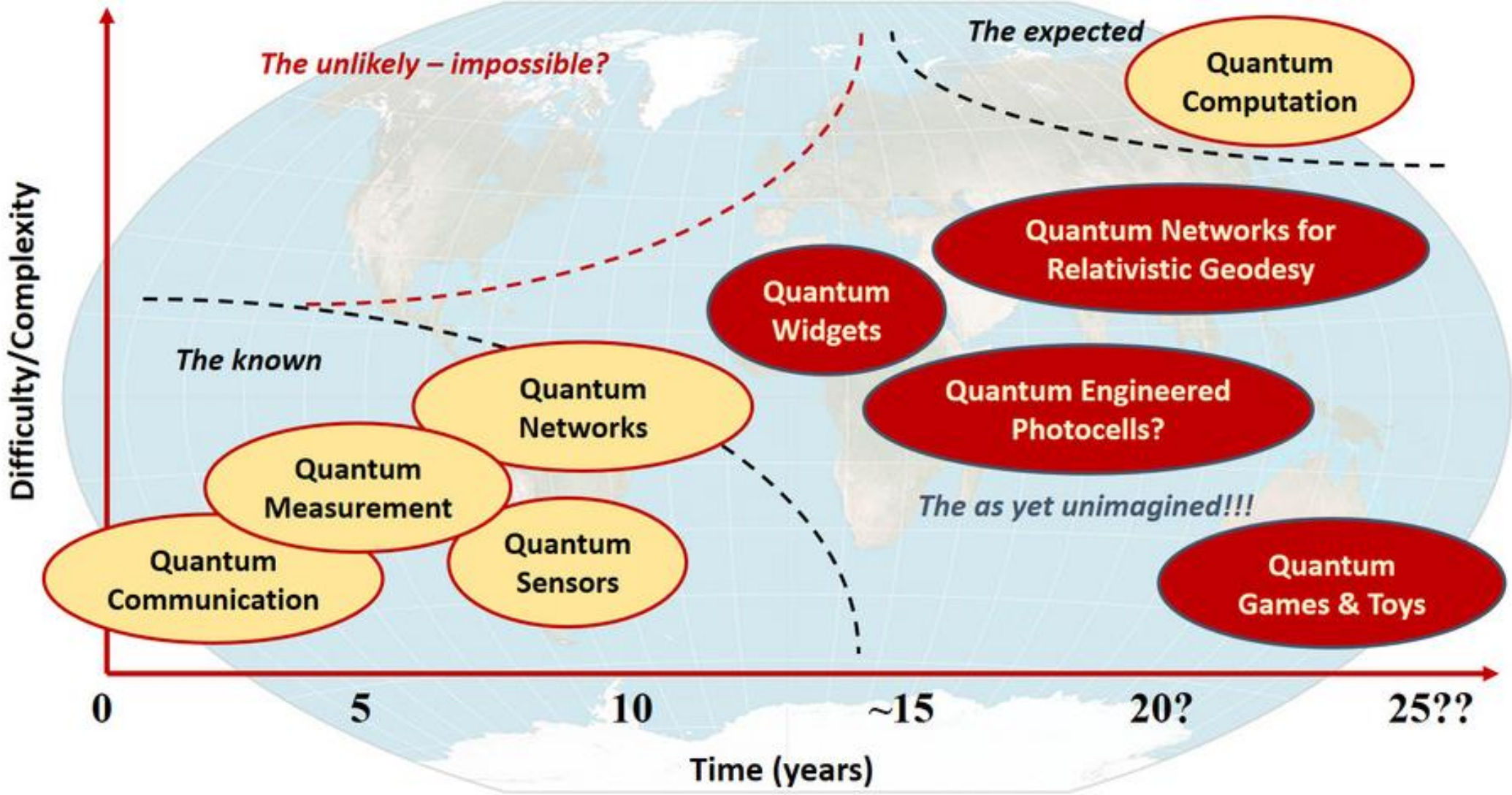
Credit: NIST

Quantum Information Science

- **QIS exploits unique quantum effects such as superposition, coherence, and entanglement to obtain, compute, and transmit information in the ways that are superior compared to classical technology (digital, Newtonian)**
- **The key concept is entanglement ("spooky action at a distance", EPR pair). Works only for only very small object (electrons, photons, atoms etc). It is proven to be essential to achieve "quantum advantage" or for "quantum teleportation"**



Quantum Applications



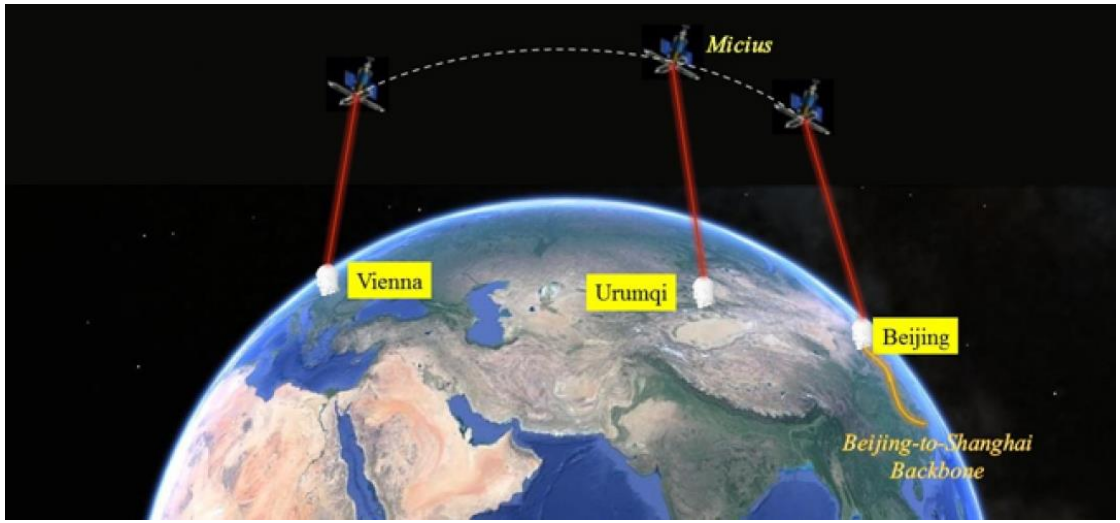
Credit: NIST

Quantum Applications

- Secure telecommunication (near future): quantum key distribution is an ultra-secure communication method that requires a key to decipher a message. If the message gets intercepted, no one else can read it.
- Quantum sensors (not far future): devices that exploits quantum entanglement to achieve a sensitivity or resolution that is better than can be achieved using only classical systems. Applications: astrophysics, high energy physics, military (quantum radars) etc.
- Scientific problems (far future): a number of NP-hard combinatorial problems can be solved efficiently on quantum computers (i.e. linear algebra and database search). Applications: quantum chemistry, traffic control, real-time risk analysis, financial, and forecasting etc.

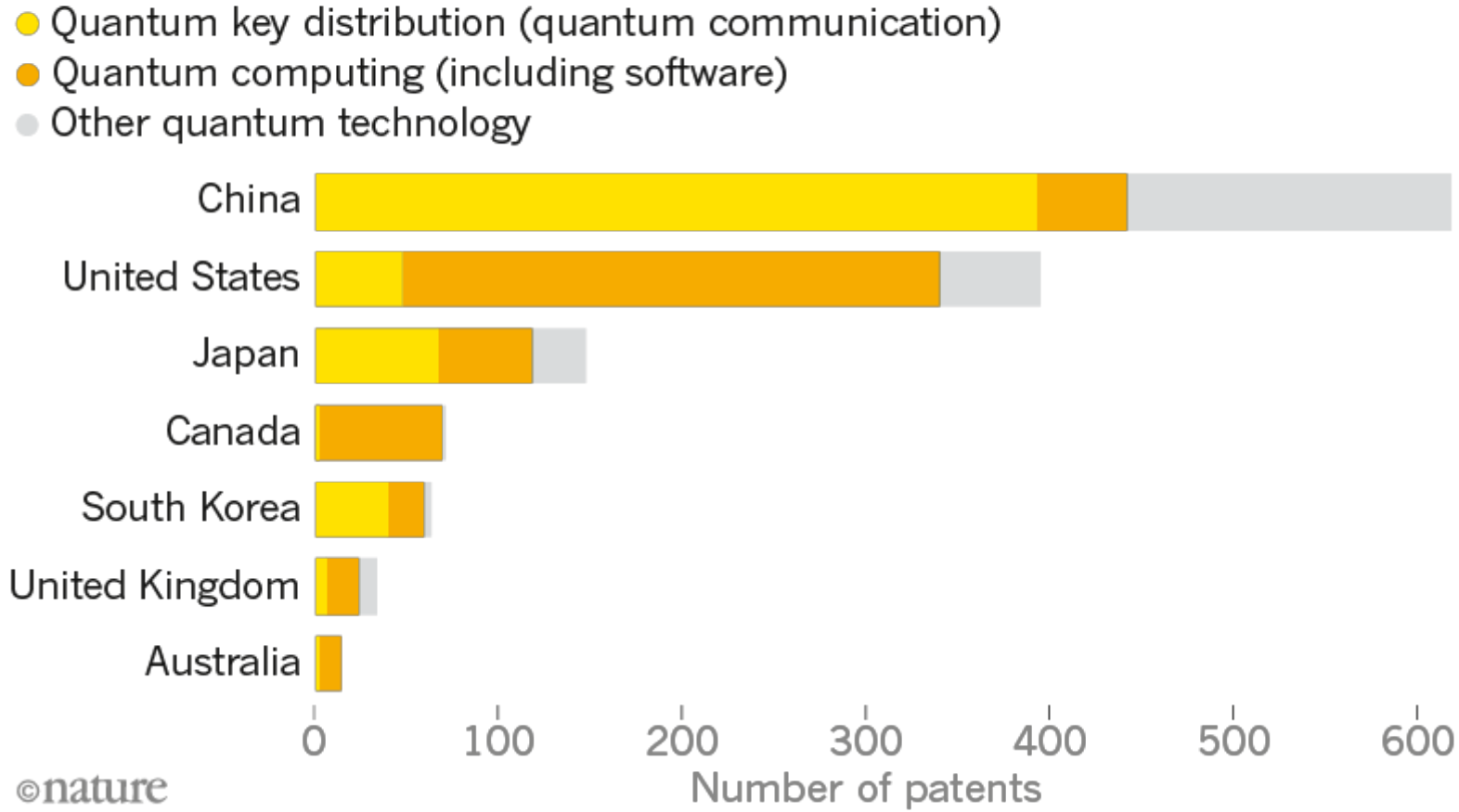
Quantum Teleportation

- Chinese scientists set the record for the farthest quantum teleportation in 2017. It demonstrated that quantum teleportation works over distance of 870 miles. It paves the way to quantum internet
- Quantum-encrypted images by encoding them as strings of numbers based on the quantum states of photons and sent them across distances of up to 4,722 miles between Beijing and Vienna in 2018
- The best way to distribute quantum entanglement around the globe is via a massive constellation of orbiting satellites according to just published MIT report
- US does not have currently satellite-based effort for quantum internet
- UChicago – Argonne – Fermilab 30 mile quantum network is under construction



Quantum patents

An analysis of global patents in quantum technology since 2012 shows China dominating quantum communication, but North America ahead on quantum computing.

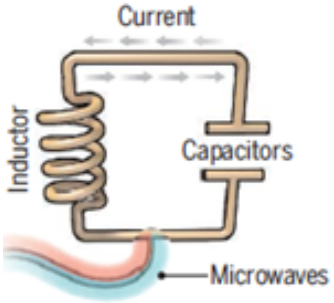


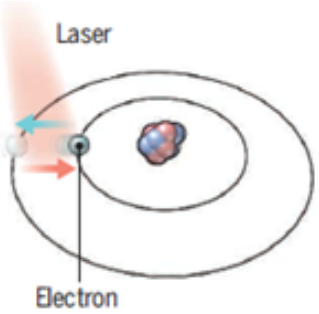
Publicly Announced National and International Initiatives in QIS

Nation	Initiative	Year Launched	Investment, Time frame	Scope
US	US Quantum Initiative	2020	Up to $625 over 5 years	National centers of QIS excellence (up to five centers for up to $25M/year)
China	National Laboratory for QIS	2018	$11.4B	Centralized quantum research facility
EU	Quantum Technologies Flagship	2018	$1.1B over 10 years	Quantum communication, metrology and sensing, simulation, computing, and fundamental science
Russia	Russian Quantum Initiative	2019	$780M	Quantum computers, quantum computing, quantum communication, quantum sensors
Germany	German Quantum Imitative	2019	$712M	Quantum computers, quantum computing, quantum communication, quantum sensors
UK	UK National Quantum Technologies Program	2014	$358M over 5 years	Sensors and metrology, quantum enhanced imaging, quantum communications technologies

Modern Quantum Computers

Operate at almost absolute zero temperature -460 F or -273 C, colder than deep space



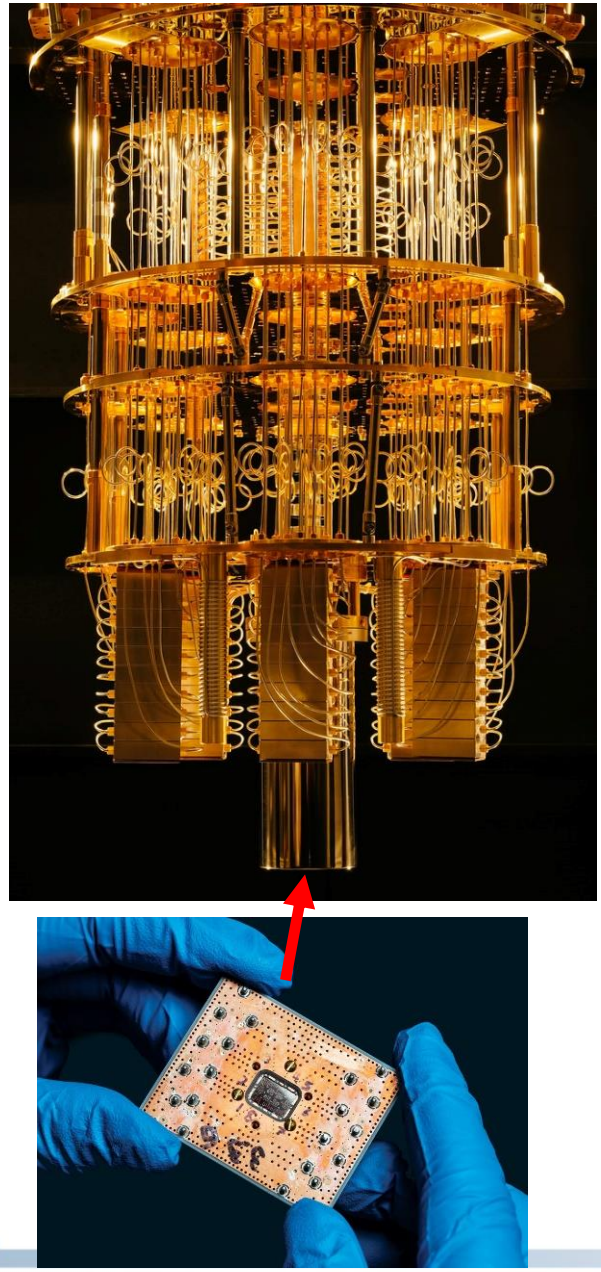


Computers are ranked by number of qubits decoherency time

		Superconducting (IBM, Google, Rigetti)	Trapped ions (IonQ, U. of Innsbruck)
Qubit Modality	Materials	Al on the Si substrate	Yb+, Ca+, Sr+, Be+, Ba+, Mg+
	Type	Transmon	Optical transitions
	Control	Microwaves	Microwaves+optics
	State	Junction phase	Atomic state of electron
Approximate Decoherency Times (ns)		~100-200	Very long
	1qb gate	10	5,000
	2qb gate	40	50,000
Fidelity	1qb gate	99.9%	99.999%
	2qb gate	99.0%	99.5%
Speed (MHz)	1qb gate	100.00	0.20
	2qb gate	25.00	0.02

Modern CPUs:
~3 GHz, 100% fidelity

IBM quantum computers



The key piece of the Quantum Computer is the Dilution Refrigerator
Working Temperature 15 mK uses mix of $^{3}He/^{4}He$

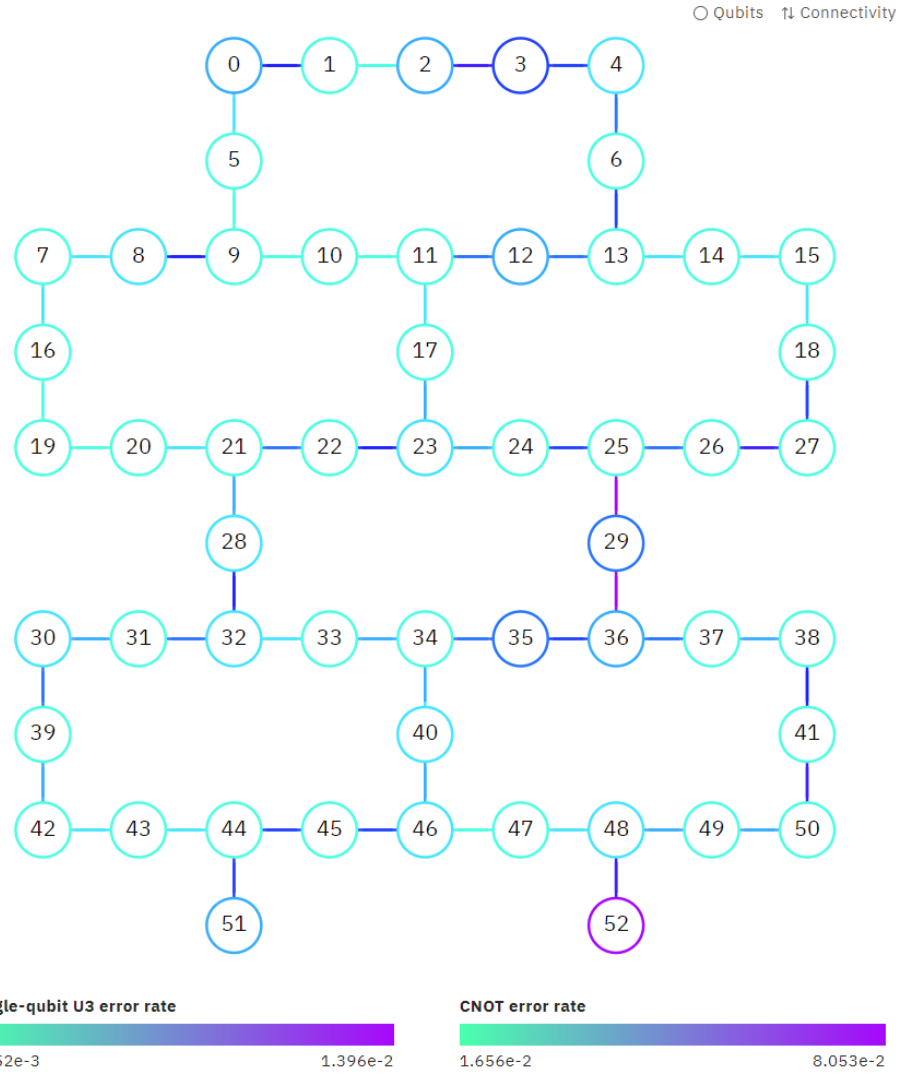
Source: IBM Research

IBM quantum computers

Everybody can get access through cloud (IBM Quantum Experience):
https://quantumexperience.ng.bluemix.net/qx/signup

Available devices (public and Q hub):

Qubits in device	Number of devices	Public
5	6	6
15	1	1
20	4	0
53	1	0

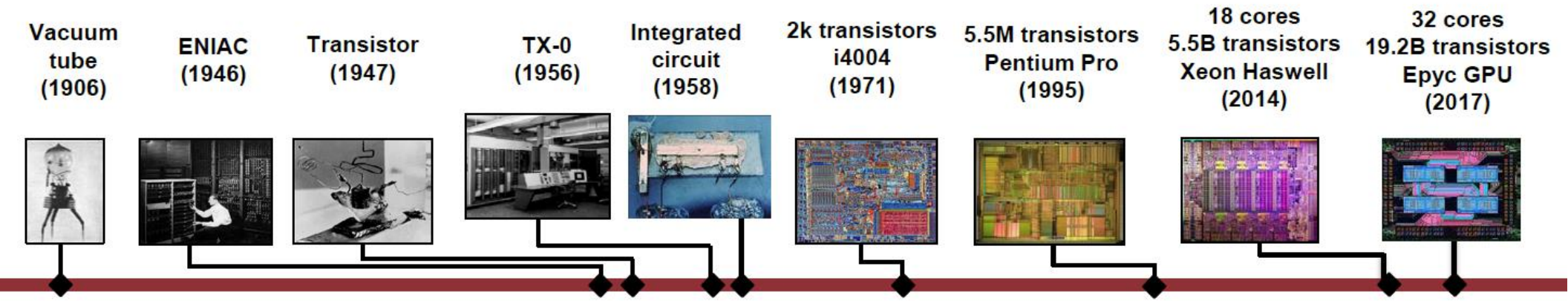


Reality check

- We have 53 noisy qubits (need millions for computation)
- Short decoherency time limits execution up to 10-100 gates maximum (need millions)
- Slow gates MHz (need GHz)
- Poor connectivity (for superconducting quantum computers)
- Slow I/O
- Quantum Winter II for Quantum Computing in future?

Hardware timeline

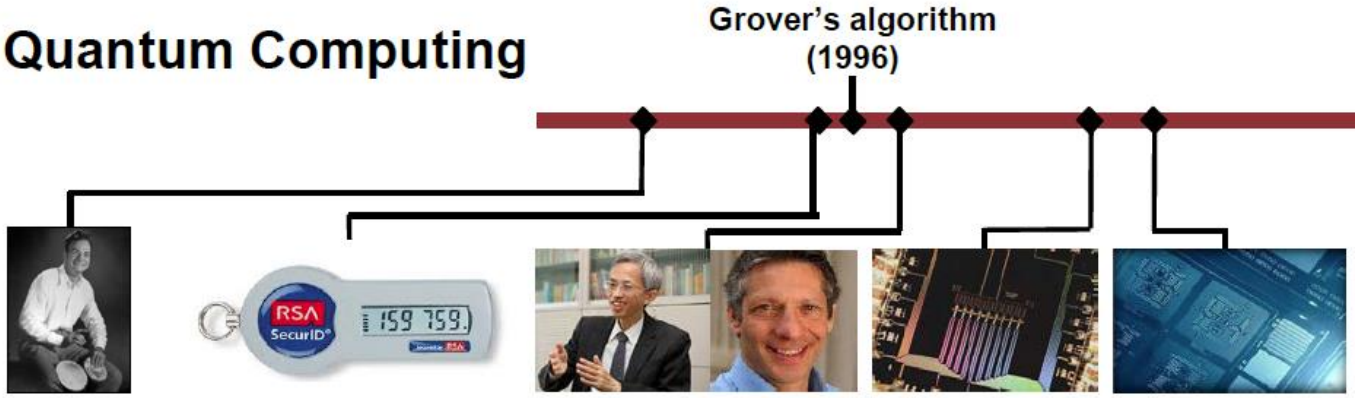
Classical Computing (Electronic)



Quantum computing is transitioning from scientific curiosity to technical reality.

Advancing from discovery to prototype to useful machines takes time.

Quantum Computing

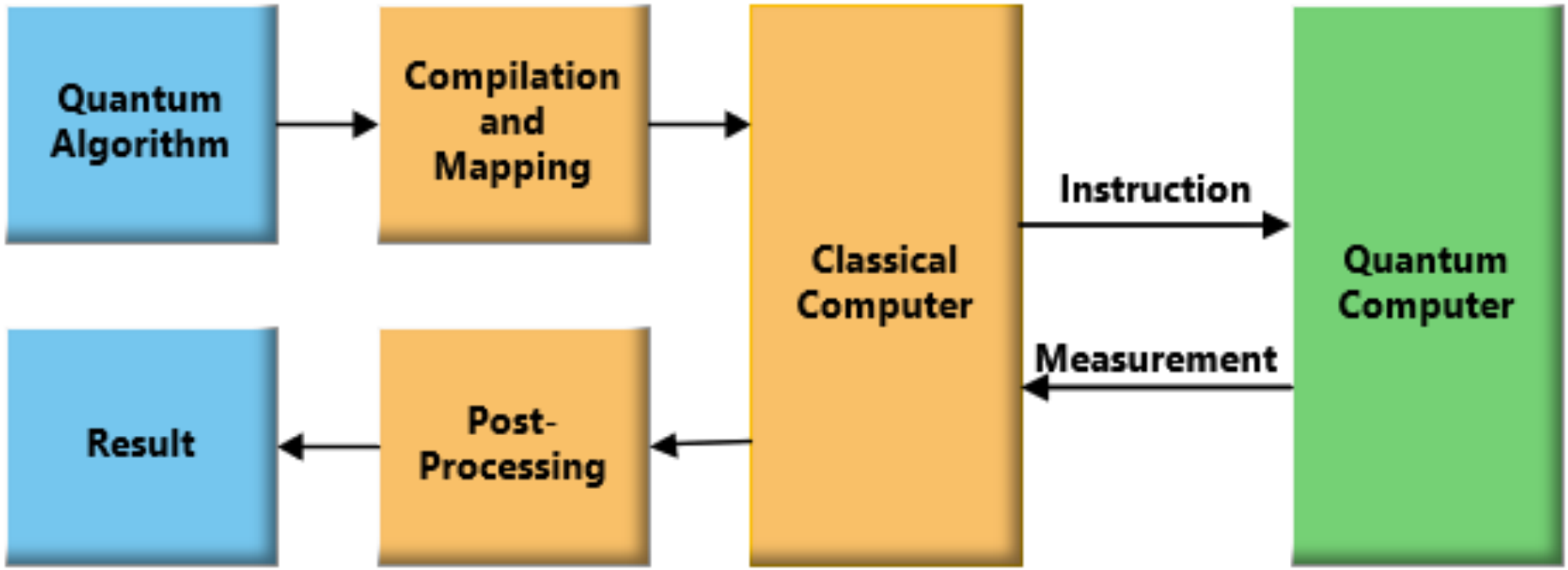


Credit: MIT

Available and announced quantum computers

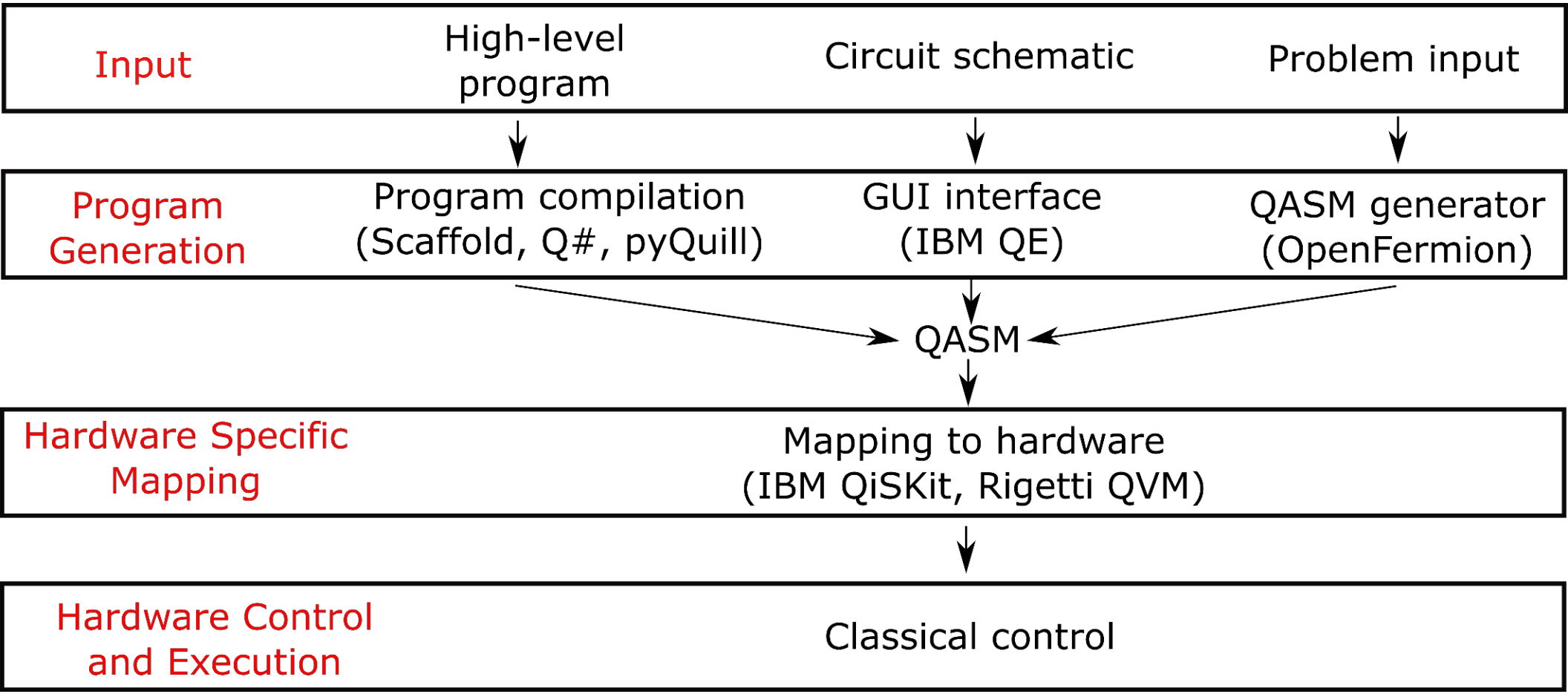
Company	Operational	Cloud Access	Framework	Announced
IBM	53 qubits	Open to Q hub members	QisKit	100+ qubit in 2021?
Rigetti	19 (8) qubits	Access by request	Forest	50+ qubits in 2020?
Google	53 qubits	No access	Cirq	72 qubit chip announced March 2018
Intel	?	No access	?	49 qubit chip announced Jan. 2018
Alibaba	11 qubits	?	Aliyun	
IonQ	11 qubits	No access	AWS and Azure	
D-Wave	2000Q (~60 qubits)	Open (1 minute per month)	Leap	5000Q announced in 2019

Hybrid Quantum/Classical Computing System

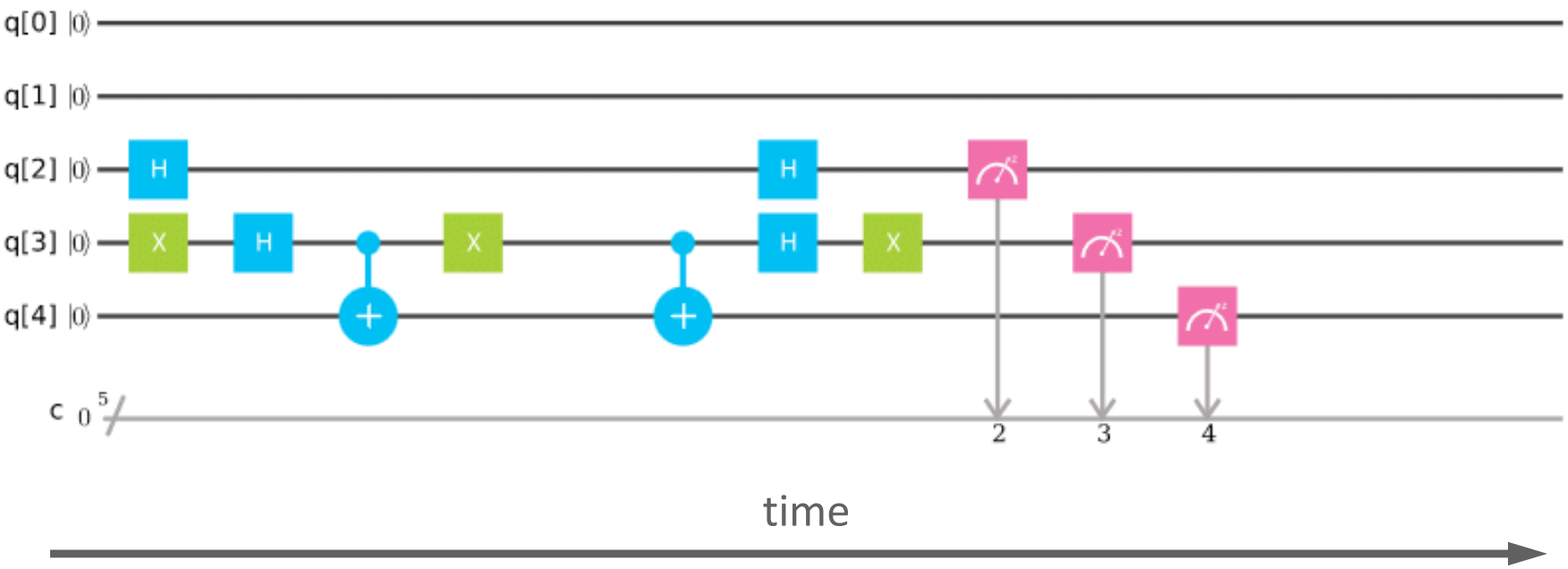


A high-level block diagram of a quantum computing system, where colors represent different levels of abstractions. Typically three levels are involved: a user level (blue), classical computation and control (yellow), and QC system (green). A quantum algorithm is compiled and mapped into a native set of instruction for the target quantum computer. The measurement of quantum register after postprocessing becomes the result.

Quantum software stack



Quantum circuits

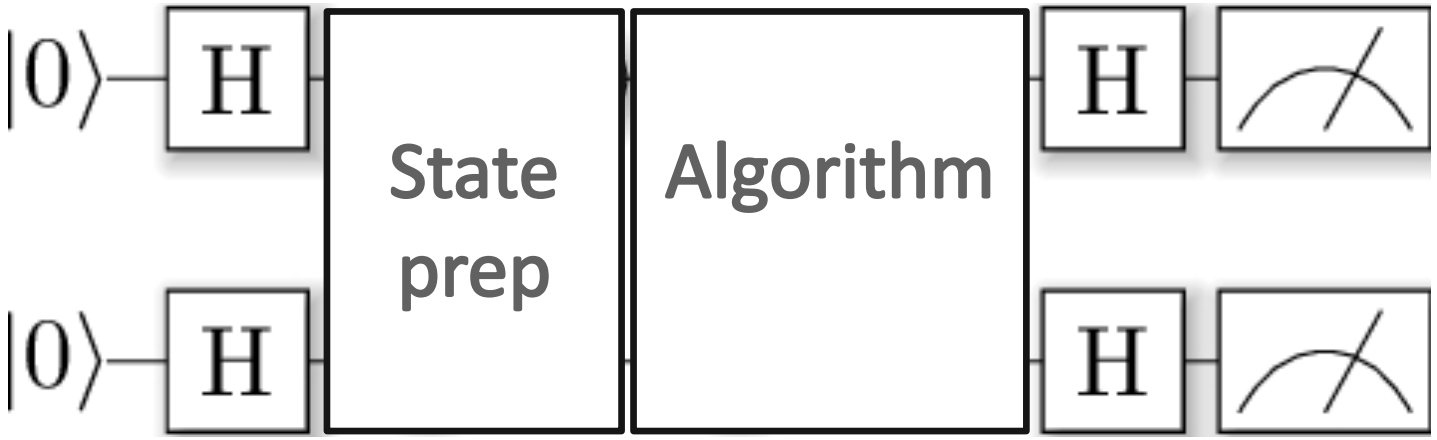


Quantum computers are programmed by using Quantum Assembly (QASM)
QASM standards: MIT QASM, IBM OpenQASM 2.0, LANL QASM, Atos QASM etc.

Qbit q2	X q3
Qbit q3	CNOT q3, q4
Qbit q4	H q2
H q2	H q3
X q3	X q3
H q3	measure q0
CNOT q3, q4	measure q1

A typical structure of a quantum algorithm

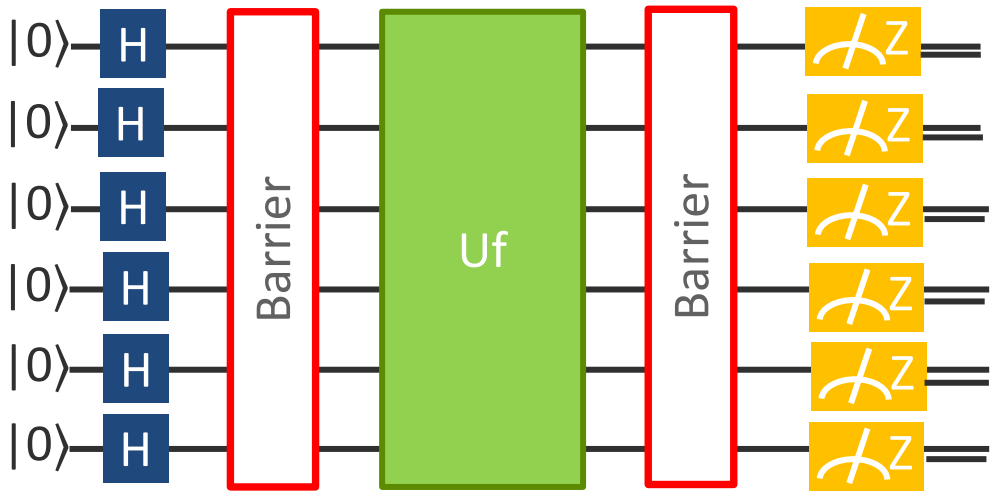
Initialize → Maximum superposition → State preparation → Logic Gates → Minimum superposition → Measurement



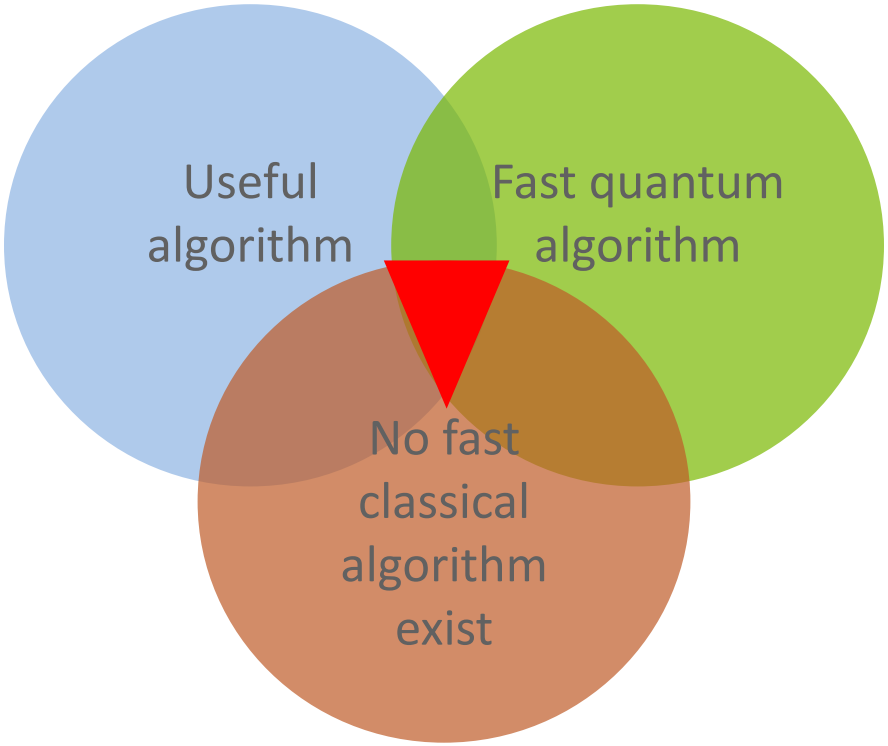
- First part of the algorithm is to make an equal superposition of all 2^n states by applying H gates
- The second part is to encode the problem into this states;
put phases on all 2^n states
- In the third part interfere all these states back to a few outcomes containing the solution

Quantum Parallelism

- Quantum Parallelism is a fundamental feature of quantum computers to get speedup compared to classical computers
- Run Uf program once and get results for all possible inputs
- But when measure I can learn value f(x) only for one input x
- It exploits entanglement to change many simultaneously amplitudes with one gate. It allows efficiently remove bad bit strings and amplify correct bit string (destructive/constructive interference)



Quantum Algorithms



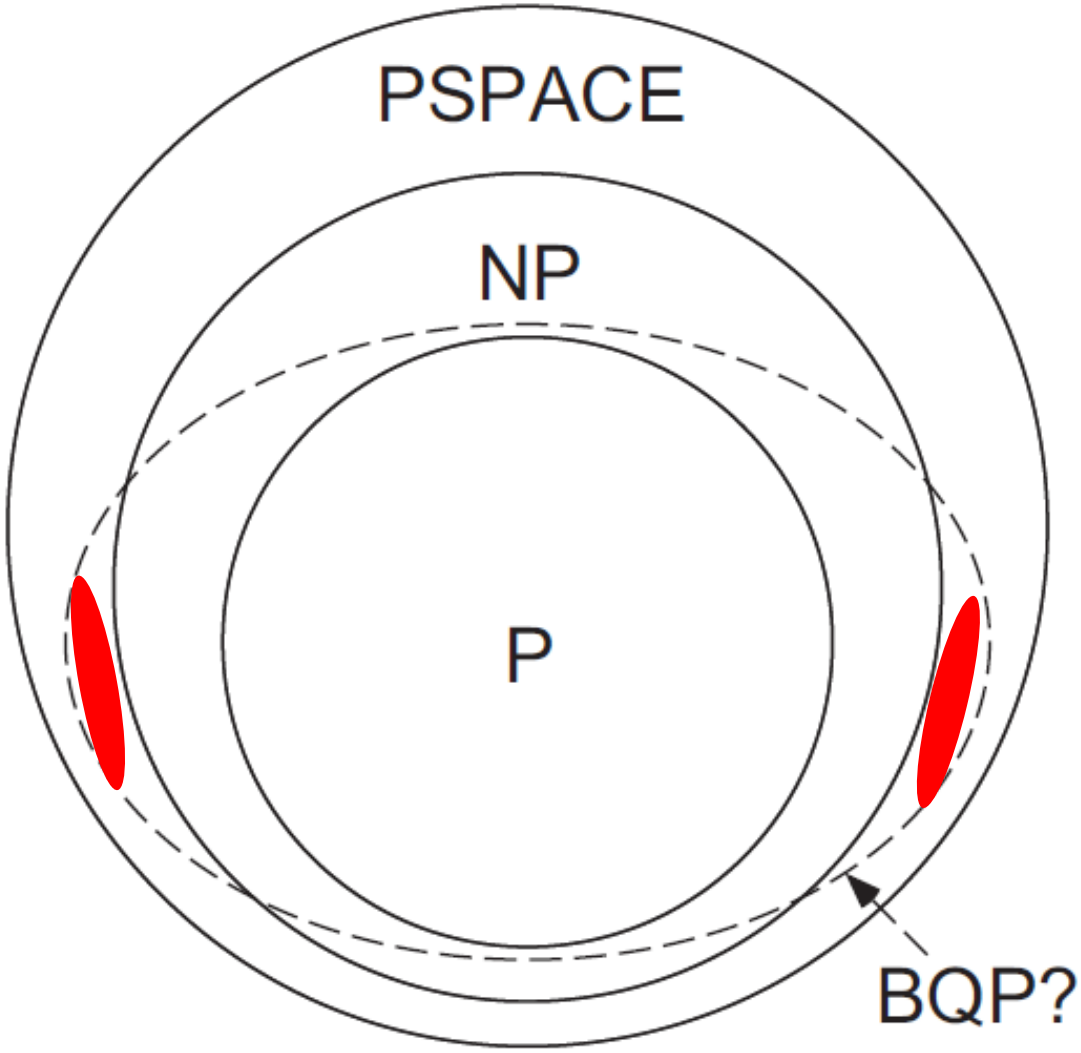
- There are two known classes algorithms hitting all three circles:
- Four main fundamental algorithms expected to provide a speedup over their classical counterparts: Shor's factoring algorithm, Grover's search algorithm, HHL's linear system solver, and quantum simulation
- Quantum machine learning?

Quantum Algorithms

Algorithm	Classical resources	Quantum resources	Quantum speedup	Requirements
Quantum simulation	2^N	$\sim N^6$	Exponential	100+ qubits, millions of gates
Factorization	2^N	N^3	Exponential	200+ qubits, millions of gates
Solving linear systems	N^2	Log(N)	Exponential	Millions of gates and qubits
Unstructured search	N	$\sqrt{N}$	$\sqrt{N}$	Millions of gates and qubits

N-complexity of the problem

The Power of Quantum Computation

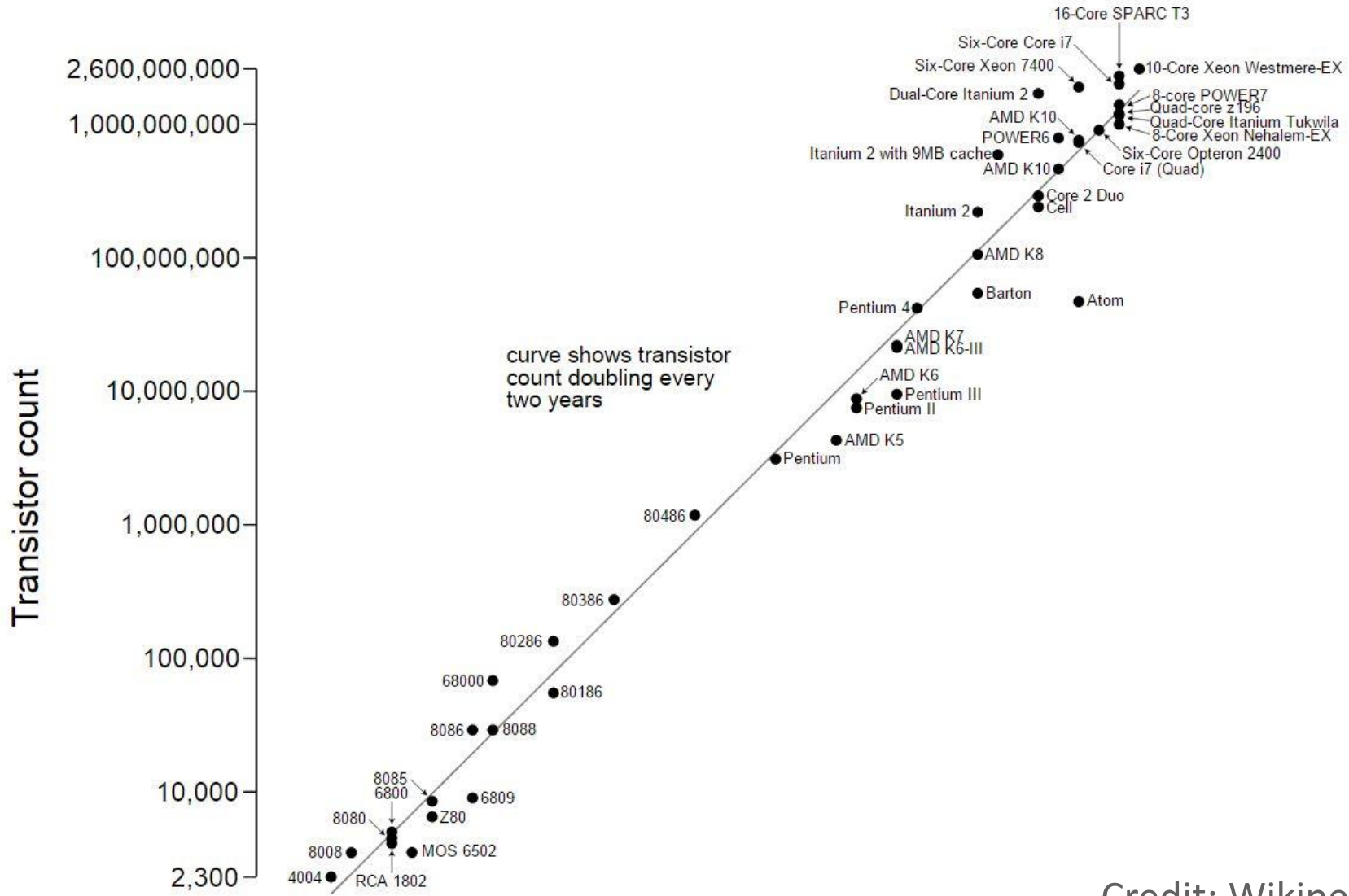


P = solved in polynomial time
NP = verified in polynomial time
PSPACE = solved in polynomial space

We do not know whether
P != NP
PSPACE is bigger than NP

BQP (Bounded error Quantum Polynomial time) is the class of decision problems solvable by a quantum computer in polynomial time, with an error probability of at most 1/3 for all instances

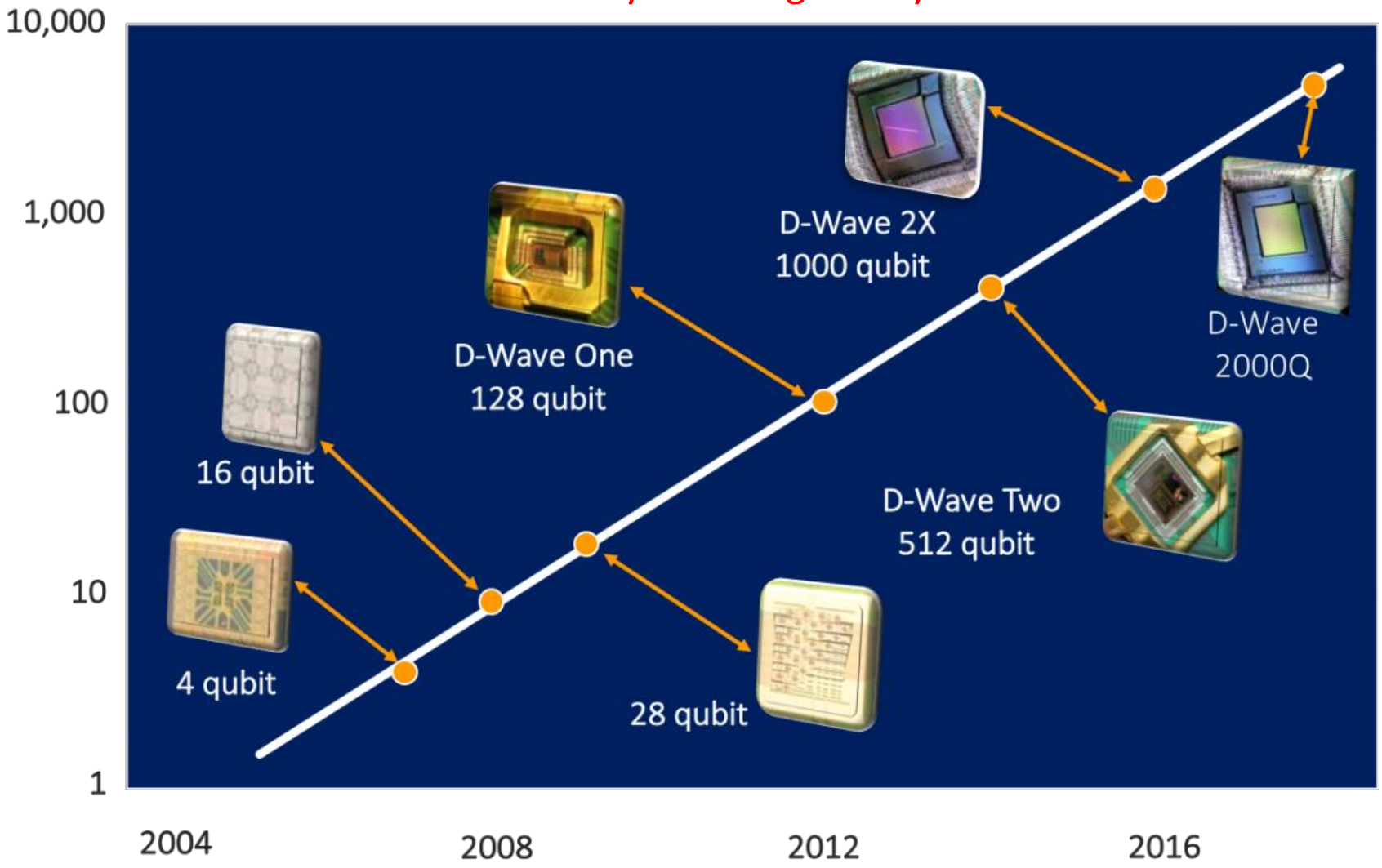
Moore's Law



Credit: Wikipedia

Rose's Law

For the past 10 years, the number of qubits on D-Wave's QPUs has been steadily doubling each year



Credit: D-Wave

Upcoming Events

- SIAM PP20 Quantum Workshop:

Recent Advances and Trends in Hybrid Quantum-Classical Algorithms - Part I and II
Thursday, February 13, 10:55 AM - 12:35 PM, Room: 605
Thursday, February 13, 3:20 PM - 5:00 PM, Room: 605

- APS 2020 March meeting presentations:

- Reinforcement Learning for Finding QAOA Parameters

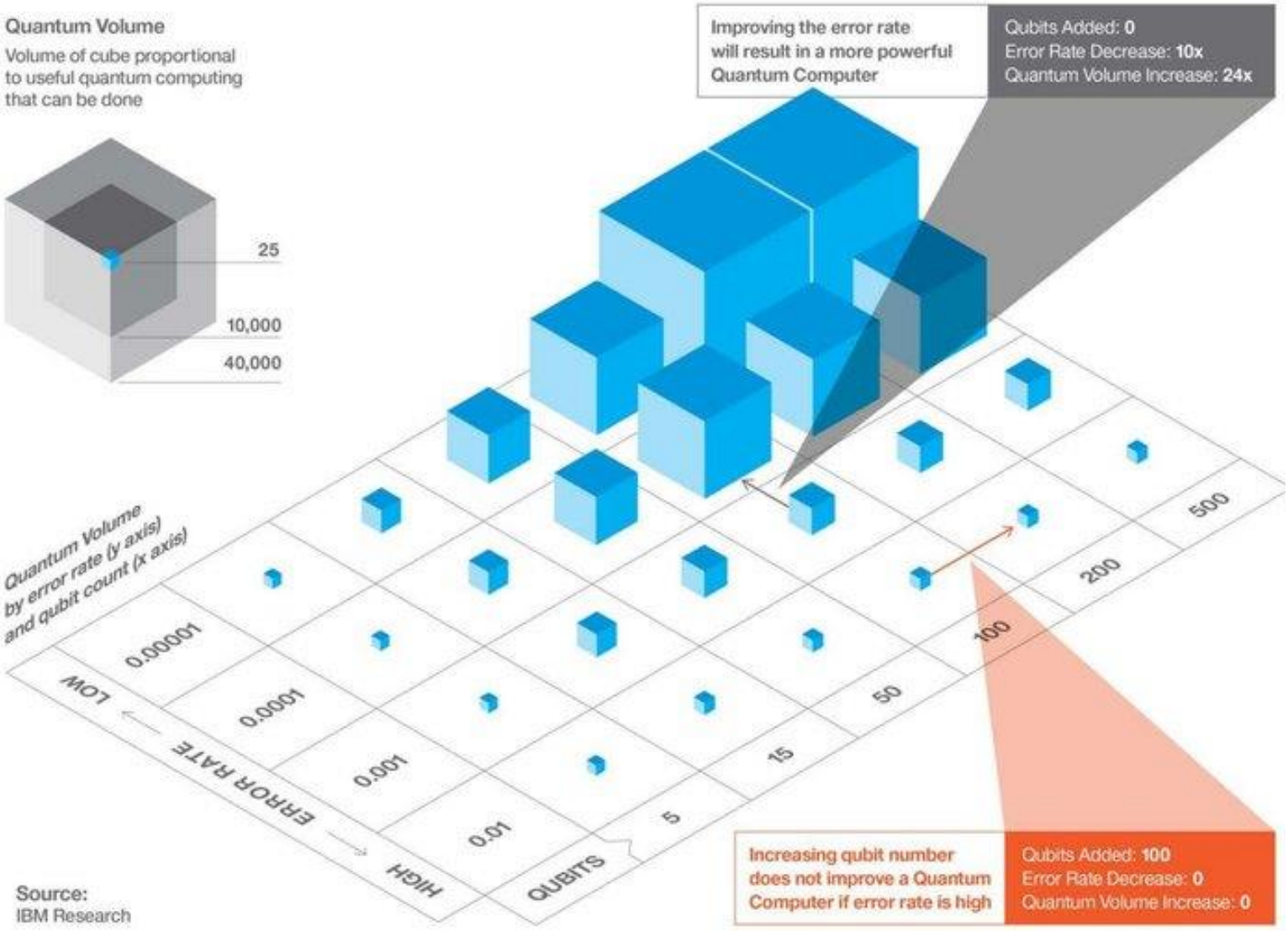
- Running large quantum circuits on small quantum computers

- Argonne Quantum Tutorial 3, May 21
- Oak Ridge Quantum User Forum, April 21-23
- SC20 Quantum Workshop:
  "1st International Workshop on Quantum Computing Software"

Acknowledgments

- This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357

Quantum Volume



Quantum Simulators

Simulator	Advantages	Disadvantages
Intel-QS	highly scalable C++ HPC code (MPI/OpenMP), freely available from Git	under development, no documentation, lacking sophisticated error models
ProjectQ	easy to use Python code, freely available from Git, works with OpenFermion	no MPI implementation, lacking documentation, lacking error models
QuaC	time dynamics, scalable code, freely available from Git, error models	under development, poor documentations, depends on PETSc
Atos	robust commercial package, easy to use, excellent documentation, error models	not freely available, no MPI implementation

Simulating Quantum Computers On Classical Computers

- Simulating a quantum gate acting on N qubits needs $O(2^N)$ memory and operations

Qubits	Memory	Time per operation
10	16 KB	Microseconds on a smartwatch
20	16 MB	Milliseconds on a smartphone
30	16 GB	Seconds on a laptop
40	16 TB	Seconds on a PC cluster
50	16 PB	Minutes on modern supercomputers
60	16 EB	Hours on post-exascale supercomputers?
70	16 ZB	Days on supercomputers in distant future?

Combinatorial Optimization on Quantum Computers

Ruslan Shaydulin, Clemson University

CLEMSON UNIVERSITY

Outline

- Introduction
- Part 1: A Very Brief Introduction to (gate model) quantum computing
- Part 2: Quantum Approximate Optimization Algorithm (QAOA)
- Part 3: Hands-on

Outline

- Introduction
- Part 1: A Very Brief Introduction to (gate model) quantum computing
- Part 2: Quantum Approximate Optimization Algorithm (QAOA)
- Part 3: Hands-on

Warning: the goal is to prepare you to start applying QAOA to your problems, so **we will get technical**

Why Computer Science is a Science?

- Turing machine links abstract notion of "computability" to physical world
- Church-Turing Thesis: "anything that is computable in the physical world is computable by a Turing machine"

Why Computer Science is a Science?

- Turing machine links abstract notion of "computability" to physical world
- Church-Turing Thesis: "anything that is computable in the physical world is computable by a Turing machine"
- Alternatively (equivalently): "**Any bounded physical system can be simulated by a Turing machine, to any desired precision**"

Why Computer Science is a Science?

- Turing machine links abstract notion of "computability" to physical world
- Church-Turing Thesis: "anything that is computable in the physical world is computable by a Turing machine"
- Alternatively (equivalently): "**Any bounded physical system can be simulated by a Turing machine, to any desired precision**"

- Note that this doesn't mean we can do it efficiently!

Why Computer Science is a Science?

- Turing machine links abstract notion of "computability" to physical world
- Church-Turing Thesis: "anything that is computable in the physical world is computable by a Turing machine"
- Alternatively (equivalently): "**Any bounded physical system can be simulated by a Turing machine, to any desired precision**"

- Church-Turing thesis stands unchallenged

Extended Church-Turing Thesis (ECT)

- "feasibly computable in the physical world = efficiently computable by a deterministic Turing machine (in P)"

Extended Church-Turing Thesis (ECT)

- "feasibly computable in the physical world = efficiently computable by a deterministic Turing machine (in P)"

- Quantum computing is a counter-example to ECT!

Extended Church-Turing Thesis (ECT)

- "feasibly computable in the physical world = efficiently computable by a deterministic Turing machine (in P)"

- Quantum computing is a counter-example to ECT!
- Quantum supremacy experiment is an experimental demonstration that overthrows ECT (but still preserves the original one)

Extended Church-Turing Thesis (ECT)

- "**feasibly computable in the physical world = efficiently computable by a deterministic Turing machine (in P)**"

- Quantum computing is a counter-example to ECT!
- Quantum supremacy experiment is an experimental demonstration that overthrows ECT (but still preserves the original one)

- Quantum supremacy experiment was done on a problem **with limited practical applications**
- Next stage: a race to develop algorithms that leverage this novel hardware to **solve a practical problem faster** than a classical computer – Quantum Advantage

Algorithms with proven performance

Classical

- Matrix multiplication
- Dijkstra's algorithm for shortest path

Quantum

- Shor's algorithm for integer factoring
- Grover's algorithm for unstructured search

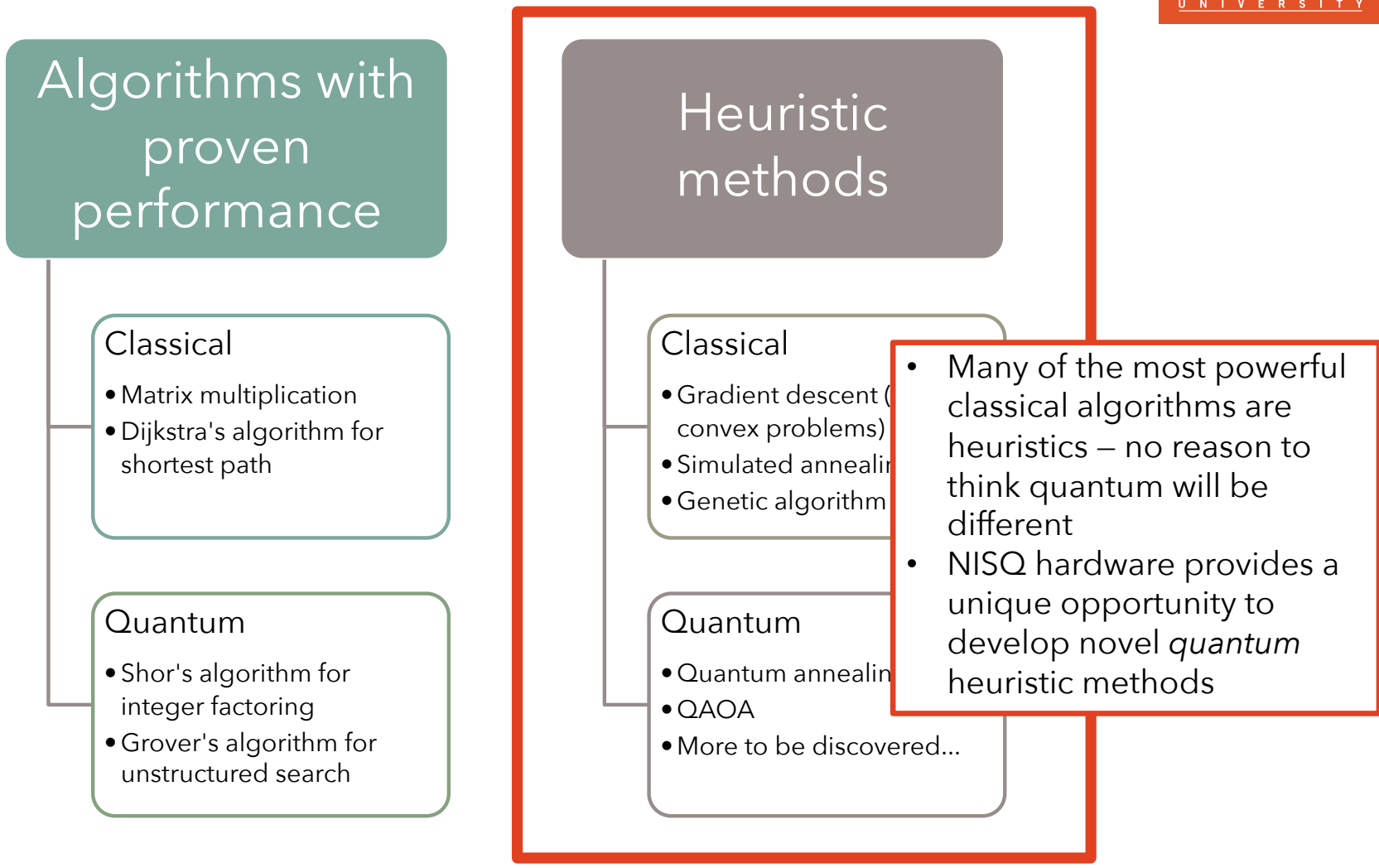
Heuristic methods

Classical

- Gradient descent (for non-convex problems)
- Simulated annealing
- Genetic algorithm

Quantum

- Quantum annealing
- QAOA
- More to be discovered...

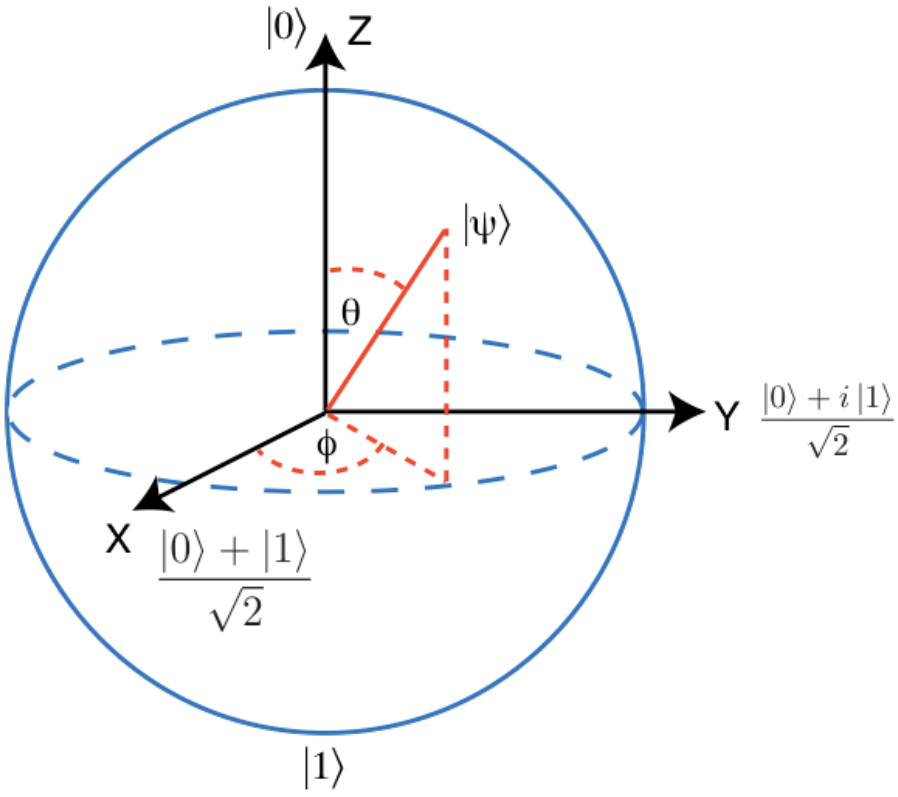

Algorithms with proven performance
Classical
• Matrix multiplication
• Dijkstra's algorithm for shortest path
Quantum
• Shor's algorithm for integer factoring
• Grover's algorithm for unstructured search
Heuristic methods
Classical
• Gradient descent (
convex problems)
• Simulated annealin
• Genetic algorithm
Quantum
• Quantum annealin
• QAOA
• More to be discovered...
• Many of the most powerful classical algorithms are heuristics – no reason to think quantum will be different
• NISQ hardware provides a unique opportunity to develop novel *quantum* heuristic methods

PART 1:
A VERY BRIEF INTRODUCTION TO (GATE MODEL) QUANTUM COMPUTING

Qubit State

- The state of a qubit is a vector in two-dimensional complex vector space span by two basis states $|1\rangle$, $|0\rangle$:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

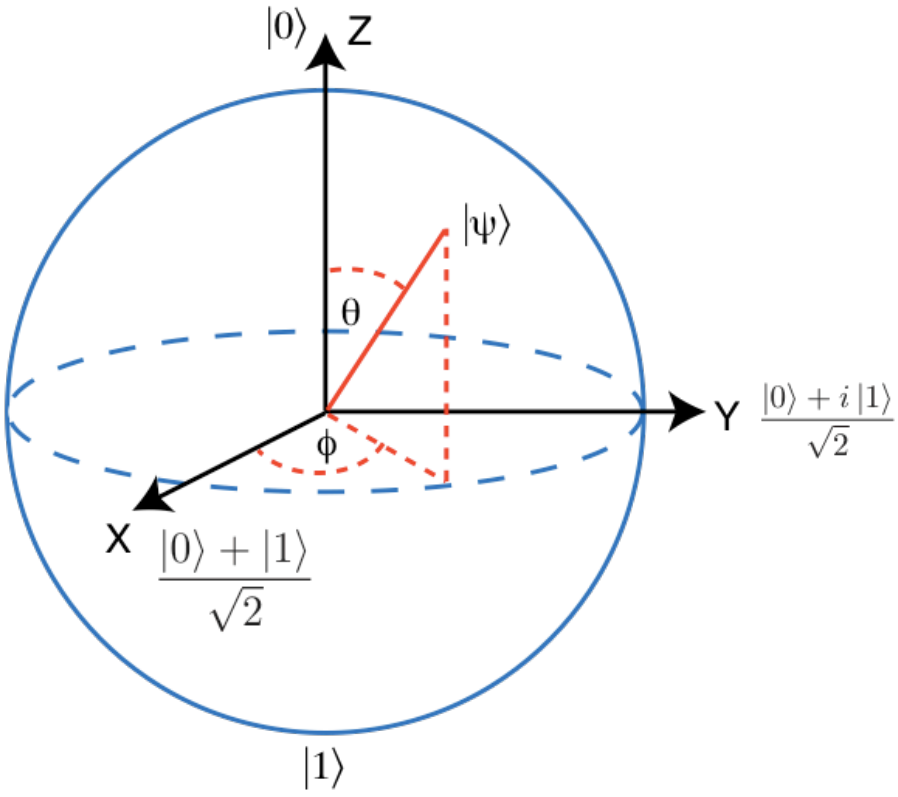


Qubit State

- The state of a qubit is a vector in two-dimensional complex vector space span by two basis states $|1\rangle, |0\rangle$:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- $|v\rangle$ is Dirac's notation for vectors. Same as $\mathbf{v}$ or $\vec{v}$.
- Note that the computational basis is something that is chosen (e.g. in designing the hardware implementation)

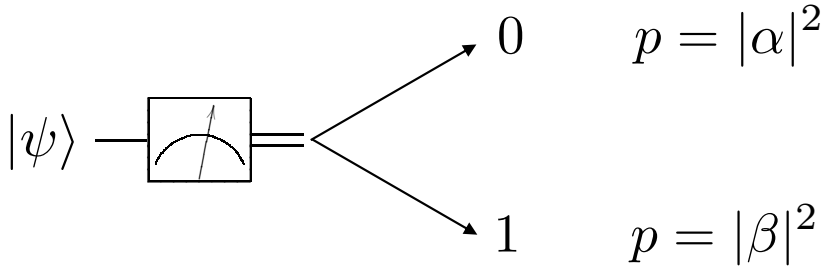


Qubit State

- The state of a qubit is a vector in two-dimensional complex vector space span by two basis states $|1\rangle, |0\rangle$:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- If we measure a qubit, we get 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$



Qubit State

- The state of a qubit is a vector in two-dimensional complex vector space span by two basis states $|1\rangle, |0\rangle$:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- If we measure a qubit, we get 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$
- Amplitudes α and β are "infinite precision", but we do not have access to them, only to the measurement result (0 or 1)

Qubit State

- The state of a qubit is a vector in two-dimensional complex vector space span by two basis states $|1\rangle, |0\rangle$:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- If we measure a qubit, we get 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$
- Amplitudes α and β are "infinite precision", but we do not have access to them, only to the measurement result (0 or 1)

- Note: if we had direct access to infinite precision amplitudes, quantum computing would not be digital! Even classical infinite precision lets us solve some NP-hard problems in polynomial time

Combining State Spaces

- Let *X* be a vector space with basis $\{|x_1\rangle, \ldots, |x_n\rangle\}$ and *Y* be a vector space with basis $\{|y_1\rangle, \ldots, |y_m\rangle\}$.

- **Classical** state spaces combine via the **Cartesian product**:
  - $X \times Y$ has basis $\{|x_1\rangle, \ldots, |x_n\rangle, |y_1\rangle, \ldots, |y_m\rangle\}$
  - $\dim(X \times Y) = \dim(X) + \dim(Y) = n + m$

- **Quantum** state spaces combine via the **tensor product**:
  - $X \times Y$ has basis $\{|x_1\rangle \otimes |y_1\rangle, |x_1\rangle \otimes |y_2\rangle, \ldots, |x_n\rangle \otimes |y_m\rangle\}$
  - $\dim(X \otimes Y) = \dim(X) \times \dim(Y) = n \times m$

2-Qubit State

- Single qubit state space *B* has basis $\{|1\rangle, |0\rangle\}$
- 2-qubit state space $B \otimes B$ has basis:

$$\{|1\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |0\rangle \otimes |0\rangle\} = \{|11\rangle, |10\rangle, |01\rangle, |00\rangle\}$$

- The state of a two-qubit system:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

n-Qubit State

- Single qubit state space *B* has basis $\{|1\rangle, |0\rangle\}$
- *n*-qubit state space $B \otimes B \otimes \ldots \otimes B$ is a 2^n-dimensional complex vector space with basis:

$$\{|00\ldots00\rangle, |00\ldots01\rangle, \ldots, |11\ldots10\rangle, |11\ldots11\rangle\}$$

- The state of an *n*-qubit system:
- Note that we need 2^n complex numbers to describe an n-qubit system

$$\begin{aligned} |\psi\rangle &= \alpha_1 |00\ldots00\rangle \\ &+ \alpha_2 |00\ldots01\rangle \\ &\ldots \\ &+ \alpha_{2^n-1} |11\ldots10\rangle \\ &+ \alpha_{2^n} |11\ldots11\rangle \end{aligned}$$

Entangled States

- Entangled states are the states that cannot be written as a tensor product of independent qubits
- Example: EPR pair $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

$$|\psi\rangle \otimes |\phi\rangle = \quad (\alpha_0 |0\rangle + \alpha_1 |1\rangle) \otimes (\beta_0 |0\rangle + \beta_1 |1\rangle)$$

Entangled States

- Entangled states are the states that cannot be written as a tensor product of independent qubits
- Example: EPR pair $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

$$
\begin{aligned}
|\psi\rangle \otimes |\phi\rangle = & \quad (\alpha_0 |0\rangle + \alpha_1 |1\rangle) \otimes (\beta_0 |0\rangle + \beta_1 |1\rangle) \\
= & \quad \alpha_0\beta_0 |00\rangle + \alpha_0\beta_1 |01\rangle + \alpha_1\beta_0 |10\rangle + \alpha_1\beta_1 |11\rangle
\end{aligned}
$$

Entangled States

- Entangled states are the states that cannot be written as a tensor product of independent qubits
- Example: EPR pair $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

$$
\begin{aligned}
|\psi\rangle \otimes |\phi\rangle = & \quad (\alpha_0 |0\rangle + \alpha_1 |1\rangle) \otimes (\beta_0 |0\rangle + \beta_1 |1\rangle) \\
= & \quad \alpha_0\beta_0 |00\rangle + \alpha_0\beta_1 |01\rangle + \alpha_1\beta_0 |10\rangle + \alpha_1\beta_1 |11\rangle \\
\neq & \quad \alpha_0\beta_0 |00\rangle + 0 |01\rangle + 0 |10\rangle + \alpha_1\beta_1 |11\rangle \\
= & \quad \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle)
\end{aligned}
$$

Entangled States

- Entangled states are the states that cannot be written as a tensor product of independent qubits
- Example: EPR pair $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

$$
\begin{aligned}
|\psi\rangle \otimes |\phi\rangle = & \quad (\alpha_0 |0\rangle + \alpha_1 |1\rangle) \otimes (\beta_0 |0\rangle + \beta_1 |1\rangle) \\
= & \quad \alpha_0\beta_0 |00\rangle + \alpha_{\mathbf{0}}\beta_{\mathbf{1}} |01\rangle + \alpha_{\mathbf{1}}\beta_{\mathbf{0}} |10\rangle + \alpha_1\beta_1 |11\rangle \\
\neq & \quad \alpha_0\beta_0 |00\rangle + \mathbf{0} |01\rangle + \mathbf{0} |10\rangle + \alpha_1\beta_1 |11\rangle \\
= & \quad \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle)
\end{aligned}
$$

Entangled States

- Entangled states are the states that cannot be written as a tensor product of independent qubits
- Example: EPR pair $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

$$
\begin{aligned}
|\psi\rangle \otimes |\phi\rangle = & \quad (\alpha_0 |0\rangle + \alpha_1 |1\rangle) \otimes (\beta_0 |0\rangle + \beta_1 |1\rangle) \\
= & \quad \alpha_0\beta_0 |00\rangle + \alpha_{\mathbf{0}}\beta_{\mathbf{1}} |01\rangle + \alpha_{\mathbf{1}}\beta_{\mathbf{0}} |10\rangle + \alpha_1\beta_1 |11\rangle \\
\neq & \quad \alpha_0\beta_0 |00\rangle + \mathbf{0} |01\rangle + \mathbf{0} |10\rangle + \alpha_1\beta_1 |11\rangle \\
= & \quad \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle)
\end{aligned}
$$

- Non-entangled states are easy to simulate (effectively classical)!

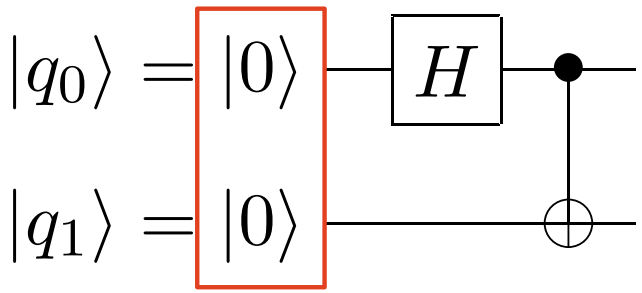
Gates and Circuits

- Computation is performed by applying *gates* (unitary matrices):

$$H \equiv \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; \quad X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; \quad Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix};$$
$$R_z(\theta) \equiv e^{-i\theta\hat{\sigma}^z/2} = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

- Looks like building these gates requires infinite precision!
- However, one can show that constant errors are tolerated, so we can *digitize* these operations

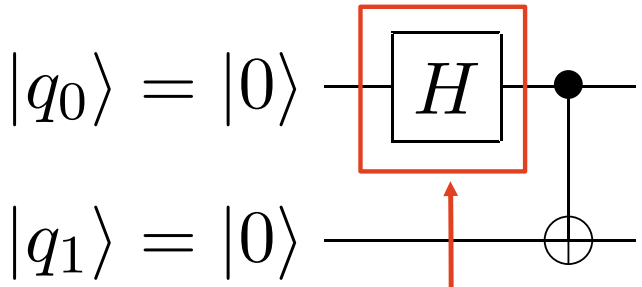
Preparing the EPR pair



$$|0\rangle \otimes |0\rangle = |00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Preparing the EPR pair

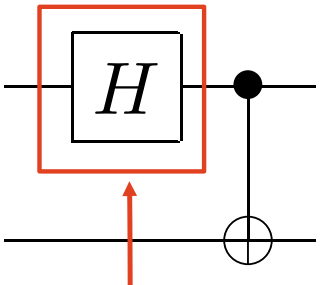
$$|0\rangle \otimes |0\rangle = |00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



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

Preparing the EPR pair

$$|0\rangle \otimes |0\rangle = |00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

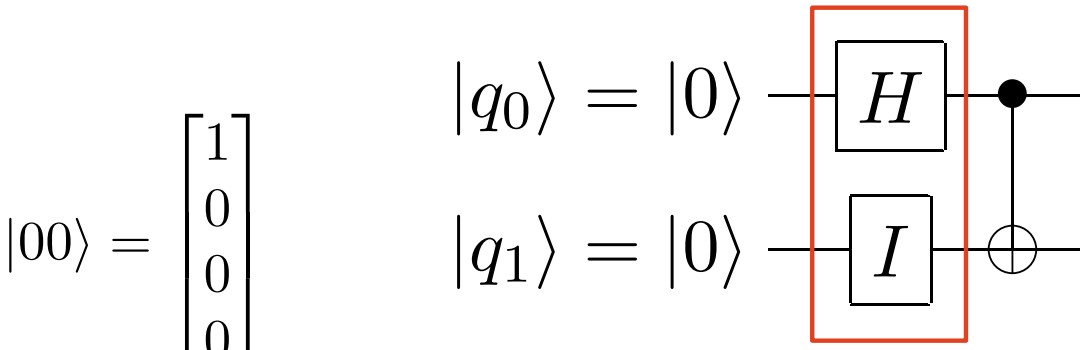


$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

How can we apply a one-qubit gate (2x2 matrix) to a two-qubit state (size 4 vector)?

Preparing the EPR pair

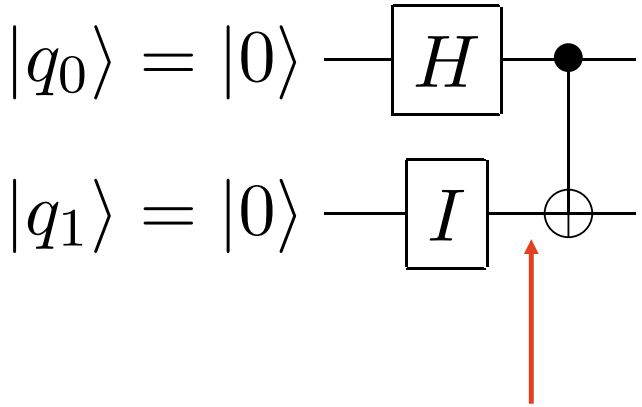
$$|0\rangle \otimes |0\rangle = |00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$$H \otimes I = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

Preparing the EPR pair

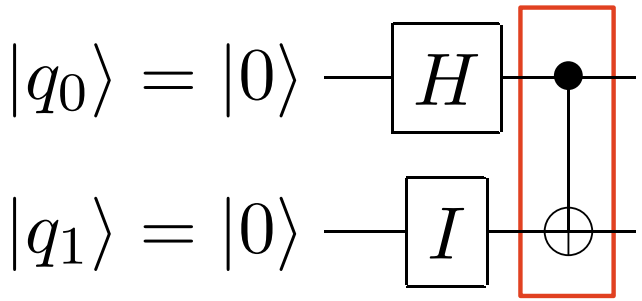
$$|0\rangle \otimes |0\rangle = |00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$$(H \otimes I)\,|00\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

Preparing the EPR pair

$$|0\rangle \otimes |0\rangle = |00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

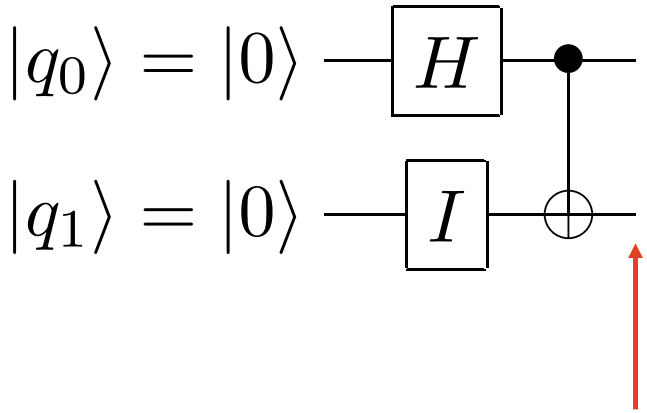


$$(H \otimes I)\,|00\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

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

Preparing the EPR pair

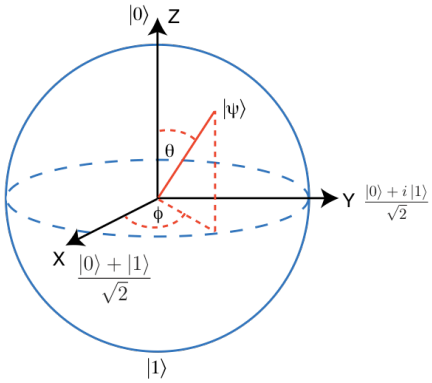
$$|0\rangle \otimes |0\rangle = |00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$$CNOT(H \otimes I)\,|00\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Recap

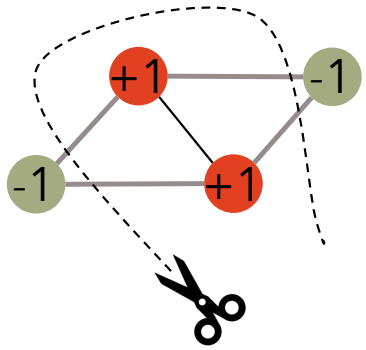
- The state of a qubit is a vector in two-dimensional complex vector space
- Quantum states combine via tensor product:
  $\dim(X \otimes Y) = \dim(X) \times \dim(Y) = n \times m$
- The state of n-qubit system is a vector in 2^n-dimensional space
- Gates are unitary matrices



PART 2:
QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM (QAOA)

Maximum Cut Problem (MAXCUT)

- The goal of maximum cut is to split the set of vertices V of a graph into two disjoint parts such that the number of edges spanning two parts is maximized.
- For example, if color denotes part, in the graph on the right 4 edges are cut:

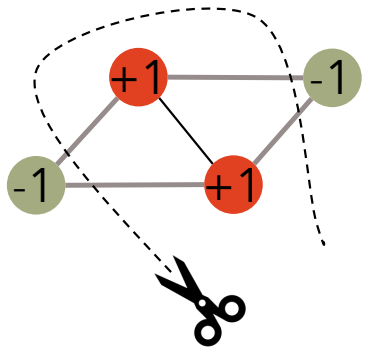


- Maximum cut can be formulated as an optimization problem:

$$\max_{\mathbf{s}} \frac{1}{2} \sum_{ij \in E} (1 - s_i s_j) \qquad s_i \in \{-1, +1\}$$

Maximum Cut Problem (MAXCUT)

- The goal of maximum cut is to split the set of vertices V of a graph into two disjoint parts such that such that the number of edges spanning two parts is maximized.
- For example, if color denotes part, in the graph on the right 4 edges are cut:



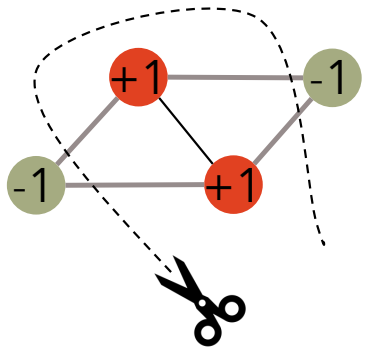
- Maximum cut can be formulated as an optimization problem:

$$\max_{\mathbf{s}} \frac{1}{2} \sum_{ij \in E} (1 - s_i s_j) \qquad s_i \in \{-1, +1\}$$

Same sign – no edge is cut (no contribution to the objective): $\frac{1}{2}(1 - s_i s_j) = 0$

Maximum Cut Problem (MAXCUT)

- The goal of maximum cut is to split the set of vertices V of a graph into two disjoint parts such that such that the number of edges spanning two parts is maximized.
- For example, if color denotes part, in the graph on the right 4 edges are cut:



- Maximum cut can be formulated as an optimization problem:

$$\max_{\mathbf{s}} \frac{1}{2} \sum_{ij \in E} (1 - s_i s_j) \qquad s_i \in \{-1, +1\}$$

Different sign – an edge is cut (contribution to the objective = 1): $\frac{1}{2}(1 - s_i s_j) = 1$

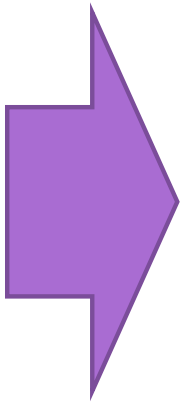
Solving Combinatorial Optimization Problems on a Quantum Computer

CLEMSON UNIVERSITY

Classical

Maximizing objective

$$\max_{\mathbf{s}} C(\mathbf{s}) = \frac{1}{2} \sum_{ij \in E} (1 - s_i s_j)$$



Quantum

Characterizing a Hamiltonian (Hermitian operator) *C*

To solve an optimization problem on a quantum computer, we need to convert it into a problem of characterizing a quantum Hamiltonian

Solving Combinatorial Optimization Problems on a Quantum Computer

CLEMSON UNIVERSITY

	Classical	Quantum
Objective	$\max_{\mathbf{s}} C(\mathbf{s}) = \frac{1}{2}\sum_{ij\in E}(1 - s_i s_j)$	Hamiltonian (Hermitian operator) C
Evaluation of objective	$C(\mathbf{s})$	Expectation of C (energy) in state s $\langle s \vert C \vert s \rangle$
Solution	$\mathbf{s} \in \{-1, +1\}^n$	Highest energy eigenstate $\vert s \rangle$

Solving Combinatorial Optimization Problems on a Quantum Computer

CLEMSON UNIVERSITY

	Classical	Quantum
Objective	$\max_{\mathbf{s}} C(\mathbf{s}) = \frac{1}{2}\sum_{ij\in E}(1 - s_i s_j)$	Hamiltonian (Hermitian operator) C
Evaluation of objective	$C(\mathbf{s})$	Expectation of C (energy) in state s $\langle s \vert C \vert s \rangle$
Solution	$\mathbf{s} \in \{-1, +1\}^n$	Highest energy eigenstate $\vert s \rangle$

If this eigenstate is a computational basis state, we can measure it and **get the solution with certainty**

Solving Combinatorial Optimization Problems on a Quantum Computer

CLEMSON UNIVERSITY

	Classical	Quantum
Objective	$\max_{\mathbf{s}} C(\mathbf{s}) = \frac{1}{2}\sum_{ij\in E}(1 - s_i s_j)$	Hamiltonian (Hermitian operator) C
Evaluation		state s
Solution	$\mathbf{s} \in \{-1, +1\}^n$	Highest energy eigenstate $\lvert s\rangle$

How do we construct this Hamiltonian?

Notation Reminder

$$|x\rangle = \mathbf{x} = \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{column vector} \qquad \langle y| = |y\rangle^\dagger = \begin{bmatrix} y_1^* & y_2^* & \cdots & y_n^* \end{bmatrix} \quad \text{conjugate transpose}$$

Notation Reminder

$$|x\rangle = \mathbf{x} = \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{column vector} \qquad \langle y| = |y\rangle^\dagger = \begin{bmatrix} y_1^* & y_2^* & \cdots & y_n^* \end{bmatrix} \quad \text{conjugate transpose}$$

$$\langle y|x\rangle = y_1^* x_1 + \cdots + y_n^* x_n = \begin{bmatrix} y_1^* & y_2^* & \cdots & y_n^* \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{inner product}$$

Notation Reminder

$$|x\rangle = \mathbf{x} = \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{column vector} \qquad \langle y| = |y\rangle^\dagger = \begin{bmatrix} y_1^* & y_2^* & \cdots & y_n^* \end{bmatrix} \quad \text{conjugate transpose}$$

$$\langle y|x\rangle = y_1^* x_1 + \cdots + y_n^* x_n = \begin{bmatrix} y_1^* & y_2^* & \cdots & y_n^* \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{inner product}$$

$$|x\rangle\langle y| = \begin{bmatrix} x_1 y_1^* & \cdots & x_1 y_n^* \\ \vdots & \ddots & \vdots \\ x_n y_1^* & \cdots & x_n y_n^* \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} y_1^* & y_2^* & \cdots & y_n^* \end{bmatrix} \quad \text{outer product}$$

Constructing MAXCUT Hamiltonian

- MAXCUT objective:

$$\max_{\mathbf{s}} \frac{1}{2} \sum_{ij \in E} (1 - s_i s_j) \qquad s_i \in \{-1, +1\}$$

Constructing MAXCUT Hamiltonian

- MAXCUT objective:

$$\max_{\mathbf{s}} \frac{1}{2} \sum_{ij \in E} (1 - s_i s_j) \qquad s_i \in \{-1, +1\}$$

- MAXCUT Hamiltonian is constructed by **mapping binary variables s_i onto the eigenvalues of Z**

$$C = \frac{1}{2} \sum_{ij \in E} (I - Z_i Z_j)$$

- Want to show:

$$C(\mathbf{x}) = \langle x | \, C \, | x \rangle$$

Pauli Z operator

- Consider Pauli Z operator

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

Pauli Z operator

- Consider Pauli Z operator

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

- Note that it has eigenvalues -1, +1 with eigenvectors being computational basis states

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

$$Z|1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} = (-1)|1\rangle$$

Pauli Z operator

- Note that it has eigenvalues -1, +1 with eigenvectors being computational basis states

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

$$Z|1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} = (-1)|1\rangle$$

- Therefore,

$$\langle 0|Z|0\rangle = \langle 0|0\rangle = 1$$

$$\langle 1|Z|1\rangle = (-1)\langle 1|1\rangle = -1$$

Pauli Z operator

- Note that it has eigenvalues -1, +1 with eigenvectors being computational basis states

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

$$Z|1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} = (-1)|1\rangle$$

- Therefore,

$$\langle x| Z |x\rangle = (-1)^x \quad x \in \{0, 1\}$$

Pauli Z operator

- Acting on the i-th qubit:

$$\begin{aligned}
\langle x_0 \dots x_n | \, Z_i \, | x_0 \dots x_n \rangle &= \langle x_0 \dots x_n | \, I \otimes \dots \otimes Z_i \otimes \dots \otimes I \, | x_0 \dots x_n \rangle \\
&= (-1)^{x_i} \langle x_0 \dots x_n | x_0 \dots x_n \rangle \\
&= (-1)^{x_i} \\
x_i \in \{0, 1\}, \quad & i = 1, \dots n
\end{aligned}$$

Pauli Z operator

- Acting on the i-th and j-th qubit:

$$\begin{aligned}\langle x_0 \dots x_n | \, Z_i Z_j \, | x_0 \dots x_n \rangle &= \langle x_0 \dots x_n | \, I \otimes \dots \otimes Z_i \otimes Z_j \otimes \dots \otimes I \, | x_0 \dots x_n \rangle \\ &= (-1)^{x_i} (-1)^{x_j} \, \langle x_0 \dots x_n | x_0 \dots x_n \rangle \\ &= (-1)^{x_i} (-1)^{x_j} \\ x_i \in \{0, 1\}, \quad i &= 1, \dots n\end{aligned}$$

- (note that here we reorder qubits such that i-th and j-th qubit are adjacent)

Constructing MAXCUT Hamiltonian

- MAXCUT objective:

$$\max_{\mathbf{s}} \frac{1}{2} \sum_{ij \in E} (1 - s_i s_j) \qquad s_i \in \{-1, +1\}$$

- MAXCUT Hamiltonian is constructed by **mapping binary variables s_i onto the eigenvalues of Z**

$$C = \frac{1}{2} \sum_{ij \in E} (I - Z_i Z_j)$$

Verifying MAXCUT Hamiltonian

- MAXCUT objective:

$$\max_{\mathbf{s}} \frac{1}{2} \sum_{ij \in E} (1 - s_i s_j) \qquad s_i \in \{-1, +1\}$$

- Let's reformulate it in the following way:

$$x_i = \frac{1}{2}(1 - s_i)$$

$$s_i = 1 \rightarrow x_i = 0, \qquad (-1)^{x_i} = 1 = s_i$$

$$s_i = -1 \rightarrow x_i = 1, \qquad (-1)^{x_i} = -1 = s_i$$

Verifying MAXCUT Hamiltonian

- MAXCUT objective:

$$\max_{\mathbf{s}} \frac{1}{2} \sum_{ij \in E} (1 - s_i s_j) \qquad s_i \in \{-1, +1\}$$

- Let's reformulate it in the following way:

$$x_i = \frac{1}{2}(1 - s_i)$$

$$s_i = 1 \rightarrow x_i = 0, \qquad (-1)^{x_i} = 1 = s_i$$

$$s_i = -1 \rightarrow x_i = 1, \qquad (-1)^{x_i} = -1 = s_i$$

- New objective:

$$\max_{\mathbf{x}} \frac{1}{2} \sum_{ij \in E} (1 - (-1)^{x_i}(-1)^{x_j}) \qquad x_i \in \{0, 1\}$$

Verifying MAXCUT Hamiltonian

- MAXCUT objective:

$$C(\mathbf{x}) = \frac{1}{2} \sum_{ij \in E} (1 - (-1)^{x_i} (-1)^{x_j}) \qquad x_i \in \{0, 1\}$$

- MAXCUT Hamiltonian:

$$C = \frac{1}{2} \sum_{ij \in E} (I - Z_i Z_j)$$

- Want to show:

$$C(\mathbf{x}) = \langle x | \, C \, | x \rangle$$

Verifying MAXCUT Hamiltonian

$$\langle x| C |x\rangle = \langle x_0 \dots x_n| \frac{1}{2} \sum_{ij \in E} (I - Z_i Z_j) |x_0 \dots x_n\rangle$$

$$= \frac{1}{2} \sum_{ij \in E} \langle x_0 \dots x_n| (I - Z_i Z_j) |x_0 \dots x_n\rangle$$

$$= \frac{1}{2} \sum_{ij \in E} (1 - \langle x_0 \dots x_n| Z_i Z_j |x_0 \dots x_n\rangle)$$

$$= \frac{1}{2} \sum_{ij \in E} (1 - (-1)^{x_i} (-1)^{x_j})) = C(\mathbf{x})$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots n$$

Constructing MAXCUT Hamiltonian

- MAXCUT objective:

$$\max_{\mathbf{s}} \frac{1}{2} \sum_{ij \in E} (1 - s_i s_j) \qquad s_i \in \{-1, +1\}$$

- MAXCUT Hamiltonian is constructed by **mapping binary variables s_i onto the eigenvalues of Z**

$$C = \frac{1}{2} \sum_{ij \in E} (I - Z_i Z_j)$$

- **Note that the same procedure would work for any (unconstrained) binary objective!**

Constructing MAXCUT Hamiltonian

- MAXCUT objective:

$$\max_{\mathbf{s}} \frac{1}{2} \sum_{ij \in E} (1 - s_i s_j) \qquad s_i \in \{-1, +1\}$$

- MAXCUT Hamiltonian is constructed by **mapping binary variables s_i onto the eigenvalues of Z**

$$C = \frac{1}{2} \sum_{ij \in E} (I - Z_i Z_j)$$

- **Now all we need to do is find a quantum state $|x\rangle$ that maximizes $\langle x| C |x\rangle$!**

Quantum Approximate Optimization Algorithm (QAOA)

- QAOA prepares a parameterized "trial" (ansatz) state of the form:

$$
\begin{aligned}
|\psi(\boldsymbol{\theta})\rangle &= |\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle \\
&= e^{-i\beta_p B} e^{-i\gamma_p C} \cdots e^{-i\beta_1 B} e^{-i\gamma_1 C} H^{\otimes n} |0\rangle .
\end{aligned}
$$

- Here *C* is the problem Hamiltonian, e.g. for MAXCUT: $C = \frac{1}{2} \sum_{ij \in E} (I - Z_i Z_j)$
- *B* is the mixer Hamiltonian: $B = \sum_i X_i$

Quantum Approximate Optimization Algorithm (QAOA)

- QAOA prepares a parameterized "trial" (ansatz) state of the form:

$$
\begin{aligned}
|\psi(\boldsymbol{\theta})\rangle &= |\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle \\
&= e^{-i\beta_p B} e^{-i\gamma_p C} \cdots e^{-i\beta_1 B} e^{-i\gamma_1 C} H^{\otimes n} |0\rangle .
\end{aligned}
$$

- Then a classical optimizer is used to vary the parameters $\boldsymbol{\beta}, \boldsymbol{\gamma}$ to maximize:

$$
f(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \langle \psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) | \, C \, | \psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle .
$$

Quantum Approximate Optimization Algorithm (QAOA)

- QAOA prepares a parameterized "trial" (ansatz) state of the form:

$$
\begin{aligned}
|\psi(\boldsymbol{\theta})\rangle &= |\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle \\
&= e^{-i\beta_p B} e^{-i\gamma_p C} \cdots e^{-i\beta_1 B} e^{-i\gamma_1 C} H^{\otimes n} |0\rangle \,.
\end{aligned}
$$

- Note that for p $\rightarrow \infty$ QAOA can *at least* exactly approximate adiabatic quantum evolution and can therefore find the exact optimal solution
- For small p, picture is more mixed, but there is some indication of the potential for quantum advantage

Quantum Approximate Optimization Algorithm (QAOA)

- QAOA prepares a parameterized "trial" (ansatz) state of the form:

$$
\begin{aligned}
|\psi(\boldsymbol{\theta})\rangle &= |\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle \\
&= e^{-i\beta_p B} e^{-i\gamma_p C} \cdots e^{-i\beta_1 B} e^{-i\gamma_1 C} H^{\otimes n} |0\rangle \, .
\end{aligned}
$$

How do we implement this circuit in gates?

Implementing QAOA

- Let's assume the following gate set

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; \quad Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}; \quad Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}; \quad H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix};$$

$$S \equiv \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}; \quad R_z(\theta) \equiv e^{-i\theta Z/2} = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

Implementing QAOA

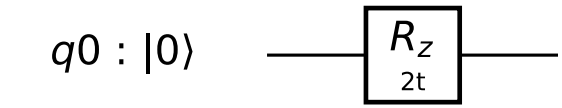
- Let's start with simple operator $e^{-iZt} = R_z(2t)$

$$Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad R_z(\theta) \equiv e^{-i\theta Z/2} = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

Implementing QAOA

- Let's start with simple operator $e^{-iZt} = R_z(2t)$

- Circuit:



$$Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad R_z(\theta) \equiv e^{-i\theta Z/2} = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

Implementing QAOA

- Slightly more complicated operator: $e^{-iZZt} = e^{-iZ \otimes Zt}$
- Remember that Z has eigenvectors $|0\rangle$, $|1\rangle$ with eigenvalues 1,-1 and $e^A |v\rangle = e^\lambda |v\rangle$ if $A|v\rangle = \lambda |v\rangle$
- Then:

$$
\begin{aligned}
e^{-iZ\otimes Zt}|00\rangle &= e^{-i(1\times 1)t}|00\rangle = e^{-it}|00\rangle \\
e^{-iZ\otimes Zt}|01\rangle &= e^{-i(1\times -1)t}|01\rangle = e^{it}|01\rangle \\
e^{-iZ\otimes Zt}|10\rangle &= e^{-i(-1\times 1)t}|10\rangle = e^{it}|10\rangle \\
e^{-iZ\otimes Zt}|11\rangle &= e^{-i(-1\times -1)t}|11\rangle = e^{-it}|11\rangle
\end{aligned}
$$

- Adds a phase factor with the sign depending on parity! In general:

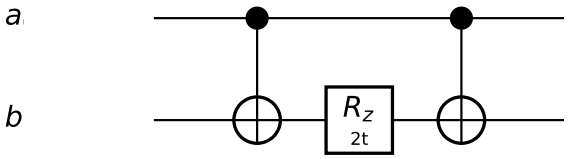
$$e^{-iZ\otimes Zt}|ab\rangle = e^{-i(-1)^{a\oplus b}t}|ab\rangle$$

Implementing QAOA

- Slightly more complicated operator: $e^{-iZZt} = e^{-iZ\otimes Zt}$
- Adds a phase factor with the sign depending on parity! In general:

$$e^{-iZ\otimes Zt}\,|ab\rangle = e^{-i(-1)^{a\oplus b}t}\,|ab\rangle$$

- Circuit:

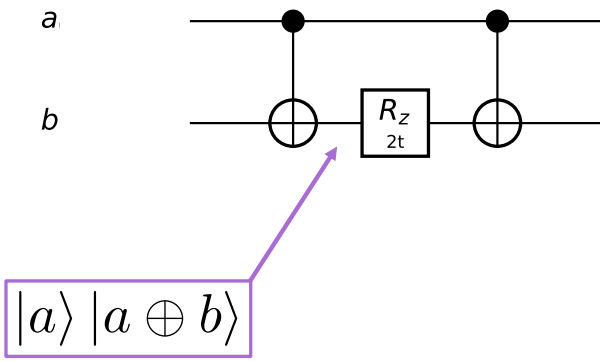


Implementing QAOA

- Slightly more complicated operator: $e^{-iZZt} = e^{-iZ\otimes Zt}$
- Adds a phase factor with the sign depending on parity! In general:

$$e^{-iZ\otimes Zt}\,|ab\rangle = e^{-i(-1)^{a\oplus b}t}\,|ab\rangle$$

- Circuit:

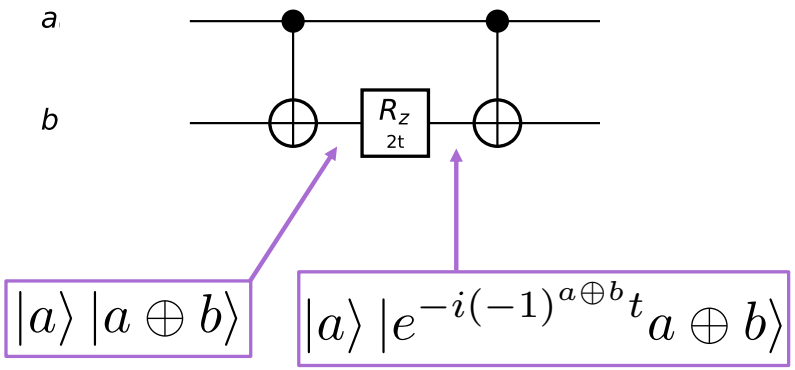


Implementing QAOA

- Slightly more complicated operator: $e^{-iZZt} = e^{-iZ\otimes Zt}$
- Adds a phase factor with the sign depending on parity! In general:

$$e^{-iZ\otimes Zt}\,|ab\rangle = e^{-i(-1)^{a\oplus b}t}\,|ab\rangle$$

- Circuit:

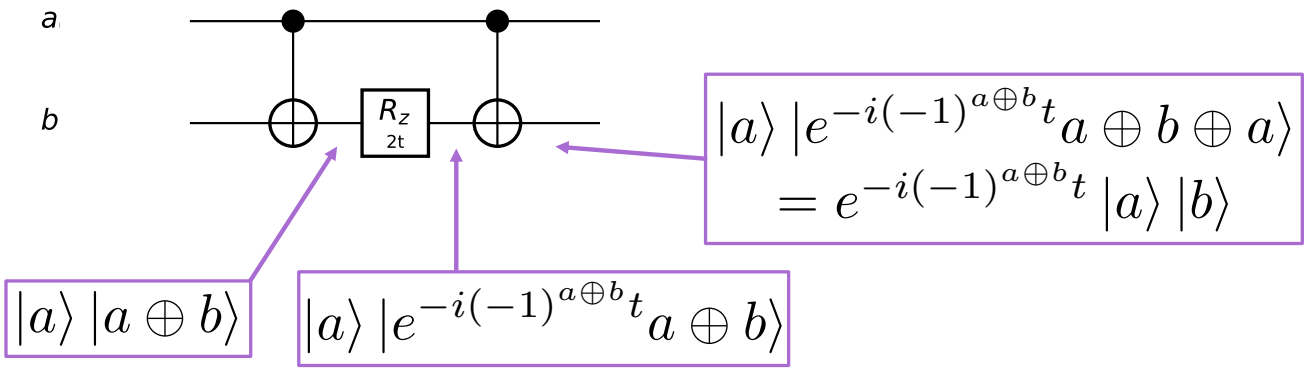


Implementing QAOA

- Slightly more complicated operator: $e^{-iZZt} = e^{-iZ\otimes Zt}$
- Adds a phase factor with the sign depending on parity! In general:

$$e^{-iZ\otimes Zt}\,|ab\rangle = e^{-i(-1)^{a\oplus b}t}\,|ab\rangle$$

- Circuit:

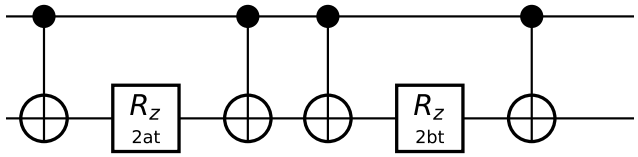


Implementing QAOA

- If terms of our Hamiltonian commute, we can just concatenate corresponding circuits:

$$e^{-iaZ_iZ_jt-ibZ_jZ_kt} = e^{-iaZ_iZ_jt}e^{-ibZ_jZ_kt}$$

- Circuit:

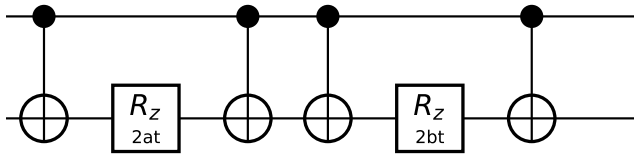


Implementing QAOA

- If terms of our Hamiltonian commute, we can just concatenate corresponding circuits:

$$e^{-iaZ_iZ_jt-ibZ_jZ_kt} = e^{-iaZ_iZ_jt}e^{-ibZ_jZ_kt}$$

- Circuit:



- Now we can simulate the operator corresponding to the problem Hamiltonian:

$$e^{-i\gamma C} = e^{-i\gamma\frac{1}{2}\sum_{ij\in E}(I-Z_iZ_j)}$$

Implementing QAOA

- QAOA prepares a parameterized "trial" (ansatz) state of the form:

$$
\begin{aligned}
|\psi(\boldsymbol{\theta})\rangle &= |\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle \\
&= e^{-i\beta_p B} e^{-i\gamma_p C} \cdots e^{-i\beta_1 B} e^{-i\gamma_1 C} H^{\otimes n} |0\rangle .
\end{aligned}
$$

- Now we can simulate the operator corresponding to the problem Hamiltonian:

$$
e^{-i\gamma C} = e^{-i\gamma \frac{1}{2} \sum_{ij \in E} (I - Z_i Z_j)}
$$

Implementing QAOA

- Mixer operator: $e^{-i\beta B} = e^{-i\beta \sum_j X_j}$
- One term: e^{-iXt}

- Note that

$$e^{-iZt} = \sum_{j=0}^{\infty} \frac{(-iZt)^j}{j!} = I - iZt - \frac{Z^2t^2}{2!} + \dots$$

$$\begin{aligned} He^{-iZt}H &= H\left(I - iZt - \tfrac{Z^2t^2}{2!} + \dots\right)H = I - iHZHt - \tfrac{HZHHZHt^2}{2!} + \dots \\ &= I - iXt - \tfrac{X^2t^2}{2!} + \dots = e^{-iXt} \end{aligned}$$

Implementing QAOA

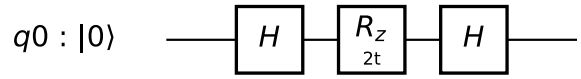
- Mixer operator: $e^{-i\beta B} = e^{-i\beta \sum_j X_j}$
- One term: e^{-iXt}

- Note that

$$e^{-iZt} = \sum_{j=0}^{\infty} \frac{(-iZt)^j}{j!} = I - iZt - \frac{Z^2t^2}{2!} + \dots$$

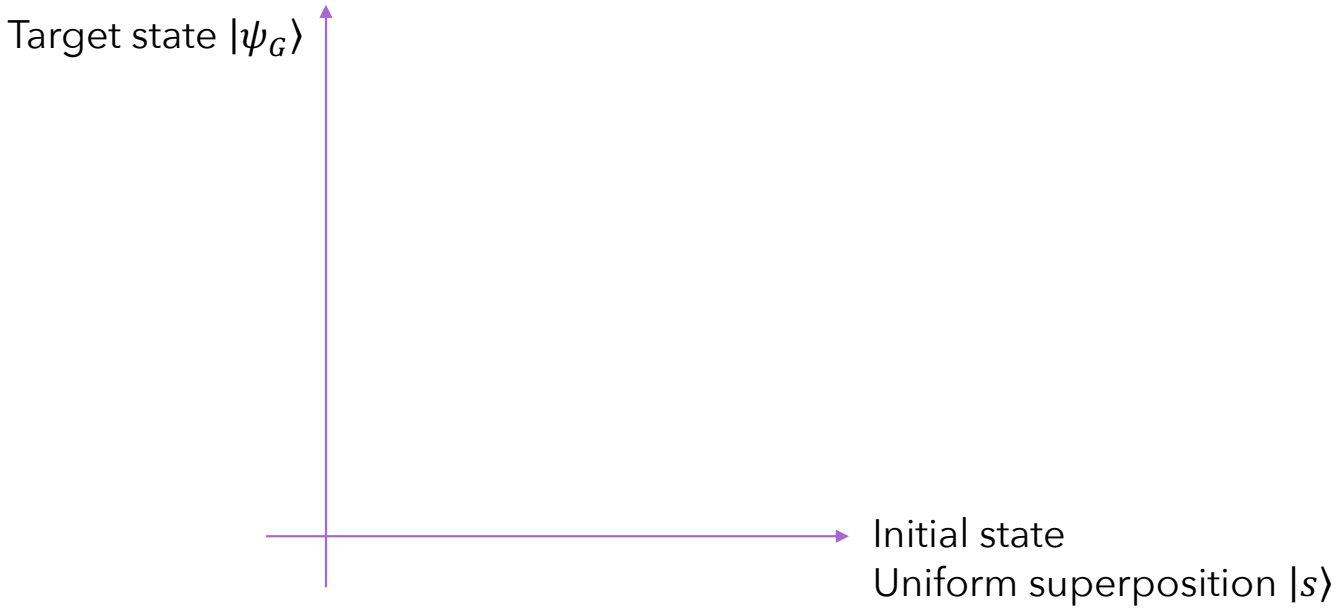
$$\begin{aligned} He^{-iZt}H &= H\left(I - iZt - \frac{Z^2t^2}{2!} + \dots\right)H = I - iHZHt - \frac{HZHHZHt^2}{2!} + \dots \\ &= I - iXt - \frac{X^2t^2}{2!} + \dots = e^{-iXt} \end{aligned}$$

- Circuit:



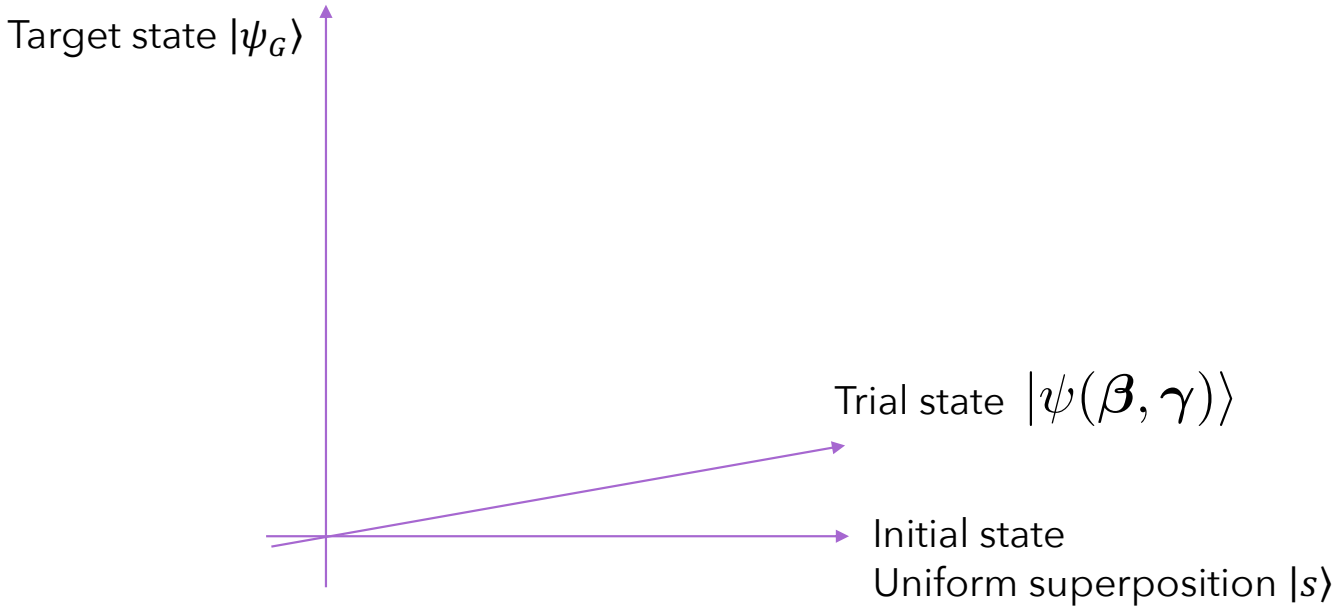
Quantum Approximate Optimization Algorithm (QAOA): Geometric Interpretation

- QAOA prepares a parameterized "trial" (ansatz) state $|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle$



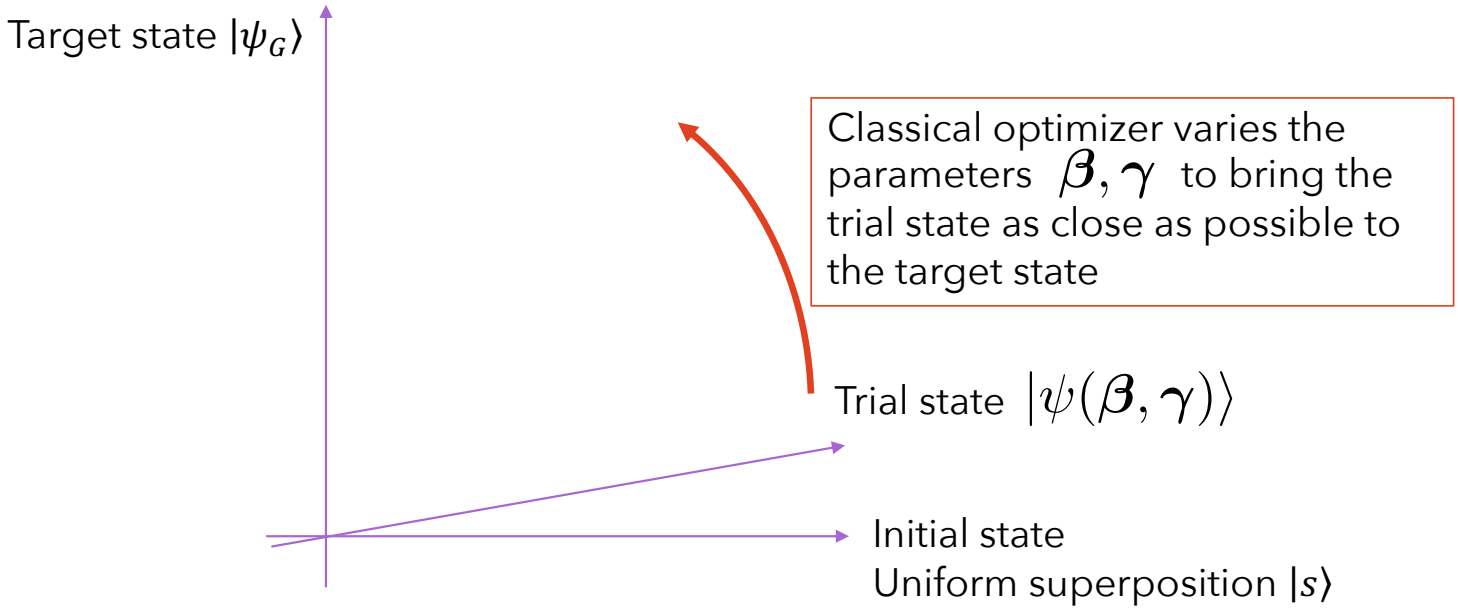
Quantum Approximate Optimization Algorithm (QAOA): Geometric Interpretation

- QAOA prepares a parameterized "trial" (ansatz) state $|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle$



Quantum Approximate Optimization Algorithm (QAOA): Geometric Interpretation

- QAOA prepares a parameterized "trial" (ansatz) state $|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle$



Quantum Approximate Optimization Algorithm (QAOA)

- QAOA prepares a parameterized "trial" (ansatz) state of the form:

$$
\begin{aligned}
|\psi(\boldsymbol{\theta})\rangle &= |\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle \\
&= e^{-i\beta_p \hat{H}_M} e^{-i\gamma_p \hat{H}_C} \cdots e^{-i\beta_1 \hat{H}_M} e^{-i\gamma_1 \hat{H}_C} |+\rangle^{\otimes n} .
\end{aligned}
$$

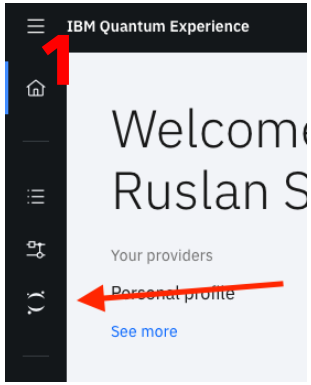
- Crucially, **the quality of QAOA solution heavily depends on the quality of the parameters found by the classical optimizer**

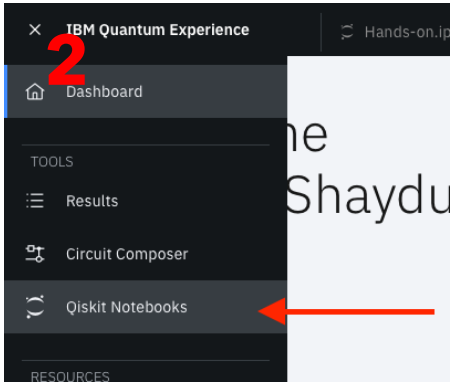
Thanks

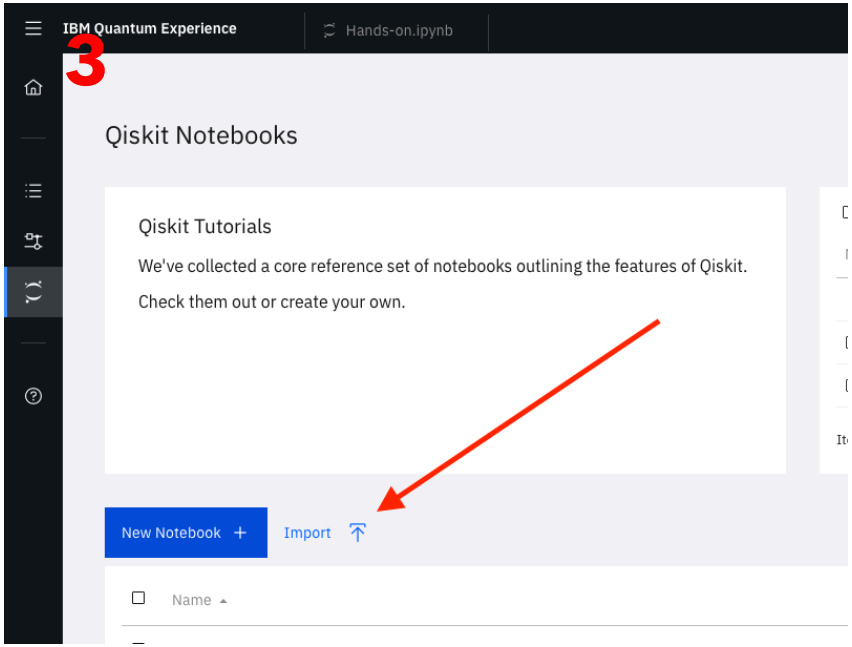
- “Introduction to Quantum Computing” tutorial at SC19 by Eleanor Rieffel and Scott Pakin.
- Paul Bernays lectures by Scott Aaronson
- 2019 Illinois Quantum Computing School lectures by Nathan Wiebe

PART 3:
HANDS-ON

- **Get the latest version of the notebook** from github.com/rsln-s/SIAM_PP_20_minitutorial or bit.ly/PP20quantum
- Go to quantum-computing.ibm.com and login (you might need to create an IBM ID if you haven't already!)
- Go to quantum-computing.ibm.com/jupyter
  If you don't like typing, there's a button on home page that takes you there:







- Import the notebook Hands-on.ipynb:
- Note that you can also run the notebook locally on your machine, but you'll have to set up your own environment