PURDUE UNIVERSITY

Quantum computing for operations research: current state and perspectives

David E. Bernal Neira

July 14, 2026

Principal Investigator of SECQUOIA

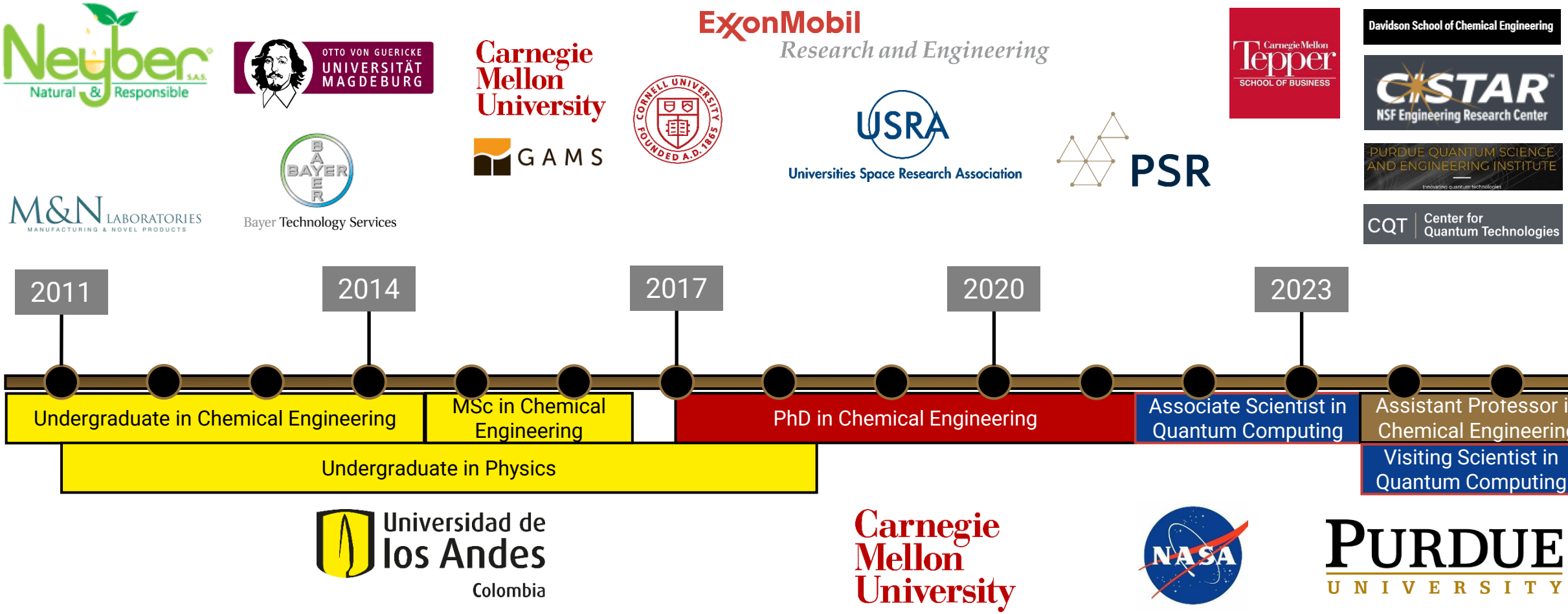
Assistant Professor - Davidson School of Chemical Engineering, Purdue University

International Federation of Operations Research Societies (IFORS) Meeting 2026

Vienna, Austria

About me

About me



SECQUOIA

Systems Engineering via Classical and Quantum Optimization for Industrial Applications

PI David E. Bernal

Postdoc Hanjing Xu
Quantum ML

Postdoc Bruna Araujo
Quantum Info Science

PhD Albert Lee
Process Superstructure

PhD Yirang Park
Quantum for Pharma

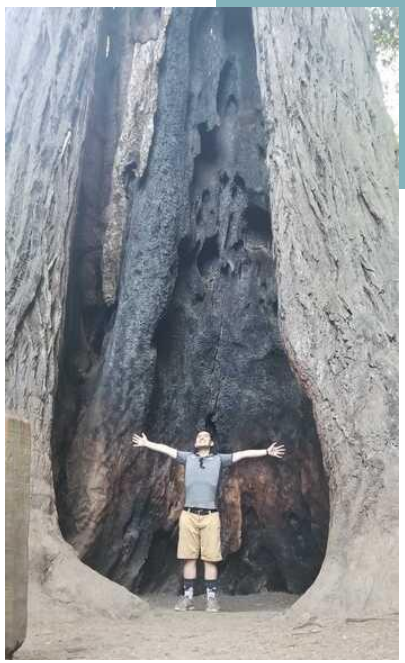
PhD Anurag Ramesh
Benchmarking Quantum

PhD Sergey Gusev
Quantum Conic opt

PhD Andres Cabeza
Process Synthesis

PhD Anja Hribljan
Quantum PINNs

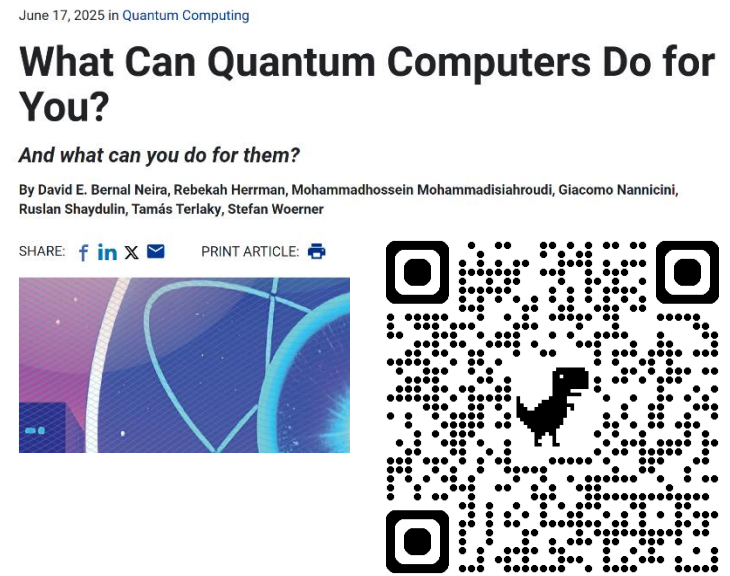
PhD Mazhar Laljee
Enhanced Oil Recovery



Quantum Computing and Operations Research

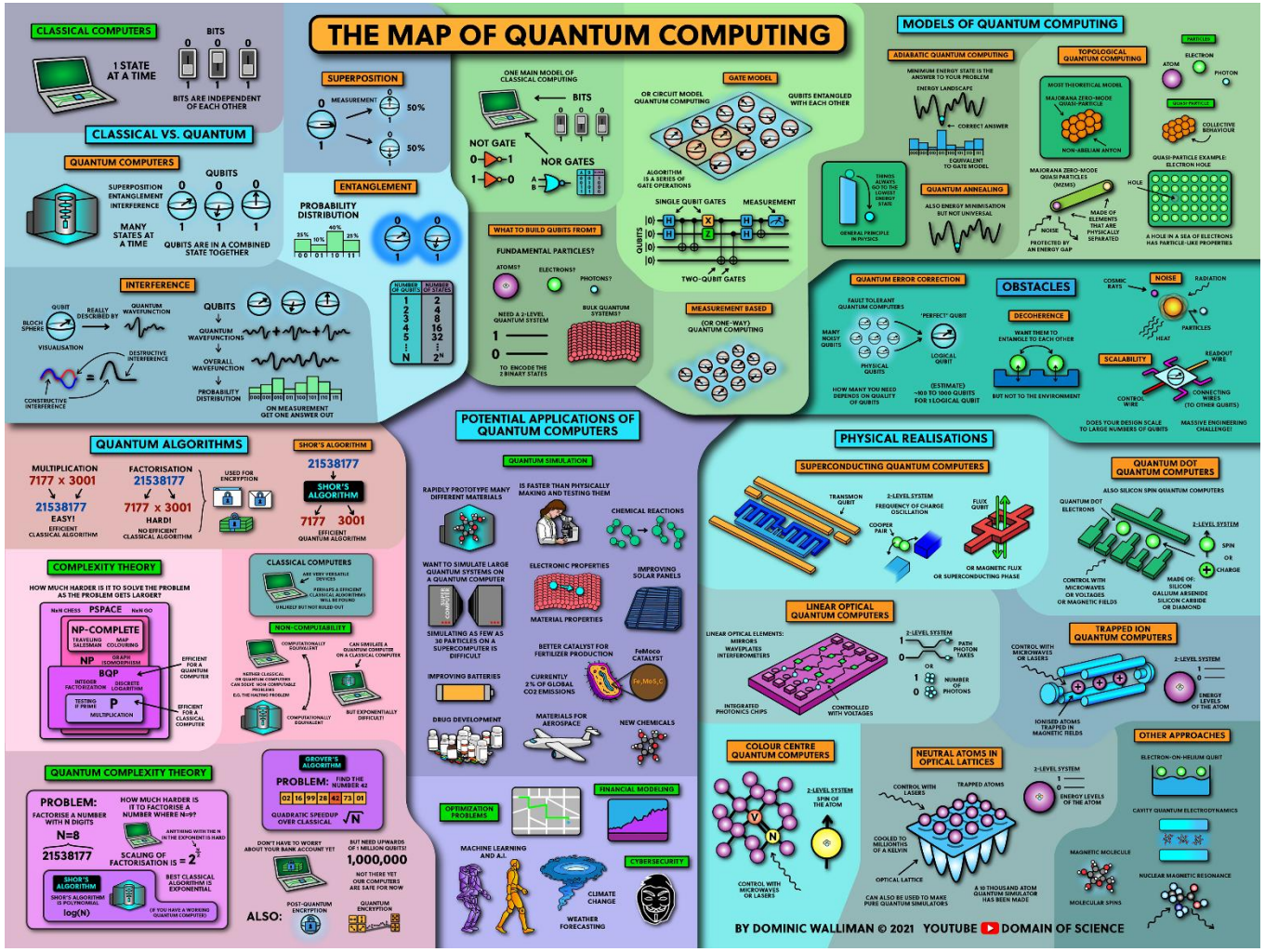
INFORMS Quantum Computing and Operations Research (QCOR), est. Oct. 2024

- Co-Chairs: Tamás Terlaky (Lehigh U.), David Bernal Neira (Purdue U.)
- Members: Giacomo Naniccini (USC), Rebekah Hermann (UTK), Ruslan Shaydulin (JPMorgan Chase), Stefan Woerner (IBM Zurich), Carleton Coffrin (LANL), Mohammadhossein Mohammadisiahroudi (U. of Maryland), Gizem Özbaygın (Bilkent U.), Ojas Parekh (Sandia National Labs)



Quantum Computing

You might have heard about it



[Domain of Science Dominic Walliman]

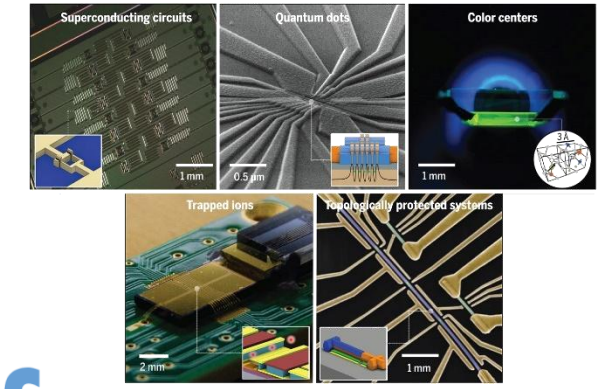
Introduction to Quantum Computing

- Practical and functional differences of Quantum Computing (QC) and Classical Computing (CC)
  - Promise to accelerate certain computational tasks
  - Few available experimental evidence of advantage
- Significant progress in algorithms driven by research
  - QC does not extend CC computability, distinction in matter of efficiency
- Progress in hardware driven by interest from public and private sectors
- Progress in software ecosystem driven by growing community

Retrieved from doi.org/10.1038/s41586-019-1666-5, doi.org/10.1038/d41586-020-03434-7, doi.org/10.1103/PhysRevLett.127.180501

Quantum Algorithm Zoo

Retrieved from Quantum Algorithm Zoo quantumalgorithmzoo.org/

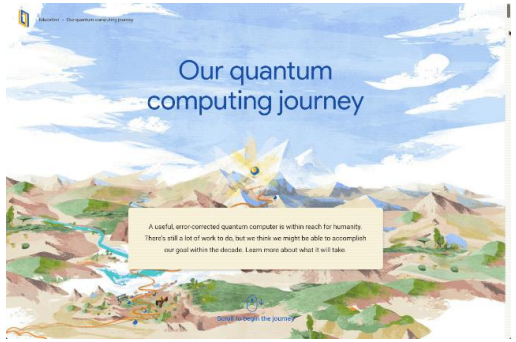


Retrieved from DOI: 10.1126/science.abb2823

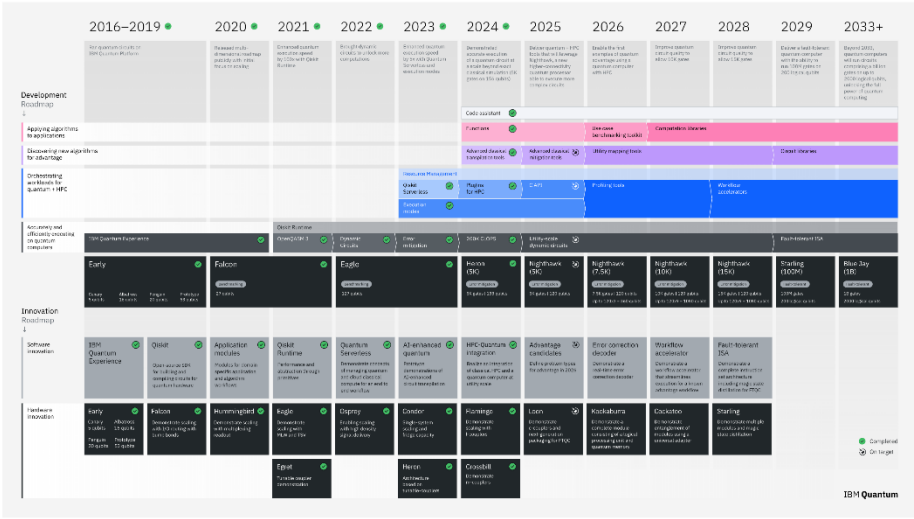
Retrieved from Quantum Open Software Foundation qosf.org/

Introduction to Quantum Computing

- Progress in 40 years but still not achieving “full promise of QC”
- We are living in the Noisy Intermediate Scale Quantum (NISQ) devices era
  - Moderate size (~100s qubits) devices
    - **Too many for classical simulation**
    - **Too few for error correction**
  - Devices subject to physical noise
    - **Distinction of physical and logical qubits**
- Fault-tolerance estimated to be reached soon (?)



Retrieved from quantumai.google/learn/map

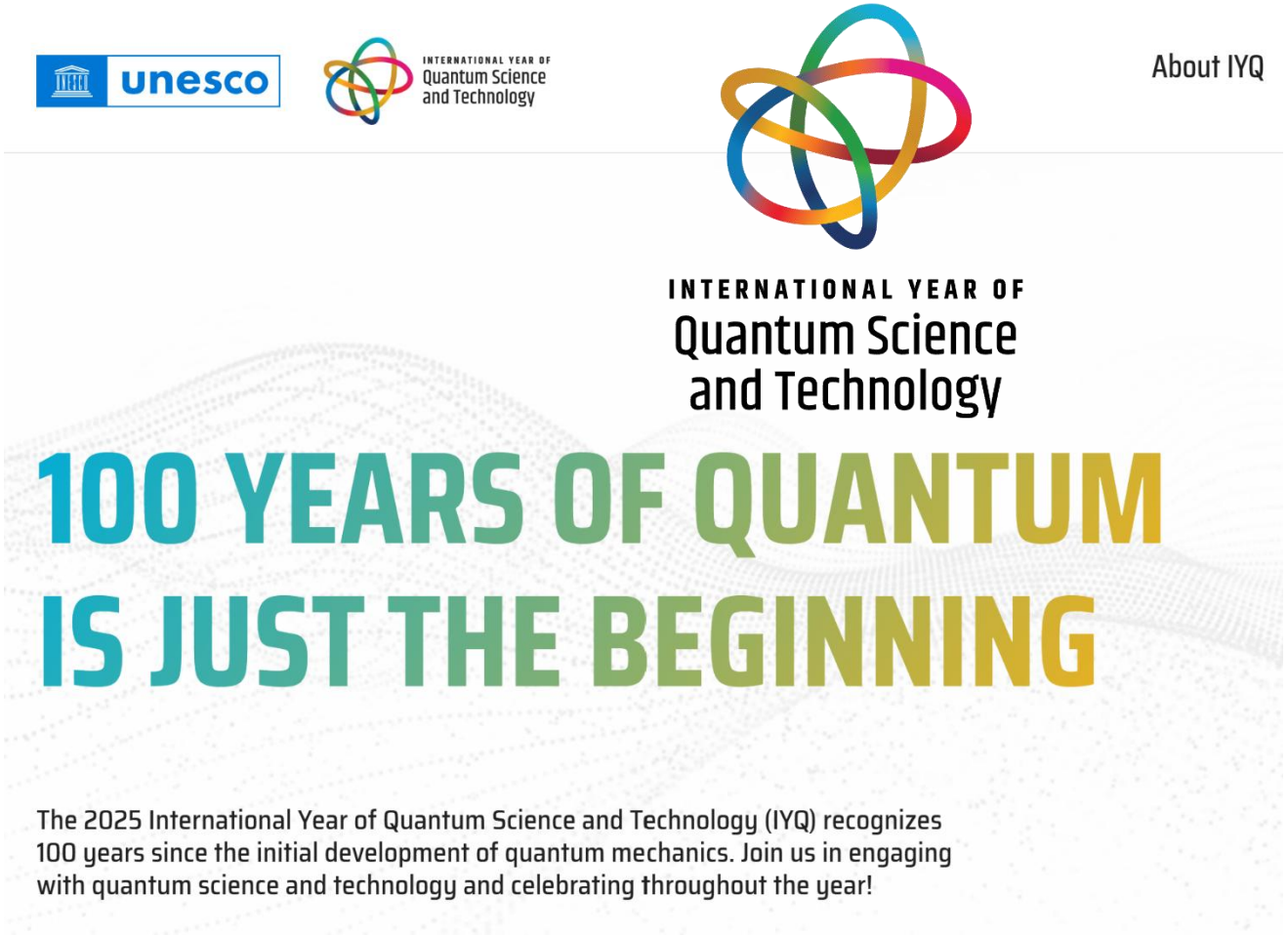


Retrieved from https://www.ibm.com/quantum/technology

- Advantage experiments highlight the physical possibility of speedups, it is now an engineering challenge
  - Observed for tailored problems **not corresponding to** any industrial application **yet***

Happy (Belated) Quantum Year!

According to the UN!



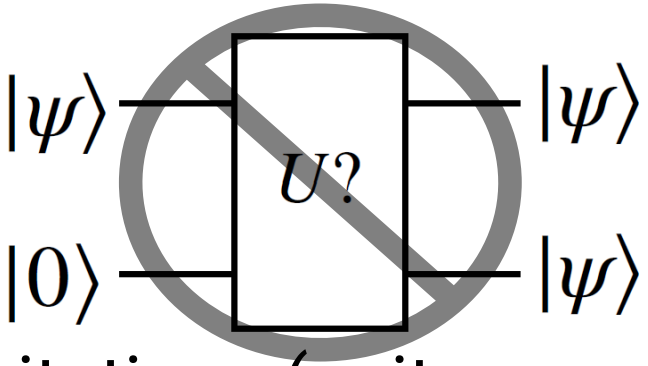
Challenges and uses of quantum computing

Challenges of QC – No copying (states)

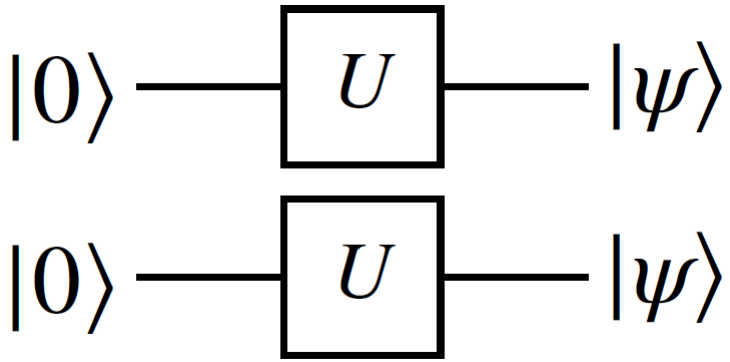
Imagine you have an arbitrary quantum state $|\psi\rangle$ and you would like a copy of it.

- You need to design unitary matrix U that takes in one state and outputs two

This is not possible! See “no-cloning theorem”



In practice we can do this!



Even with all those limitations (unitary operators), the quantum gate model is Turing complete!

Opportunities for QC

- **What useful computations can you do with unitary transforms on exponentially large matrices?**

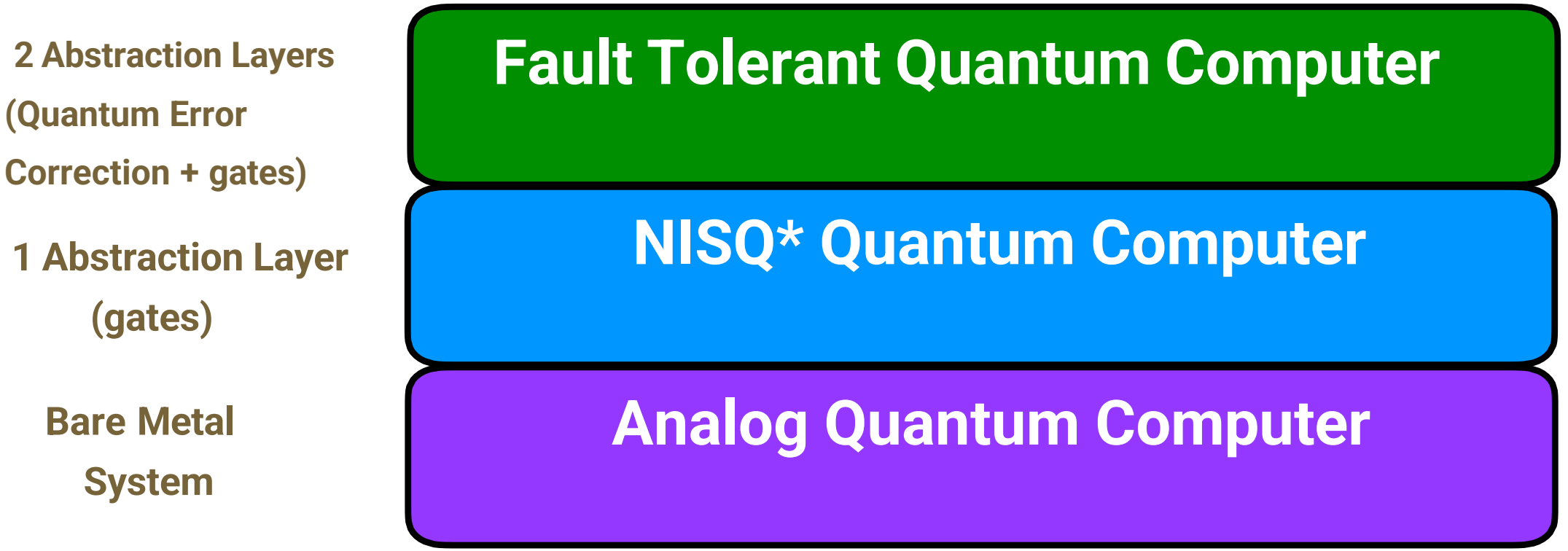
- **Only a few great answers… *so far!***
  - Simulate Quantum Systems (solve Schrödinger Equation)
  - Factor Integers (Shor's Algorithm)
  - Unstructured Search (Grover's Algorithm)
  - Solving Linear Systems of Equations Approximately (HHL)
  - Simulating Large Collections of Oscillating Springs (arXiv:2303.13012)

"Quantum Algorithm Implementations for Beginners"

arXiv:1804.03719

Types of Quantum Computers

Models of Quantum Computation



Analog Quantum Computation

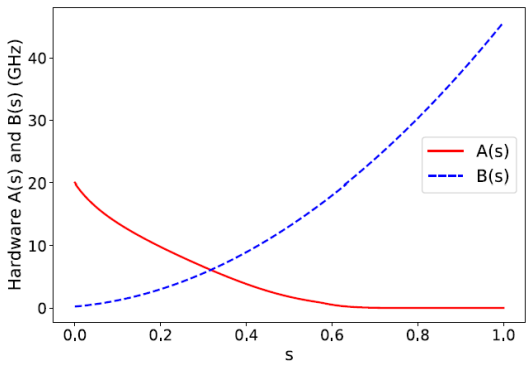
- What does it do?
  - Apply control parameters to evolve a real-world quantum system in time
- Natural use cases
  - Quantum Simulation (open and closed), Optimization
- Arguments For
  - Very efficient use of quantum hardware resources
  - Very fast (no abstraction layers)
- Arguments Against
  - Highly specialized to specific applications (i.e. limited Hamiltonian options)
  - Hardware noise can be limiting

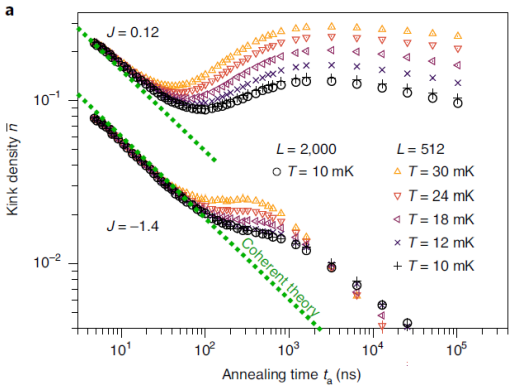
Solve this ODE

$$t \in [0, T], |\psi_0\rangle$$

Time evolution, initial state

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





$$H_{D-Wave}(t) = A(t)\left(\sum_i \hat{\sigma}_i^X\right) + B(t)\left(\sum_{i,j} J_{ij}\hat{\sigma}_i^Z\hat{\sigma}_j^Z + \sum_i \hat{\sigma}_i^Z\right)$$

1. King, Andrew D., et al. "Coherent quantum annealing in a programmable 2,000 qubit Ising chain." Nature Physics 18.11 (2022): 1324-1328.
2. Bernal Neira, David. "Coherent simulation with thousands of qubits." Nature Physics 18.11 (2022): 1273-1274.

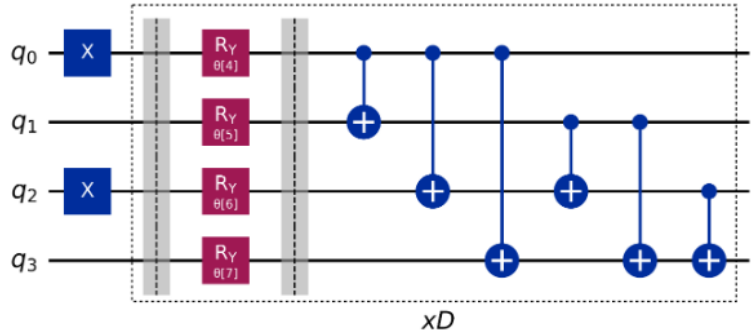
NISQ Gate-Based Computation

- What does it do?
  - Apply quantum imperfect gates to implement a noisy universal quantum computation
  - Note: these usually include arbitrary rotation gates
- Natural use cases
  - Quantum Simulation (closed), Variational Quantum Algorithms, Quantum Machine Learning (?)
- Arguments For
  - Flexible (all types of quantum computations are in scope)
  - Fast (just 1 abstraction layer)
- Arguments Against
  - Hardware noise can be limiting
  - Unclear how to extend beyond qubit coherence time

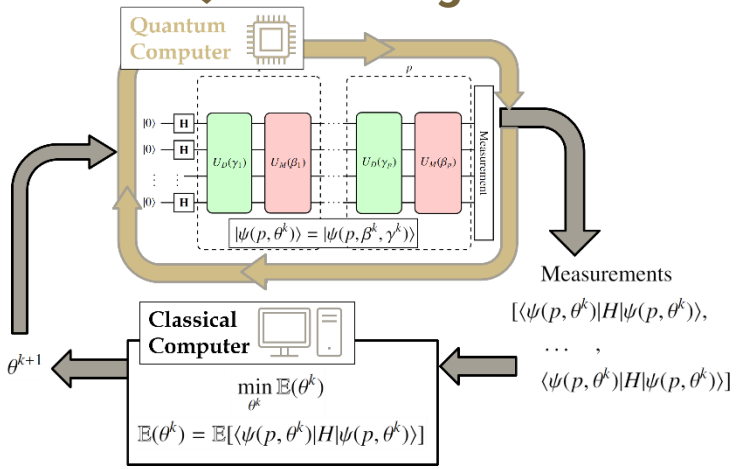
Apply Unitary matrices (U) to a quantum state (ψ)

$$U_n|\dots|U_2|U_1|\psi\rangle$$

Quantum Circuit



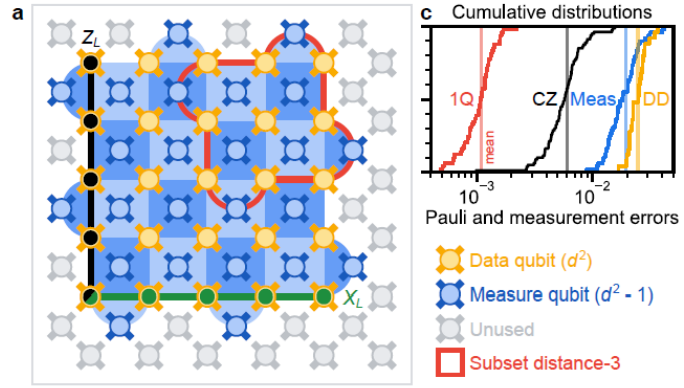
Variational Quantum Algorithms

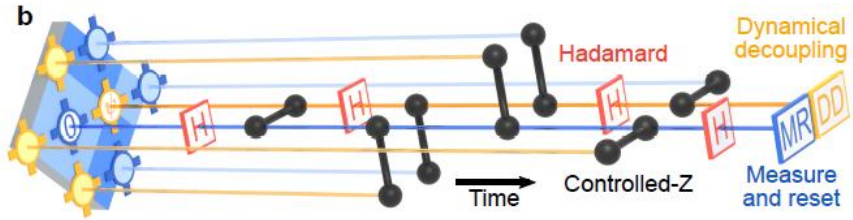


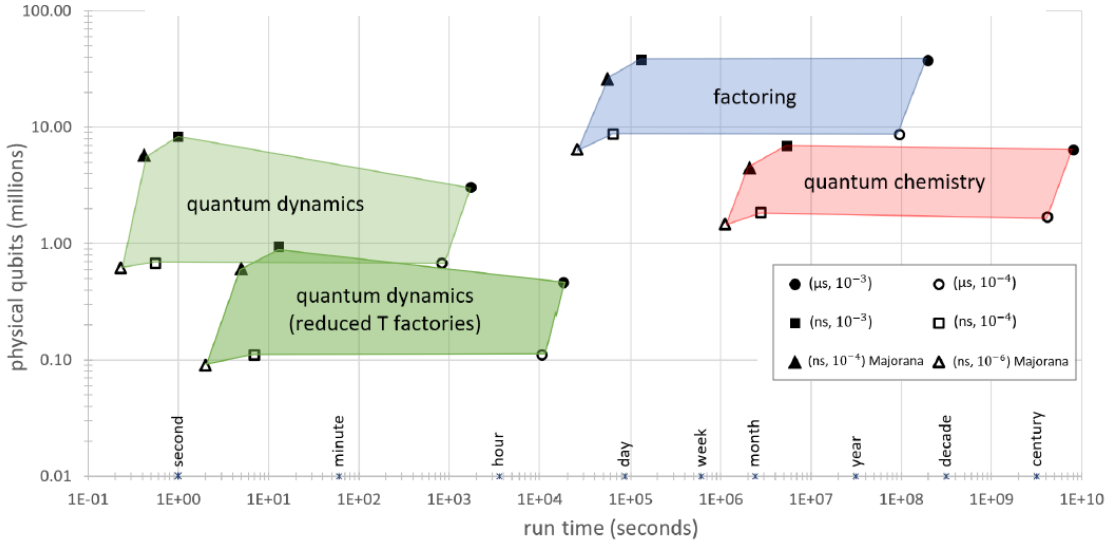
1. Bernal Neira, David E., et al. "Utilizing modern computer architectures to solve mathematical optimization problems: A survey." Computers & Chemical Engineering (2024): 108627.

Fault-Tolerant Gate-Based Computation

- What does it do?
  - Apply quantum error corrected gates to implement perfect universal quantum computation
  - Note: a restrictive gate set (e.g., Clifford+T)
- Natural use cases
  - Quantum Simulation (high-accuracy), Factoring, Linear Systems, Nonlinear Systems (?)
- Arguments For
  - Reliable, Algorithms with Proofs
  - Quantum Error Correction (QEC) enables going beyond the limit of qubit coherence time
- Arguments Against
  - Requires a LOT of physical qubits
  - Algorithms require a LOT of gates (e.g., >106)
  - Slow, QEC overheads take time (2 abstraction layers)







1. Google Quantum AI "Suppressing quantum errors by scaling a surface code logical qubit." Nature 614, no. 7949 (2023): 676-681.
2. Beverland, Michael E., et al. "Assessing requirements to scale to practical quantum advantage." arXiv preprint arXiv:2211.07629 (2022).

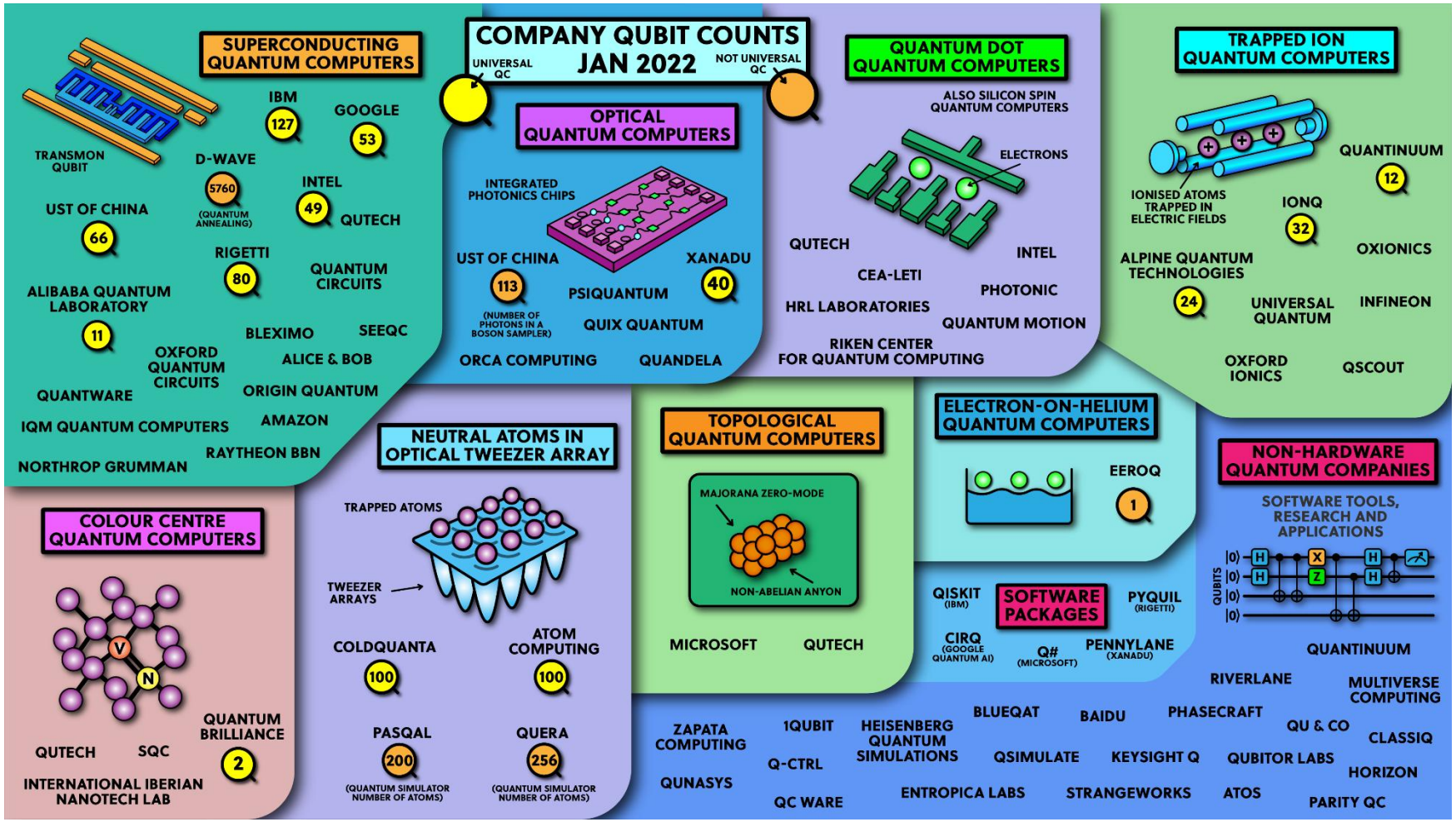
Who is doing what – 3 types of Quantum Computers

	Available Today		Available Soon?
	Noisy Analog	Noisy Gate-Based	Fault-Tolerant Gate-Based
Who?	D-Wave QuEra PASQAL	IBM Q Quantum AI IONQ rigetti XANADU QUANTINUUM qci	PsiQuantum atom computing D-Wave qci Quantum AI ALICE & BOB **Public Roadmaps** IBM Q PASQAL Infleqtion QuEra

1. Carleton Coffrin, Los Alamos National Laboratory

WHY SO MANY?

WE DON'T HAVE YET THE QUANTUM "TRANSISTOR"



[Domain of Science Dominic Walliman]

State of the affairs of Quantum Computing as of 7/1/2026

- Few <u>limited</u> quantum speedup demos achieved with quantum annealing (both application and tailored).
- No quantum advantage for application problems*.
- Three quantum supremacy demos on «self-simulation» - *Sycamore*, *Jiuzhang, Zuchongzhi*

Article | Published: 23 October 2019

Quantum supremacy using a programmable superconducting processor

Frank Arute, Kunal Arya, [...] John M. Martinis

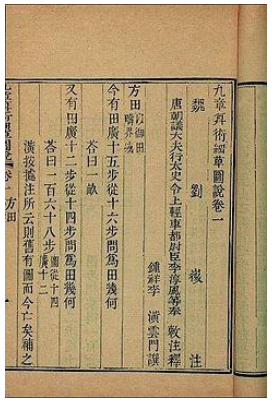
Nature **574**, 505–510(2019) | Cite this article

Quantum computational advantage using photons

Han-Sen Zhong[1,2,*], Hui Wang[1,2,*], Yu-Hao Deng[1,2,*], Ming-Cheng Chen[1,2,*], Li-Chao Peng[1,2], ...

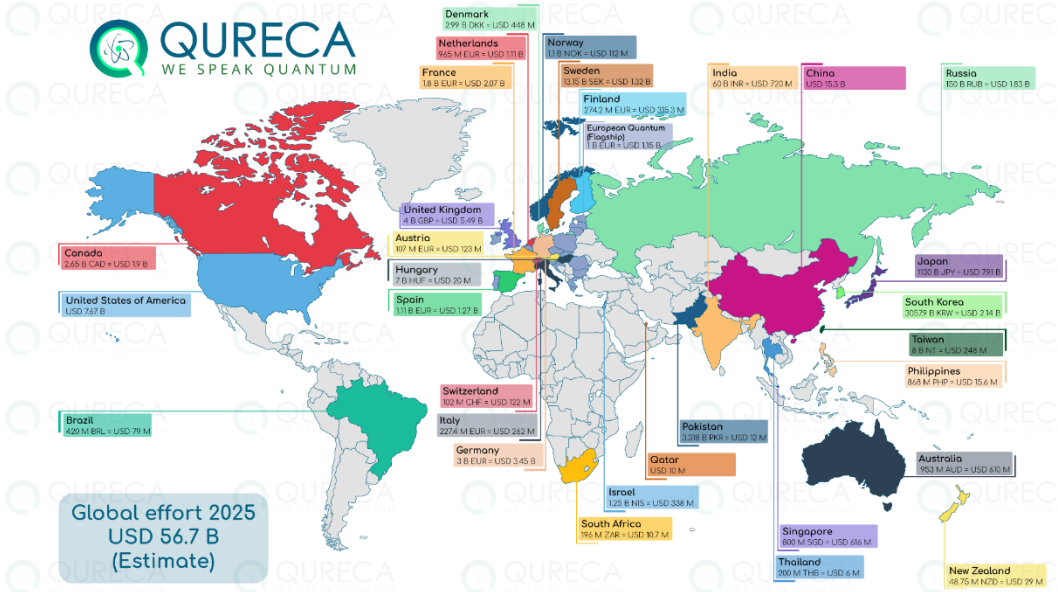
+ See all authors and affiliations

Science 03 Dec 2020:
eabe8770
DOI: 10.1126/science.abe8770



WORLD QUEST:

Discover an <u>interesting, useful</u> application where we see quantum advantage



- Sep 19 Paper Leaks where problem is solved in 5min in Quantum Computer
- Oct 19 IBM pre-announce counter-test estimate (days)
- Oct 19 Google announce (10k years)
- May 2020 Alibaba announce counter-test estimate (20 days)
- Followup works seem to bring the number down to days

- In the latest case: Quantum computer took 1.2 hrs in a task that would take 8 years in the most powerful computer with the corrected estimates

1. Arute, F., Arya, K., Babbush, R. et al. Quantum supremacy using a programmable superconducting processor.
2. Zhong, H., Wang, H., Deng, Y. et al. Quantum computational advantage using photons
3. Wu et al. Strong quantum computational advantage using a superconducting quantum processor
4. https://thequantumdaily.com/2020/09/10/global-quantum-ecosystem-an-overview-on-quantum-initiatives-worldwide/

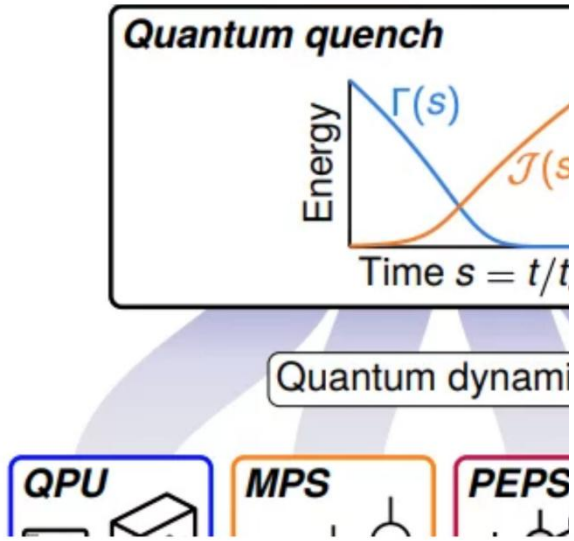
Breaking news!

THE QUANTUM INSIDER

News Exclusive

D-Wave-Led Research Team Reports (
Advantage In Quantum Simulation Tas

Research Matt Swayne • March 7, 2024



Science

Current Issue First release papers Archive About Submit manuscript

HOME > SCIENCE > VOL. 388, NO. 6743 > BEYOND-CLASSICAL COMPUTATION IN QUANTUM SIMULATION

RESEARCH ARTICLE | QUANTUM INFORMATION

Beyond-classical computation in quantum simulation

ANDREW D. KING, ALBERTO NOCERA, MAREK M. RAMS, JACEK DZIARMAGA, ROELAND WIERSEMA, WILLIAM BERNOUDY, JACK RAYMOND, NITIN KAUSHAL, NICLAS HEINSDORF, [...], AND MOHAMMAD H. AMIN +53 authors Authors Info & Affiliations

SCIENCE • 12 Mar 2025 • Vol 388, Issue 6743 • pp. 199-204 • DOI: 10.1126/science.ado6285

47,719 42 CHECK ACCESS

Editor's summary

Quantum computers should be able to solve certain problems that classical computers cannot; however, at the current stage of development, imperfections in quantum computing hardware diminish this comparative advantage. King *et al.* contrasted the performance of their quantum annealing processor to state-of-the-art classical simulations of topical problems such as the quantum dynamics of the transverse-field Ising model. The researchers found that across a range of graph topologies, the quantum processor was able to outperform classical simulations. The results provide a challenge to classical computing, in which method improvement has in the past tempered claims of quantum advantage. —Jelena Stajic

Breaking news 2!

J.P.Morgan | CHASE

Solutions · Who We Serve · Insights · About Us · Careers · News · Con

Technology at our firm > News >

JPMorganChase, Quantinuum, Argonne National Laboratory, Oak Ridge National Laboratory and University of Texas at Austin advance the application of quantum computing to potential real-world use cases beyond the capabilities of classical computing

TECHNOLOGY >

JPMorganChase, Quantinuum, Argonne National Laboratory, Oak Ridge National Laboratory and University of Texas at Austin advance the application of quantum computing to potential real-world use cases

The joint research team achieves a quantum computing milestone, realizing Certified Quantum Rand

nature

Explore content · About the journal · Publish with us

nature > articles > article

Article | Open access | Published: 26 March 2025

Certified randomness using a trapped-ion quantum processor

Minzhao Liu, Ruslan Shaydulin, Pradeep Niroula, Matthew DeCross, Shih-Han Hung, Wen Yu Kon, Enrique Cervero-Martín, Kaushik Chakraborty, Omar Amer, Scott Aaronson, Atithi Acharya, Yuri Alexeev, K. Jordan Berg, Shouvanik Chakrabarti, Florian J. Curchod, Joan M. Dreiling, Neal Erickson, Cameron Foltz, Michael Foss-Feig, David Hayes, Travis S. Humble, Niraj Kumar, Jeffrey Larson, Danylo Lykov, ... Marco Pistoia + Show authors

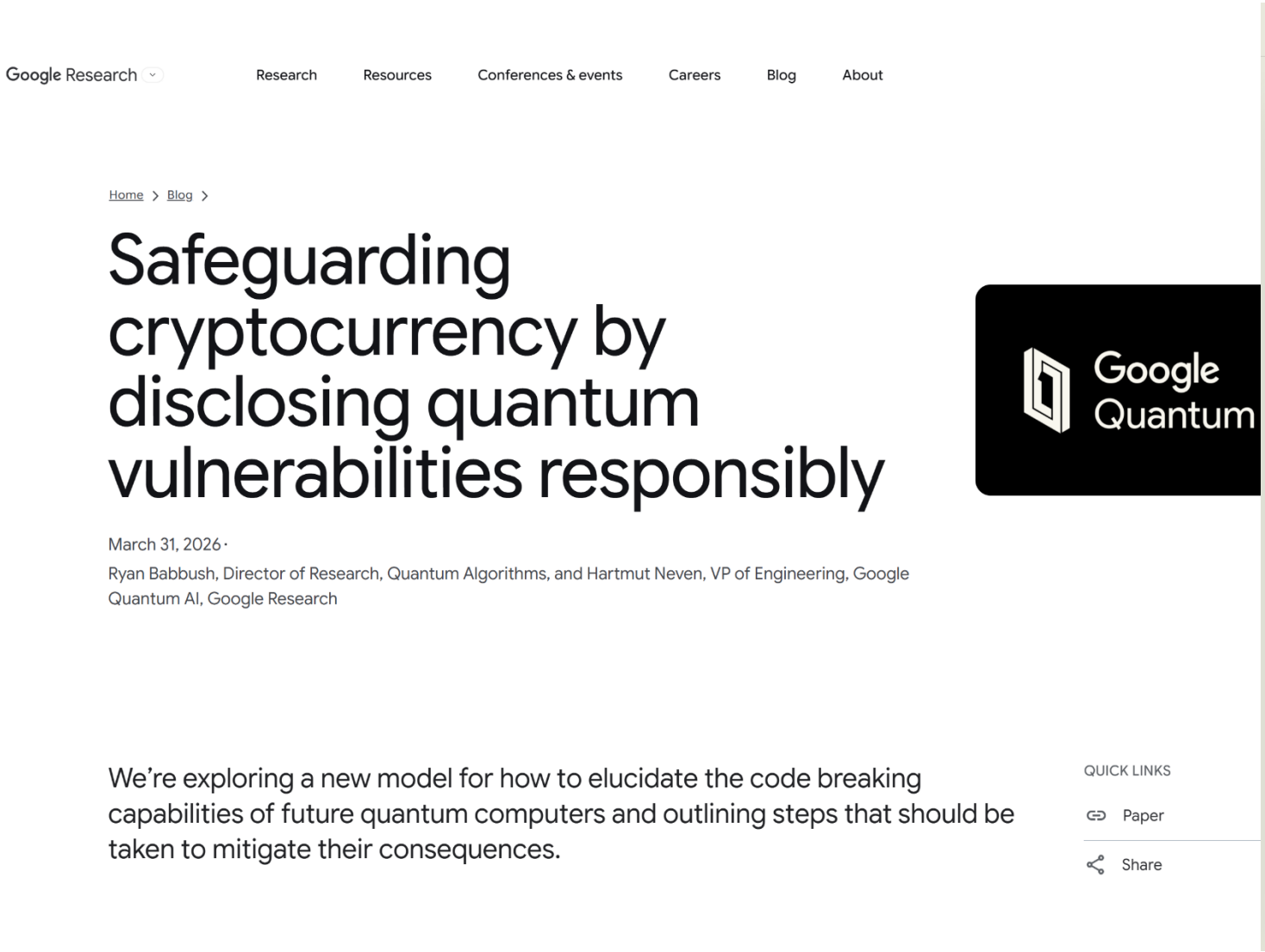
Nature **640**, 343–348 (2025) | Cite this article

This article has been updated

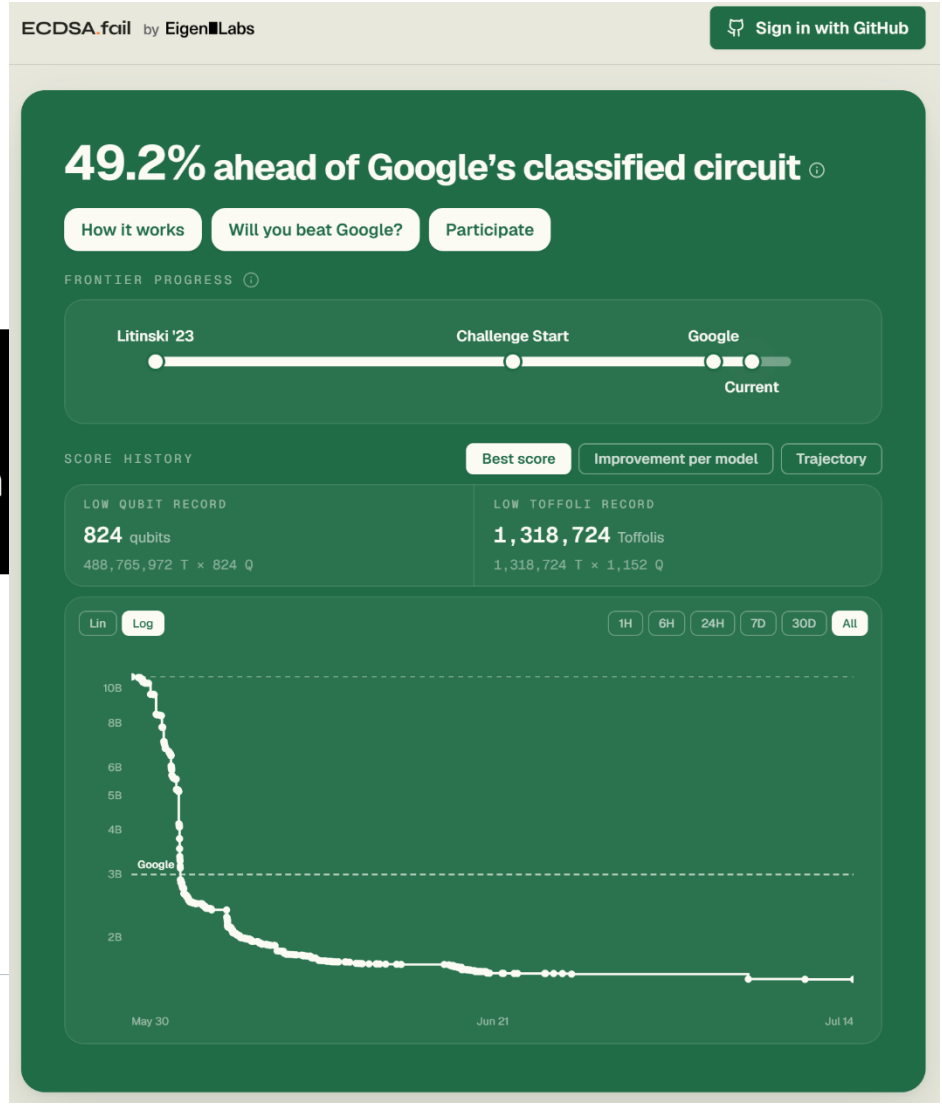
Abstract

Although quantum computers can perform a wide range of practically important tasks beyond the abilities of classical computers[1,2], realizing this potential remains a challenge. An example is to use an untrusted remote device to generate random bits that can be certified to contain a certain amount of entropy[3]. Certified randomness has many applications but is impossible to achieve solely by classical computation. Here we demonstrate the generation of certifiably random bits using the 56-qubit Quantinuum H2-1 trapped-ion quantum computer accessed over the Internet. Our protocol leverages the classical hardness of recent random

Breaking news 3!

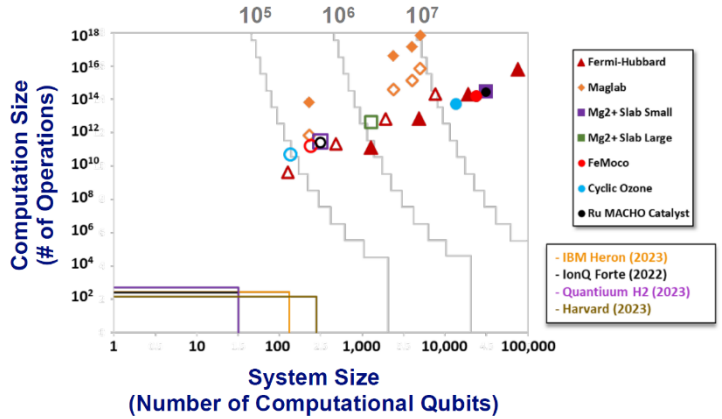


Google Quantum



Current gap between Hardware and applications

- What will it take to achieve high-impact quantum computing applications?
  - Very limited understanding…
  - DARPA's Quantum Benchmarking program is at the bleeding edge
- Preliminary 2023 Findings



POLITICO

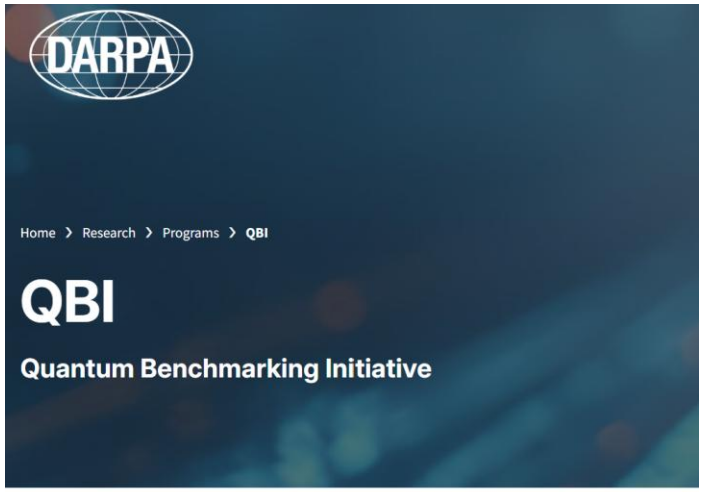
Washington's quantum reality check

By AARON MAK | 07/08/2026 04:01 PM EDT | Updated 07/08/2026 04:33 PM EDT

The White House held a quantum innovation summit on Tuesday. | White House handout.

Quantum is in the air in Washington.

The White House held a summit with quantum companies on Tuesday to discuss innovation in the sector. IBM, PsiQuantum, D-Wave and Quantinuum told DFD that they were among the companies present.



QBI goals

In the simplest terms, the Quantum Benchmarking Initiative (QBI) seeks to determine whether it's possible to build an industrially-useful computer by 2033. Specifically, QBI is designed to rigorously verify and validate whether any quantum computing approach can achieve utility-scale operation — meaning its computational value exceeds its cost.

QBI is not a competition between performers: DARPA is interested in evaluating all viable approaches for which there is available funding. Successful performers will progress through three stages:

- Stage A: Describe a **utility-scale quantum computer concept** that has a plausible path to realization in the near term.
- Stage B: Describe a **Research and Development Plan** capable of realizing the utility-scale quantum computer, the risks associated with that plan and the planned risk mitigation steps, and the prototypes needed to burn down these risks. | Learn more about Stage B teams
- Stage C: Work with the Government to **Verify and Validate** that their utility-scale quantum computer concept can be constructed as designed and operated as intended.

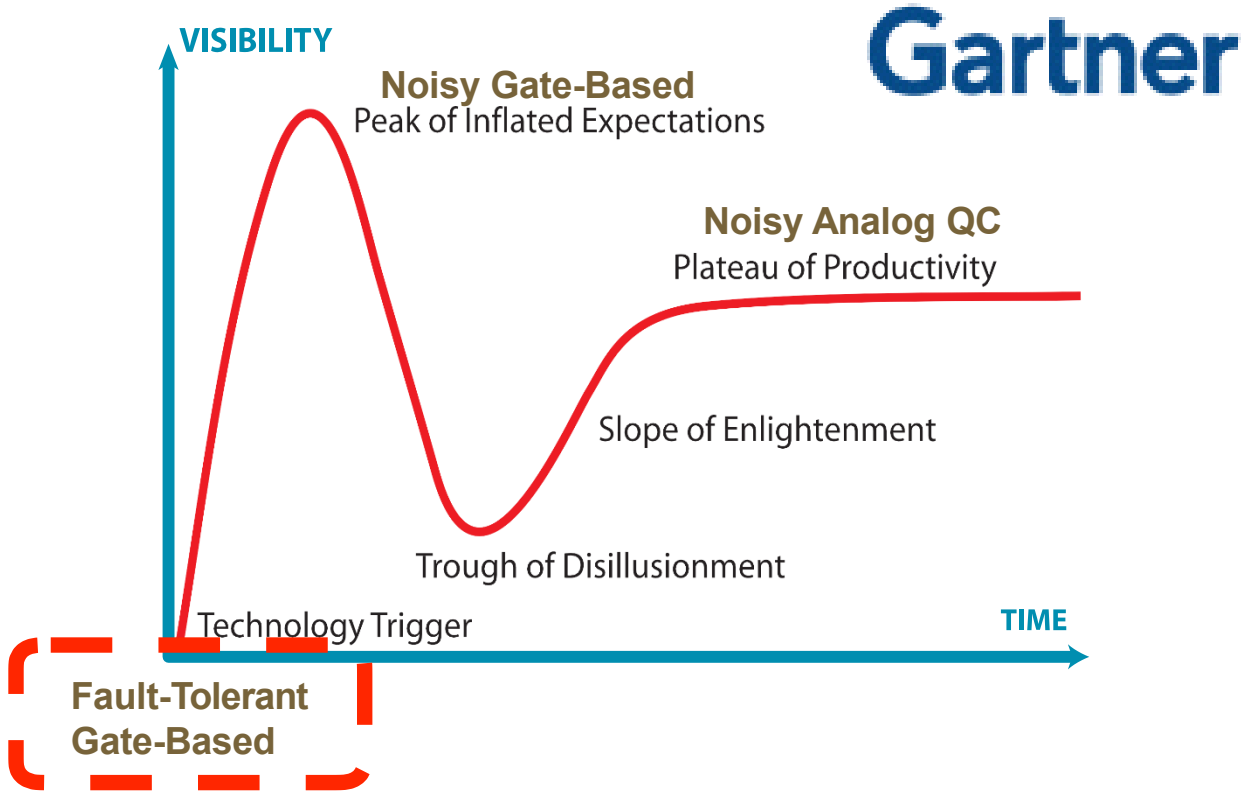
Who will win the Quantum Computing race?

- I don’t know!
- Each player has a plausible pitch where their tech is the best!

How long before we have a very large and useful Fault-Tolerant quantum computer?

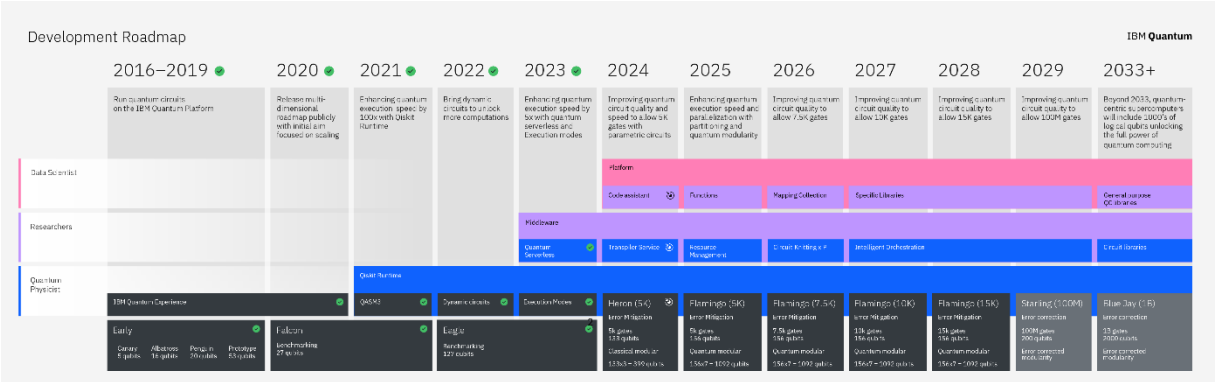
12 months ago most folks would say “10-15 years”

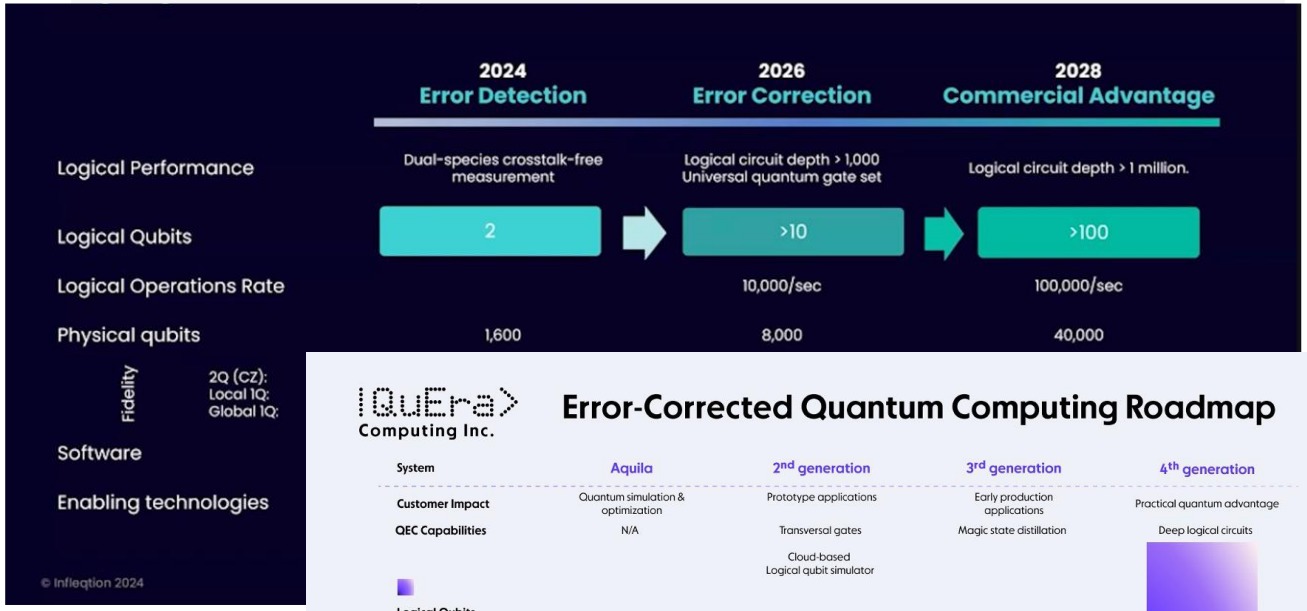
Today it seems much sooner (2026-2030)

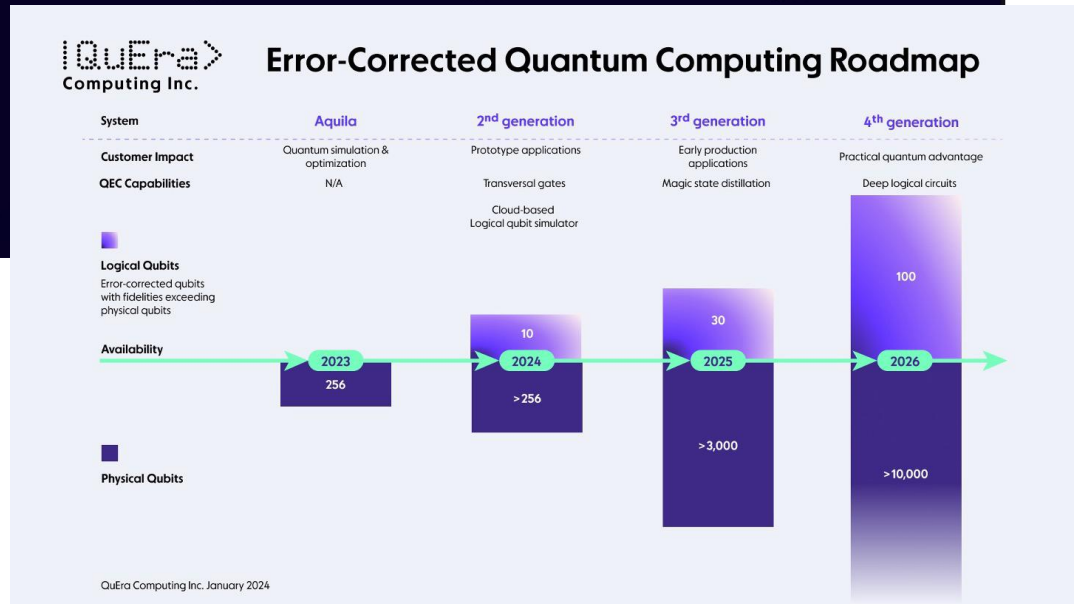


Public FTQC Roadmaps

- IBM
  - 200 logical qubits by 2029
  - 100M Gates
- Infleqtion
  - 100 logical qubits by 2028
  - 1-100M Gates
- QuEra
  - 100 logical qubits by 2026
    - Today, they have 48







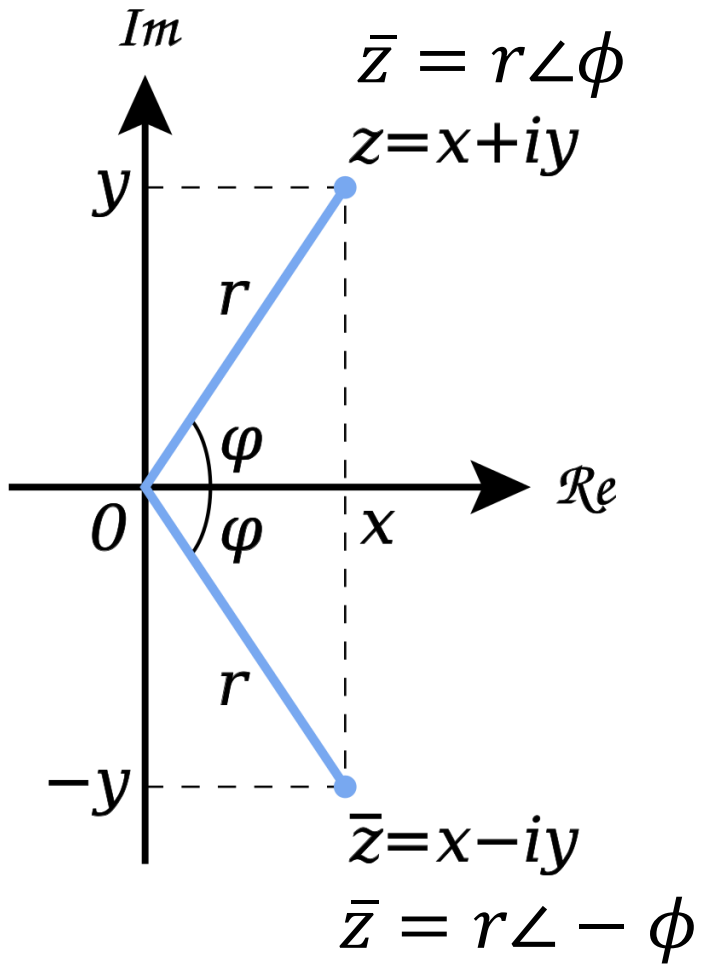
Overview of Quantum Mechanics: Basics to understand and use quantum computing

Thanks to Carleton Coffrin from Los Alamos National Laboratory, Davide Venturelli from USRA and NASA QuAIL, Giacomo Nannicini from USC (before at IBM Quantum), and many more for inspiration for these slides!

Preliminaries

Refresher on complex numbers

- A complex number has two parts
  - "Real" horizontal axis
  - "Imaginary" vertical axis
- Can be represented in polar form as a magnitude r and an angle ϕ
- Two important operations
  - Complex conjugate (i.e., $z^*, \bar{z}$)
  - Complex Conjugate and Matrix Transpose, A^{*T} (i.e., A^H, $A^\dagger$)



1. Based on introduction lecture by Carleton Coffrin
2. https://en.wikipedia.org/wiki/Complex_number
3. https://en.wikipedia.org/wiki/Complex_conjugate

Preliminaries

Kronecker product ⊗ or Tensor Product

- A recursive matrix operation
  - Useful for representing and working efficiently with exponentially growing structures

$$u \otimes v = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \otimes \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_1 v_1 \\ u_1 v_2 \\ u_2 v_1 \\ u_2 v_2 \\ u_3 v_1 \\ u_3 v_2 \end{pmatrix}$$

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}$$

$$A \otimes B = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \otimes \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \\ a_{31}b_{11} & a_{31}b_{12} & a_{32}b_{11} & a_{32}b_{12} \\ a_{31}b_{21} & a_{31}b_{22} & a_{32}b_{21} & a_{32}b_{22} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \otimes \begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} = \begin{pmatrix} 1\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} & 2\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} \\ 3\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} & 4\begin{pmatrix} 0 & 5 \\ 6 & 7 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{pmatrix}$$

1. Based on introduction lecture by Carleton Coffrin
2. https://en.wikipedia.org/wiki/Tensor_product

Preliminaries
Dirac Notation (BraKet)

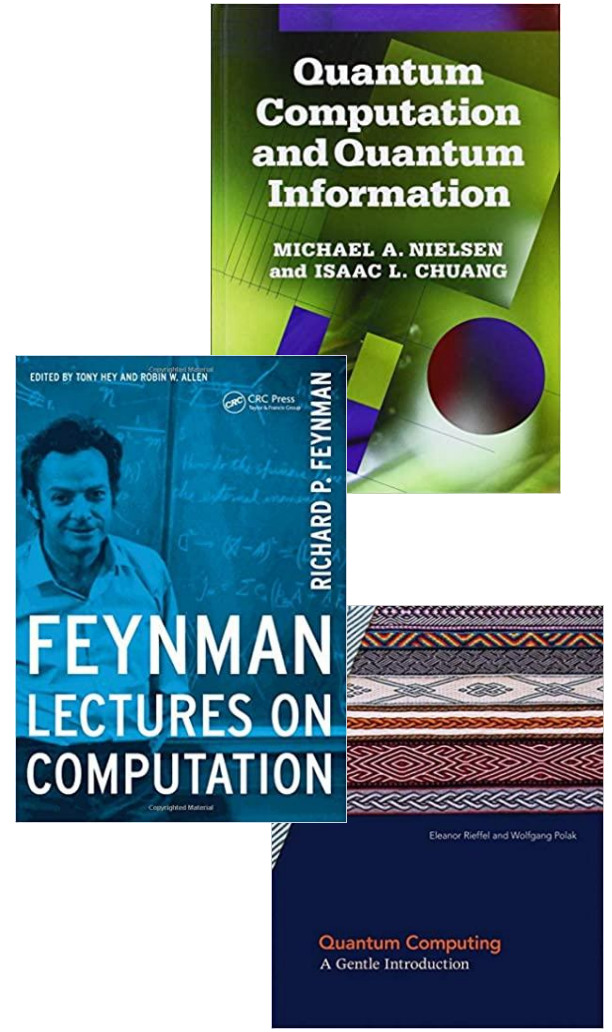
- Given a complex Euclidean space $\mathbb{S} \equiv \mathbb{C}^n$
  - $|\psi\rangle \in \mathbb{S}$ denotes a column vector and is called a **Ket**
  - $\langle\psi| \in \mathbb{S}$ denotes a row vector and is called a **Bra**
    - $\langle\psi|$ is the conjugate transpose of $|\psi\rangle$ (i.e., $\langle\psi| = |\psi\rangle^\dagger$)
  - $\langle\psi|\psi\rangle$ is a scalar and is known as a **BraKet**

- Quantum computing works with large complex Euclidean spaces $\left(\mathbb{C}^2\right)^{\otimes q}$

1. Based on introduction lecture by Carleton Coffrin

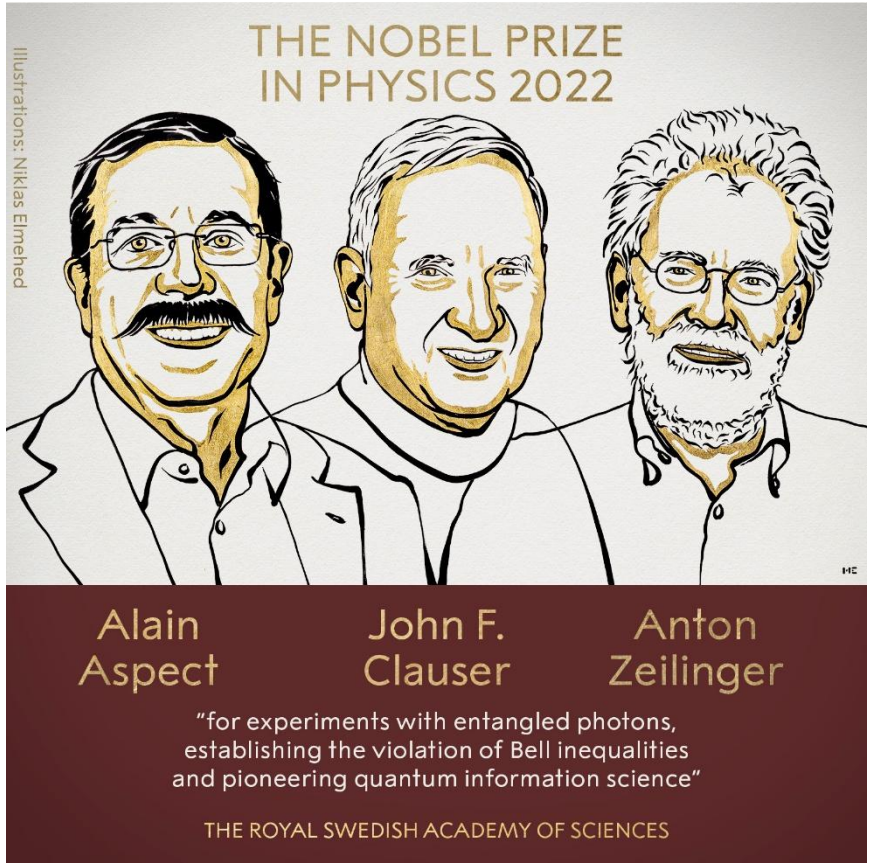
Crash course on Quantum Mechanics

- Quantum Mechanics is the physics theory that describes and predicts the outcome of experiments with systems that are sufficiently small, cold and isolated
- Quantum Computing uses quantum Mechanics as “information processing”
  - EXPERIMENT → COMPUTATION
- Five concepts that are required to USE (not understand!) QM for QC:
  - Quantum State
    - Qubits, Qubit Registers, Wavefunction
  - Quantum Coherent Operations
    - Shrödinger or Unitary Evolution, Gates
  - Measurement
    - Collapse, Projection, Probability Amplitude, Born Rule
  - Interference
    - Particle-wave duality, quasi-probabilities
  - Quantum Incoherent Operations
    - Dissipation, Dephasing, Decoherence, Noise



1. Nannicini, Giacomo. "An introduction to quantum computing, without the physics." SIAM Review 62.4 (2020): 936-981.

How to teach quantum Mechanics and Computing to non-Physicists?



Overview of Quantum Mechanics: Quantum State

The Quantum State (QUBITs)

- A state is a representation of a physical system through a collection of variables that fully describes its physics within the theory (e.g., in thermodynamics, is P, V, T, …)
- A QUBIT is the simplest quantum state you can think of a two-level system

BraKet notation for generic quantum states $|\psi\rangle$

- The two states define the COMPUTATIONAL BASIS $|0\rangle$ and $|1\rangle$

Standard basis

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

{0} State

$$|\psi\rangle = \begin{matrix}|0\rangle\\|1\rangle\end{matrix}\begin{pmatrix}1.0+0.0i\\0.0+0.0i\end{pmatrix}$$

Split between {0,1} states

$$|\psi\rangle = \begin{matrix}|0\rangle\\|1\rangle\end{matrix}\begin{pmatrix}\sqrt{0.5}+0.0i\\\sqrt{0.5}+0.0i\end{pmatrix}$$

- Think of them as the X and Y axis of a fixed-length arrow
  - Unit vector requirement $\langle\psi|\psi\rangle = 1$
- The quantum state is fully specified by its components on the axes which are complex numbers

Notation for qubit states $|\psi\rangle_{qubit} = \psi_0|0\rangle + \psi_1|1\rangle$

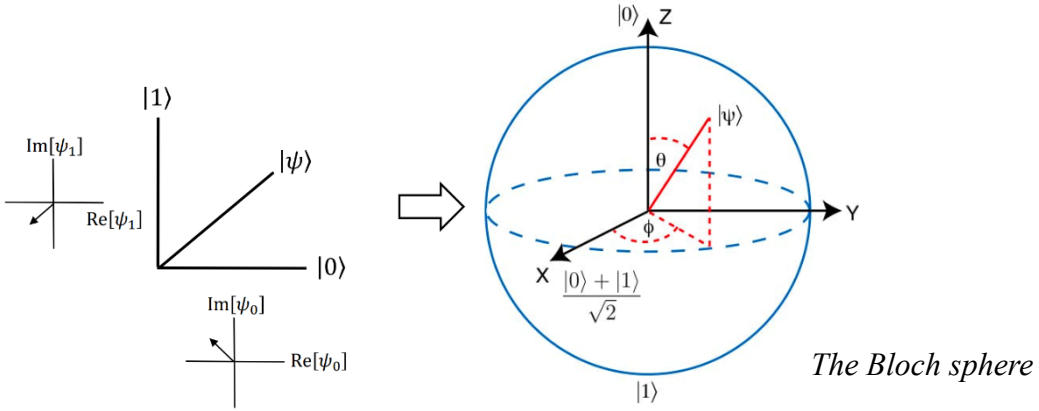
The Quantum State (QUBITs)

- The quantum state is fully specified by its components on the axes which are complex numbers

$$\text{Notation for qubit states } |\psi\rangle_{qubit} = \psi_0|0\rangle + \psi_1|1\rangle$$

- Unit vector requirement $\langle\psi|\psi\rangle = 1$, meaning that

$$|\psi_0|^2 + |\psi_1|^2 = 1$$



The Bloch sphere

Insight: QM/QC is mostly linear algebra with complex numbers

The “Exponentiality” of QM

- One qubit is fully described by its **wavefunction**, i.e., 2 complex numbers
- N qubits are fully described by their global wavefunction, a vector with 2^N complex numbers (**probability amplitudes**)
  - The quantum state vector grows exponentially with the number of qubits (that’s why we need $\bigotimes$)
- Compute over exponentially large vectors!

Size of $|\psi\rangle$ is $2^{|qubits|}$

$$|\psi\rangle = \overset{q_1}{\begin{pmatrix} 1.0+0.0i \\ 0.0+0.0i \end{pmatrix}}, \overset{q_1,q_2}{\begin{pmatrix} 1.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \end{pmatrix}}, \overset{q_1,q_2,q_3}{\begin{pmatrix} 1.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \\ 0.0+0.0i \end{pmatrix}}, \dots,$$

Two qubit example

- Quantum state grows as $\left(\mathbb{C}^2\right)^{\otimes q}$ (q the number of qubits)
- For two qubits we have $\left(\mathbb{C}^2\right)^{\otimes 2} = \mathbb{C}^2 \otimes \mathbb{C}^2$ with the following 4 basis elements

$$|0\rangle_{2\ qubits} = |0\rangle \otimes |0\rangle = |00\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad |1\rangle_{2\ qubits} = |0\rangle \otimes |1\rangle = |01\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

$$|2\rangle_{2\ qubits} = |1\rangle \otimes |0\rangle = |10\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \qquad |3\rangle_{2\ qubits} = |1\rangle \otimes |1\rangle = |11\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

$$|\psi\rangle_{2\ qubits} = \begin{matrix}|00\rangle\\|01\rangle\\|10\rangle\\|11\rangle\end{matrix}\begin{pmatrix}\psi_{00}\\\psi_{01}\\\psi_{10}\\\psi_{11}\end{pmatrix} =_{eg} \begin{pmatrix}0.0+0.0i\\0.0+0.0i\\1.0+0.0i\\0.0+0.0i\end{pmatrix} = |10\rangle$$

Product states

Special states that can be written as Kronecker products of others

$$|\psi\rangle = |\psi\rangle_1 \otimes |\psi\rangle_2$$

- Intuition: separable states, even if they represent two quantum states in a larger system, they are independent (one can be measured without the other)
  - Specific bit vectors or basis states

$$|2\rangle_{2\ qubits} = |1\rangle_1 \otimes |0\rangle_2 = |10\rangle = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}$$

  - Uniform superposition

$$\left(\frac{1}{\sqrt{2}}|0\rangle_1 + \frac{1}{\sqrt{2}}|1\rangle_1\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle_2 + \frac{1}{\sqrt{2}}|1\rangle_2\right) = \begin{pmatrix}1/\sqrt{2}+0.0i\\1/\sqrt{2}+0.0i\end{pmatrix} \otimes \begin{pmatrix}1/\sqrt{2}+0.0i\\1/\sqrt{2}+0.0i\end{pmatrix} = \begin{pmatrix}0.5+0.0i\\0.5+0.0i\\0.5+0.0i\\0.5+0.0i\end{pmatrix}$$

Entangled states

Special states that cannot be written as Kronecker products of others

$$|\psi\rangle \neq |\psi\rangle_1 \otimes |\psi\rangle_2$$

- Intuition: non-separable states, even if they represent two quantum states in a larger system, they are not independent (the measurement of one affects the other). You need at least 2 qubits

Maximally entangled state: Bell pair

$$|\psi\rangle = \begin{matrix}|00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle\end{matrix}\begin{pmatrix}\sqrt{0.5} \\ 0 \\ 0 \\ \sqrt{0.5}\end{pmatrix} = \sqrt{0.5}|00\rangle + \sqrt{0.5}|11\rangle$$

No setting of constants $c_1, \dots, c_8 = c_{[\![8]\!]}$ can yield a Bell state

$$\begin{pmatrix}c_1 + c_2 i \\ c_3 + c_4 i\end{pmatrix} \otimes \begin{pmatrix}c_5 + c_6 i \\ c_7 + c_8 i\end{pmatrix} \neq \begin{pmatrix}\sqrt{0.5} + 0.0i \\ 0.0 + 0.0i \\ 0.0 + 0.0i \\ \sqrt{0.5} + 0.0i\end{pmatrix}$$

Superposition

- The quantum system is in a linear combination of basis states
  - For one qubit if $\psi_0 \neq 1$ or $\psi_1 \neq 1$ (e.g. “uniform” superposition state)

$$|\psi\rangle_{qubit} = \psi_0|0\rangle + \psi_1|1\rangle =_{eg} \begin{matrix}|0\rangle \\ |1\rangle\end{matrix}\begin{pmatrix}\sqrt{0.5} \\ \sqrt{0.5}\end{pmatrix}$$

  - For 2 qubits if $\psi_{ij} \neq 1$ for any $i, j \in \{0,1\}$ (e.g., Bell pair)

$$|\psi\rangle_{2\ qubits} =_{eg} \begin{matrix}|00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle\end{matrix}\begin{pmatrix}0.25 + 0.50i|00\rangle \\ 0.25 - 0.25i|01\rangle \\ 0.50 + 0.50i|10\rangle \\ 0.00 - 0.25i|11\rangle\end{pmatrix}$$
$$= (0.25 + 0.50i)|00\rangle + (0.25 - 0.25i)|01\rangle + (0.50 + 0.50i)|10\rangle + (0.00 - 0.25i)|11\rangle$$

The “Exponentiality” of QM

One **qubit** is **fully described** by its **wavefunction**, i.e., 2 complex numbers

N qubits are fully described by their **global wavefunction**, a vector with 2^N complex numbers (probability amplitudes)

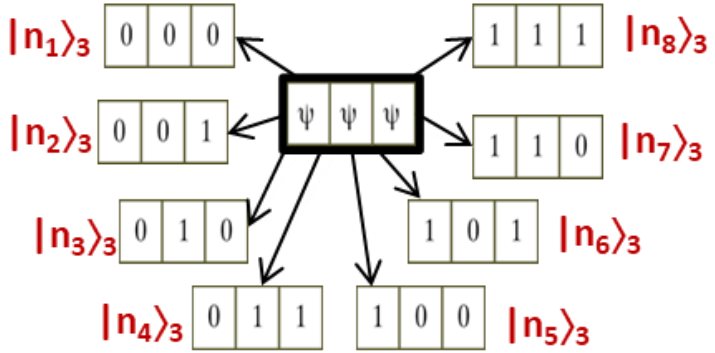
$$|\psi\rangle_{qubit} = \psi_0|0\rangle + \psi_1|1\rangle$$

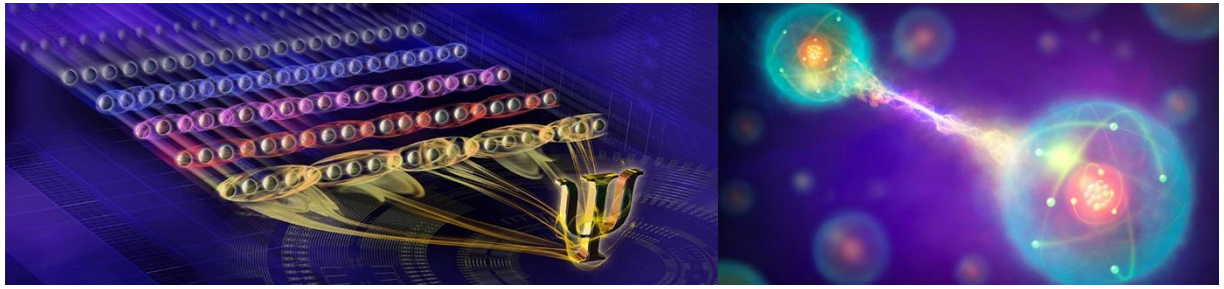
$$|\psi\rangle_{2qubits} = \psi_{00}|00\rangle + \psi_{01}|01\rangle + \psi_{10}|10\rangle + \psi_{11}|11\rangle$$

$$\neq (\psi_0|0\rangle + \psi_1|1\rangle) \otimes (\phi_0|0\rangle + \phi_1|1\rangle)$$

[if it is equal, then it is not entangled/separable]

$$|\psi\rangle_{Nqubits} = \sum_{n=1}^{2^N} \psi_n|n\rangle_n \text{ (binary representation)}$$





Source IEEE

2^{265} ≈ estimated number of atoms in the Universe.

There is no way we can store the general wavefunction of even a processor with more than 60 qubits on earth, let alone process it!

Overview of Quantum Mechanics: Quantum Coherent Operation

Coherent Operations (Gates)

- In quantum physics, the way a system changes state is through the Schrodinger equation.

$$|\psi^I(\Delta t)\rangle = T\exp\left(-\frac{i}{\hbar}\int_0^{\Delta t}\hat{H}_c^I(t)dt'\right)|\psi^I(0)\rangle$$

- In QC, this means we can create a matrix (operator) to transform the states

$$|\psi_a^I\rangle = U|\psi_b^I\rangle \quad \text{where } U \text{ is a unitary matrix, } U^\dagger U = UU^\dagger = 1$$

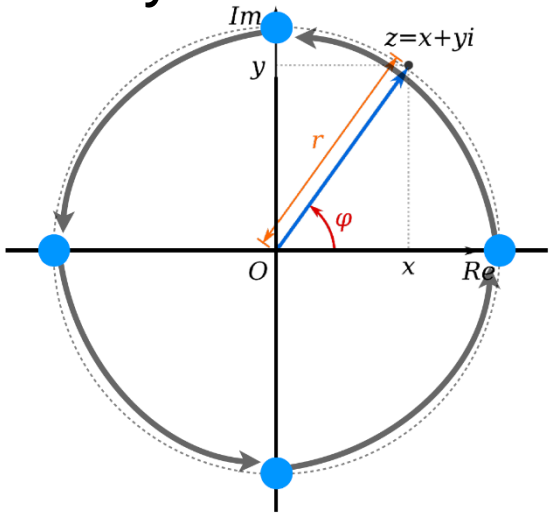
- Since the state of a system of q qubits has size $\mathbb{C}^{2^q}$ then the unitary matrix has size $\mathbb{C}^{2^q \times 2^q}$
  - Example $\psi = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, U = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$

$$U|\psi\rangle = \begin{pmatrix} 0 \\ -i \end{pmatrix}$$

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

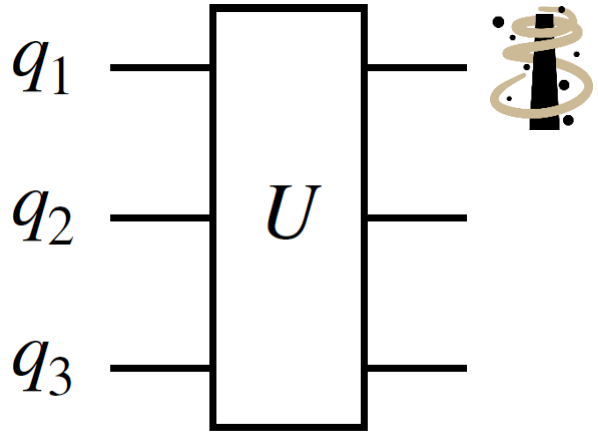
$$UUU|\psi\rangle = \begin{pmatrix} 0 \\ i \end{pmatrix}$$

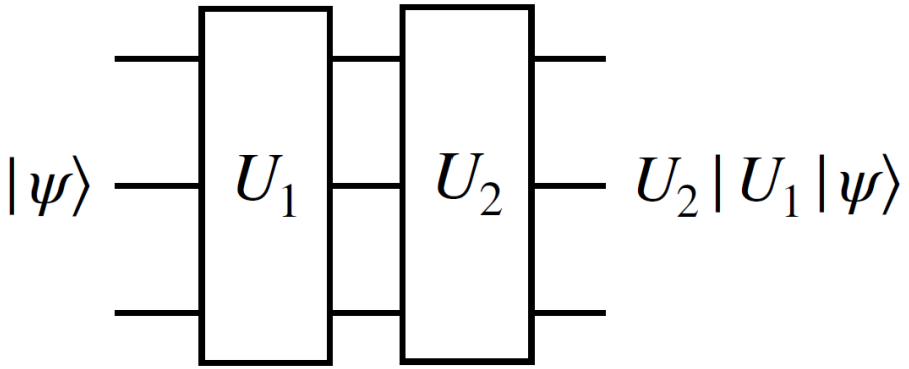
$$UUUU|\psi\rangle = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$



Quantum Circuits

- Quantum programs are often drawn as a score over a number of qubits. Gates appear as blocks in the score
  - Note, in this 3 qubit example has a quantum state of size $\mathbb{C}^8$, the gate U is an $\mathbb{C}^{8\times 8}$ matrix
- Time goes from left to right, and the gates are applied sequentially
- A gate-based quantum circuit or program consists of repeated applications of unitary gates
  - You can have mid-circuit measurements, but outside the scope of this workshop



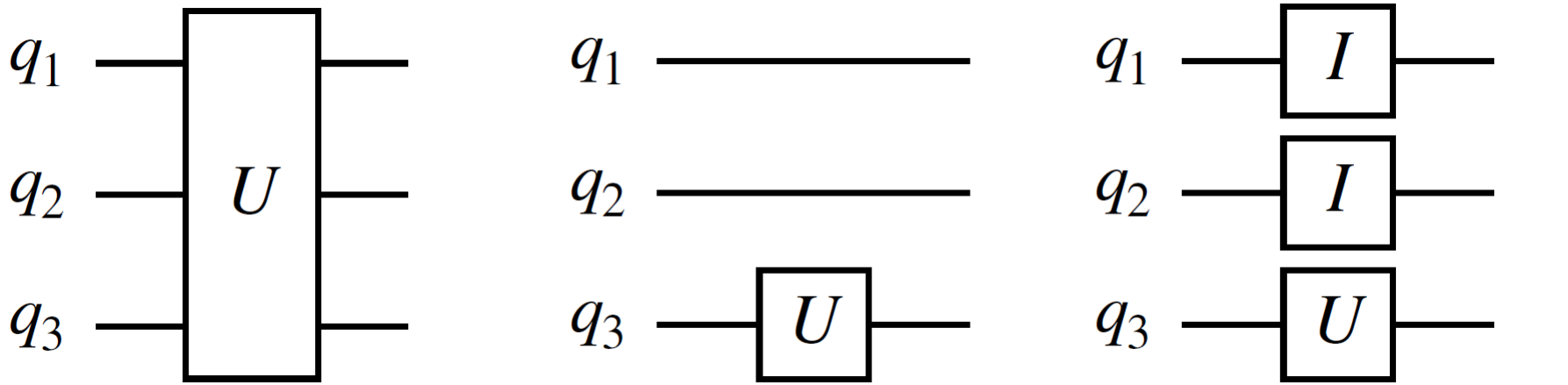


Quantum Program

$U_n| \dots |U_2|U_1|\psi\rangle$

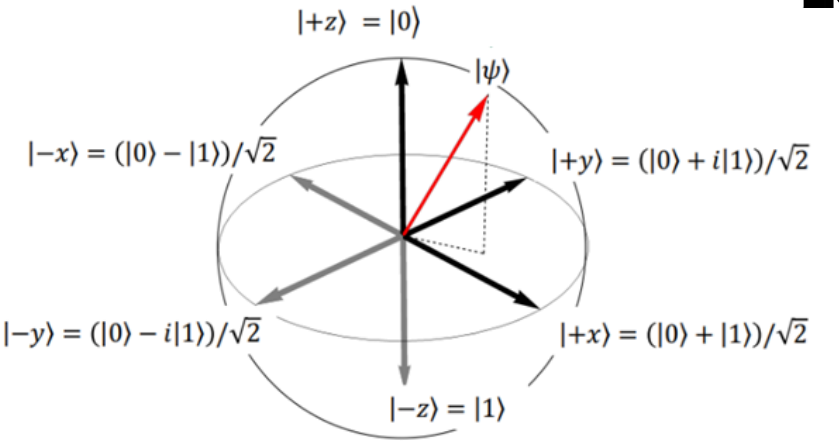
Quantum Circuits with few qubit gates

- Gates can be applied to a subset of qubits
  - Identity matrices expand the operation to all the qubits



$$I \otimes I \otimes U|\psi\rangle$$

1 Qubit Gates



- For one qubit, the gates are matrices in $\mathbb{C}^{2\times 2}$
  - A basis of single-qubit unitaries: **Pauli Gates**

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

  - X gate also known as NOT gate because

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

- Example: single qubit rotation around the X axis **«transverse field»**

$$R_x(\theta) \equiv e^{-i\frac{\theta}{2}X} = \cos\frac{\theta}{2} I - i\sin\frac{\theta}{2} X = \begin{bmatrix} \cos\theta/2 & -i\sin\theta/2 \\ -i\sin\theta/2 & \cos\theta/2 \end{bmatrix}$$

- An arbitrary change can be decomposed in maximum 3 axis rotations (Euler)

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

Obtaining superposition

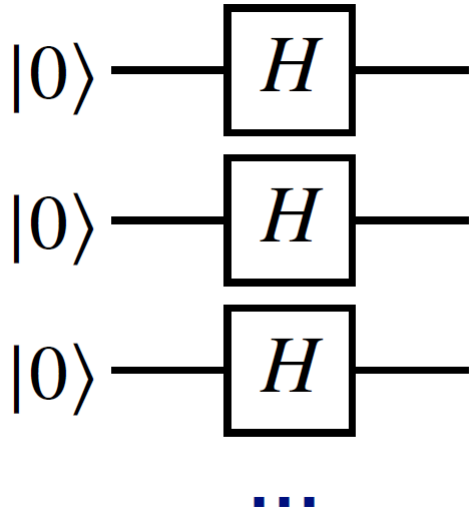
- Hadamard gate (H) generates "uniform" superposition across basis states for one qubit

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

- Applied to each qubit (in a layer) creates a fully superposed state for all qubits

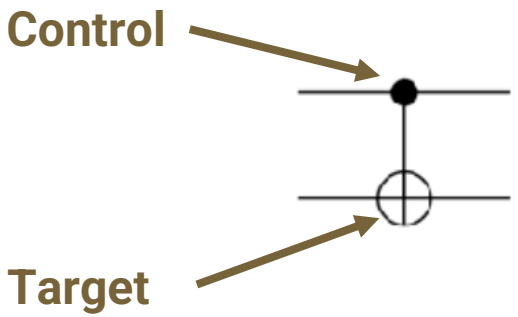
$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$



2 Qubits Gates

- Single qubit gates are not sufficient to generate entanglement between quantum states!
  - We need at least 2-qubit gates which live in $\mathbb{C}^{4\times4}$
- Controlled Not (CNOT) or CX is one of the most common
  - If control qubit is 0 then I is applied to target qubit
  - If control qubit is 1 then X is applied to target qubit



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

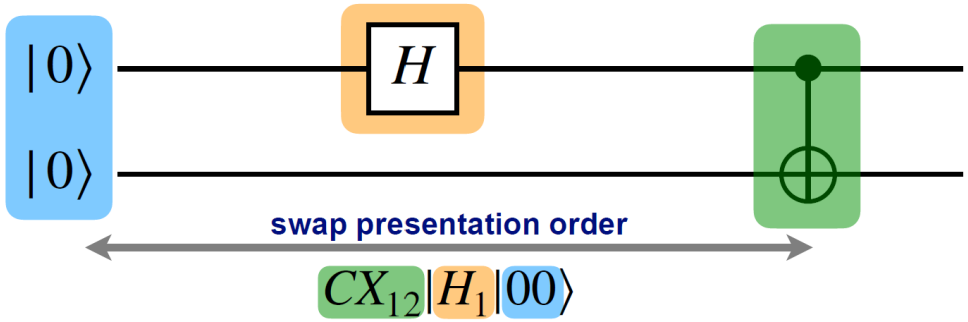
$$CX_{12}|00\rangle = |00\rangle \quad CX_{12}|01\rangle = |01\rangle$$
$$CX_{12}|10\rangle = |11\rangle \quad CX_{12}|11\rangle = |10\rangle$$

Obtaining entanglement

- Let's prepare a Bell state using quantum gates from the $|00\rangle$ state

$$|\psi\rangle = \begin{matrix}|00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle\end{matrix}\begin{pmatrix}\sqrt{0.5} \\ 0 \\ 0 \\ \sqrt{0.5}\end{pmatrix} = \sqrt{0.5}|00\rangle + \sqrt{0.5}|11\rangle$$

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

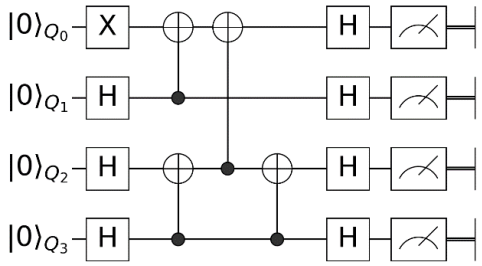


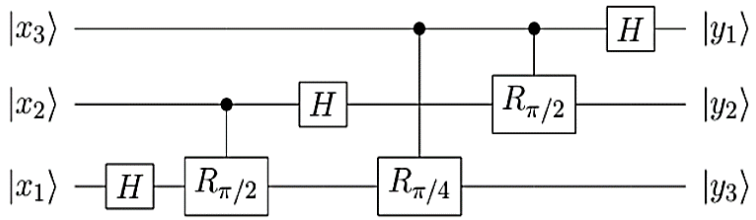
$$|\psi\rangle = \begin{pmatrix}1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0\end{pmatrix}\begin{pmatrix}1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} & 0 & -1/\sqrt{2}\end{pmatrix}\begin{pmatrix}1 \\ 0 \\ 0 \\ 0\end{pmatrix}\begin{matrix}|00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle\end{matrix}$$

$H \otimes I$

Quantum Computer program

- Examples of quantum programs





- In practice, one can usually not directly implement one single big unitary U
  - How to write a matrix of size $\mathbb{C}^{2^q \times 2^q}$?
  - Quantum Computers implement a limited set of gates (usually 1 and 2 qubit gates)

Common Single Qubit 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	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$

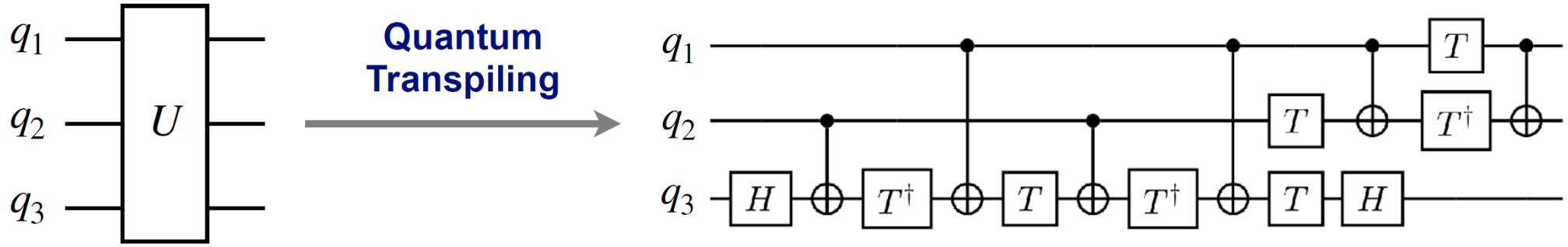
Common Two-Qubit Gate

Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

1. https://en.wikipedia.org/wiki/Quantum_computing
2. https://en.wikipedia.org/wiki/Quantum_logic_gate

Quantum Computer program

- There is a need for transpiling any desired unitary into gates supported by the hardware



Clifford + T

- Is this enough? Yes! One can have a universal gate-set able to represent any unitary
  - There are many and commercial quantum gate-based computers implement at least one

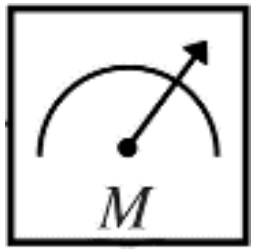
Hadamard (H)	H	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
Phase (S, P)	S	$\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}$
$\pi/8$ (T)	T	$\begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$

1. https://en.wikipedia.org/wiki/Quantum_logic_gate#Universal_quantum_gates

Overview of Quantum Mechanics: Measurement

Measurement of a quantum state

For the purpose of quantum computing, the measurement operation is well defined in the processor, and its effect is taken as a postulate (*the Born rule*):



$$|\psi\rangle_{\text{qubit}} = \psi_0|0\rangle + \psi_1|1\rangle$$

Measurement

Probability to measure 0 $= |\psi_0|^2$

Probability to measure 1 $= |\psi_1|^2$

Final state is $|\psi\rangle_{\text{qubit}} = |0\rangle$

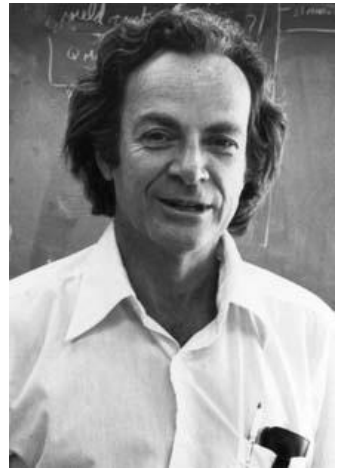
Final state is $|\psi\rangle_{\text{qubit}} = |1\rangle$

$$|\psi_0|^2 + |\psi_1|^2 = 1$$

$$|\psi\rangle_{2\text{qubit}} = \psi_{00}|00\rangle + \psi_{01}|01\rangle + \psi_{10}|10\rangle + \psi_{11}|11\rangle$$

Probability of measuring 11 is $|\psi_{11}|^2$

From coherent superposition to collapse



If you think you understand quantum mechanics, you don't understand quantum mechanics.

Measurement of a quantum state

- When you “read” a qubit you get a basis state, 0 or 1, (not the $\mathbb{C}^2$ state vector!)
- The amplitudes of the Quantum State ($|\psi\rangle$) determine the probability distribution of the bits you will observe

$$p_s = \text{diagonal}(|\psi\rangle\langle\psi|)$$

$$|\psi\rangle = \begin{matrix}|0\rangle\\|1\rangle\end{matrix}\begin{pmatrix}\sqrt{0.5}\\\sqrt{0.5}\end{pmatrix}, p_s = \begin{matrix}|0\rangle\\|1\rangle\end{matrix}\begin{pmatrix}\left|\sqrt{0.5}\right|^2\\\left|\sqrt{0.5}\right|^2\end{pmatrix} = \begin{matrix}|0\rangle\\|1\rangle\end{matrix}\begin{pmatrix}0.5\\0.5\end{pmatrix}$$

$$|\psi\rangle = \begin{matrix}|00\rangle\\|01\rangle\\|10\rangle\\|11\rangle\end{matrix}\begin{pmatrix}\sqrt{0.5}\\0\\0\\\sqrt{0.5}\end{pmatrix}, p_s = \begin{matrix}|00\rangle\\|01\rangle\\|10\rangle\\|11\rangle\end{matrix}\begin{pmatrix}0.5\\0\\0\\0.5\end{pmatrix}$$

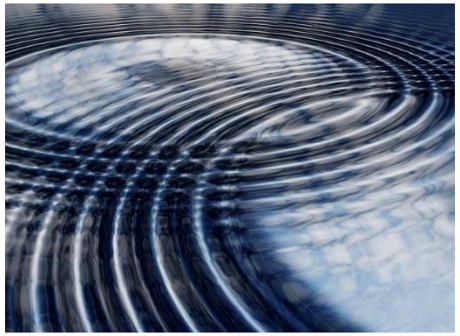
$$|\psi\rangle_{2\ qubits} =_{eg} \begin{matrix}|00\rangle\\|01\rangle\\|10\rangle\\|11\rangle\end{matrix}\begin{pmatrix}0.25+0.50i|00\rangle\\0.25-0.25i|01\rangle\\0.50+0.50i|10\rangle\\0.00-0.25i|11\rangle\end{pmatrix}, p_s = \begin{matrix}|00\rangle\\|01\rangle\\|10\rangle\\|11\rangle\end{matrix}\begin{pmatrix}|0.25|^2+|0.50|^2\\|0.25|^2+|-0.25|^2\\|0.50|^2+|0.50|^2\\|0.00|^2+|-0.25|^2\end{pmatrix} = \begin{matrix}|00\rangle\\|01\rangle\\|10\rangle\\|11\rangle\end{matrix}\begin{pmatrix}0.3125\\0.1250\\0.5000\\0.0625\end{pmatrix}$$

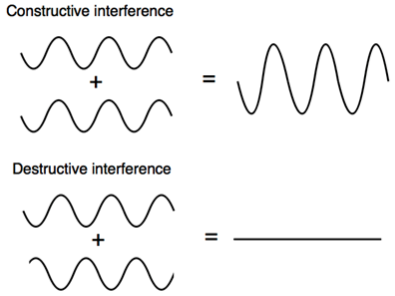
The goal of a quantum algorithm is to concentrate as much probability in the “right” basis vector (e.g., the solution of a combinatorial optimization problem)

Overview of Quantum Mechanics: Interference

Interference in probability amplitudes

- In quantum mechanics, quanta exhibit both wave-like and particle-like properties.
- When two or more quantum states are in superposition, their probability amplitudes can interfere constructively or destructively.

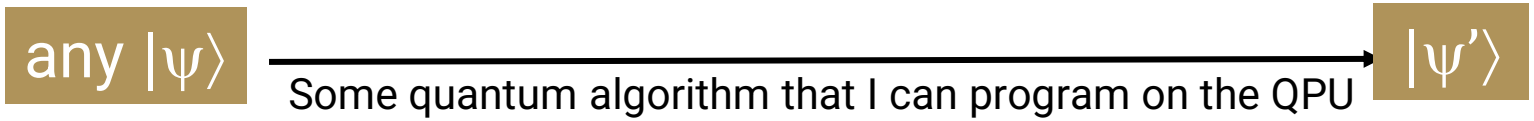




“A quantum computer is a device that exploits constructive and destructive interference among exponentially many amplitudes, which are numbers that are closely related to probabilities but can be positive, negative, or even complex.”
- Scott Aaronson

Universal Quantum Computing

- **A quantum computer is UNIVERSAL if its instruction set allows the implementation of any algorithm allowed by quantum mechanics.**



- **Why you might want a Univesal Quantum Computer?**

(1) Simulation of Quantum Systems

(2) Flexibility of implementation of multiple quantum algorithms

(e.g. Grover and Shor)

(3) Exploit all the power of quantum mechanics

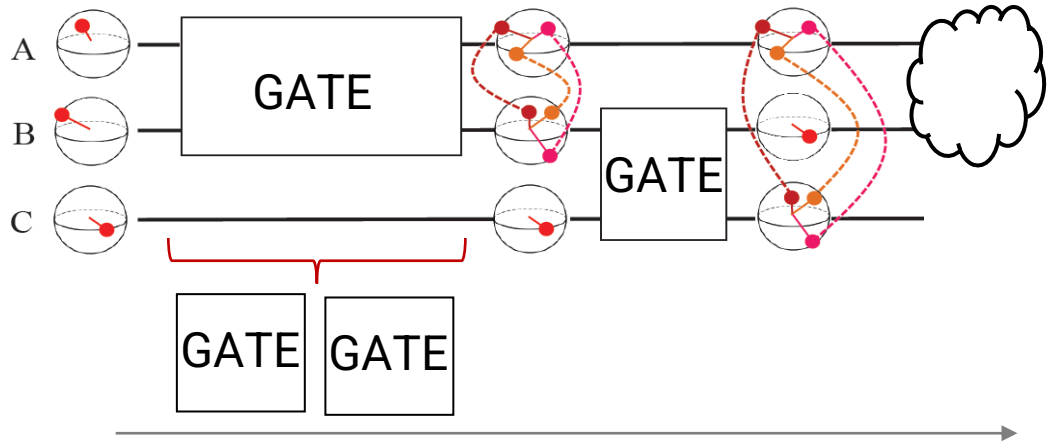
(4) Making sure that what you do is not classically simulatable efficiently

For quantum advantage in optimization heuristics, universality is not necessarily required (the final state we are searching is classical).

The time-evolution of the Ising model in a transverse field (Quantum Annealing as implemented in D-Wave) is **NOT** universal. However the general AQC procedure is universal (need more complex Problem and Driver Hamiltonians).

Gate-Model and Quantum Circuits

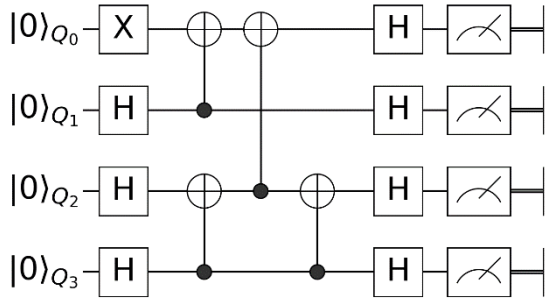
- The gate-model is a simple way to break down the quantum coherent operations we use in quantum computing.

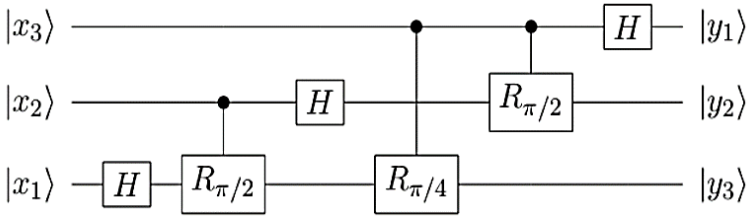


$$U = U_{BC}U_{AB}|ABC\rangle$$

$$U_{AB}|ABC\rangle = \psi_{00}|00C\rangle + \psi_{01}|01C\rangle + \psi_{10}|10C\rangle + \psi_{11}|11C\rangle$$

$$\begin{aligned} U_{BC}U_{AB}|ABC\rangle &= \psi_{00}U_{BC}|00C\rangle + \psi_{01}U_{BC}|01C\rangle \\ &\quad + \psi_{10}U_{BC}|10C\rangle + \psi_{11}U_{BC}|11C\rangle \\ &= \psi_{000}|000\rangle + \psi_{001}|001\rangle + \psi_{010}|010\rangle \\ &\quad + \psi_{011}|011\rangle + \psi_{100}|100\rangle + \psi_{101}|101\rangle \\ &\quad + \psi_{110}|110\rangle + \psi_{111}|111\rangle \end{aligned}$$

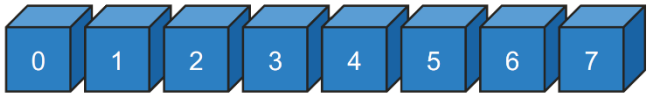




Reminder: every operation on a N qubit system is mathematically equivalent to multiplying a unitary matrix in $\mathbb{C}^{2^q \times 2^q}$ to a normalized vector.
= you cannot keep track numerically of the amplitudes of large circuits.

The Grover Algorithm

- Search over an unstructured database



Classically, worst case is querying all elements

Given:

- Power-of-2 boxes, guarantee that solution (marked) is only in one place
- An operator that negates the probability of the solution state

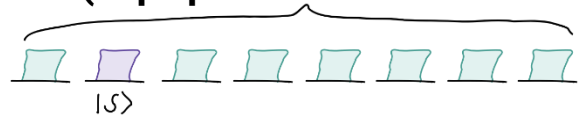
$U_\omega|\mathrm{x}\rangle = -|\mathrm{x}\rangle$ if $x = \omega$, $U_\omega|\mathrm{x}\rangle = |\mathrm{x}\rangle$ otherwise

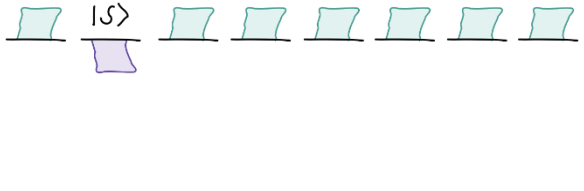
Define Grover diffusion operator

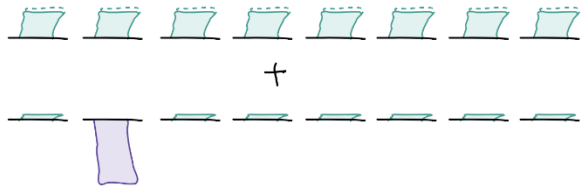
- $U_s = 2|s\rangle\langle s| - I$ with $|s\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|x\rangle$ (equal superposition)

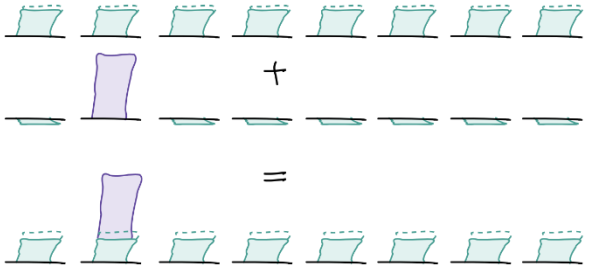
Start with complete superposition (equiprobable states)

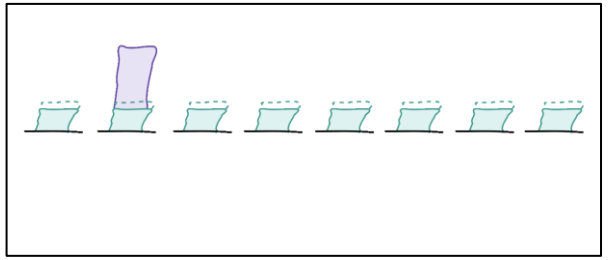
Apply for $\sqrt{N}$ rounds U_ω and U_s

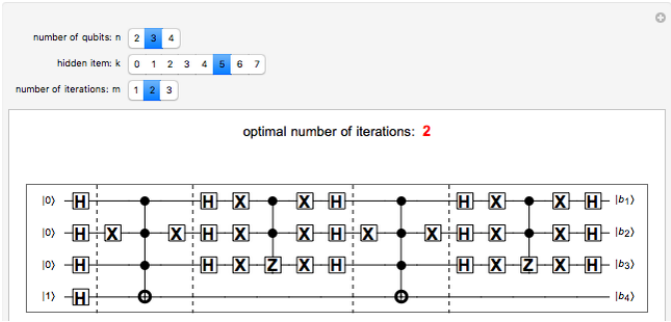


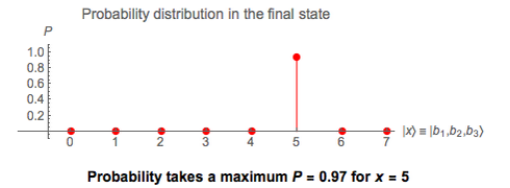






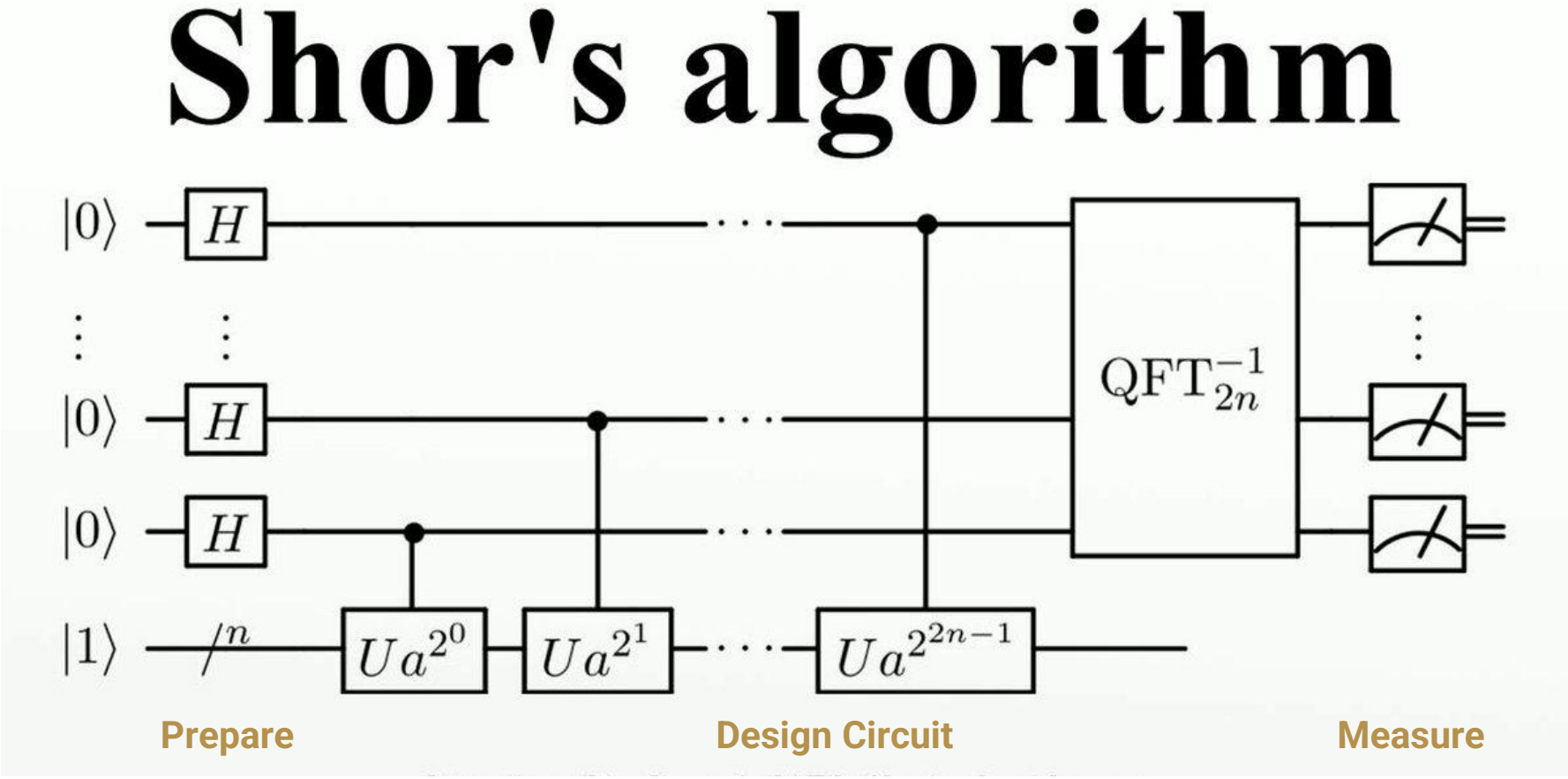






1. https://codebook.xanadu.ai/G.1
2. https://demonstrations.wolfram.com/QuantumCircuitImplementingGroversSearchAlgorithm/

Quantum Algorithms



Overview of Quantum Mechanics: Quantum Incoherent Operation

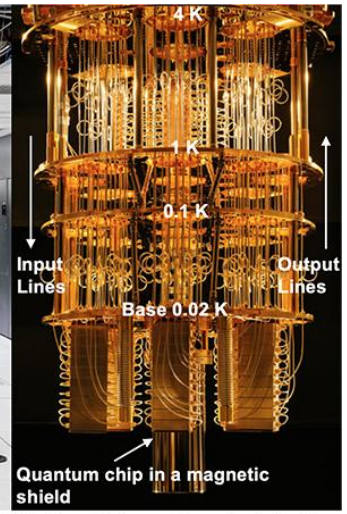
Incoherent Operations (Noise)

Final State after your algorithm if everything was coherent

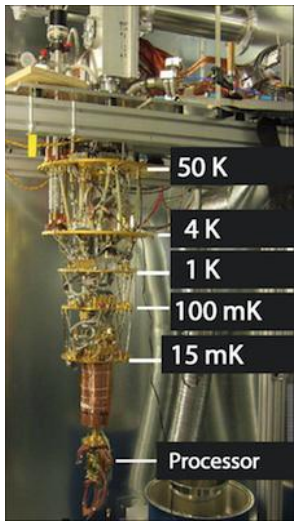
$$|\psi\rangle_{\text{q qubits}} = \sum_{n=1}^{2^q} \psi_n |\text{solution(n)}\rangle_n$$

- A random guess algorithm has $|\psi_{n=\text{target}}|^2 \approx 1/\sqrt{2^N}$
- A good algorithm has $|\psi_{n=\text{target}}|^2 \gg 1/\sqrt{2^N}$

Dilution fridge setup: outside view



Dilution fridge setup: inside view



Source from IBM and D-Wave

BUT....

We are currently using Noisy-Intermediate-Scale QPUs (NISQ)

$$|\psi^I(\Delta t)\rangle = T \exp\left[\int_0^{\Delta t} H^I(t)\, dt' \right] |\psi^I(0)\rangle$$

The Shroedinger equation applies only approximately and for a limited time

There is interaction with the environment which is always on and it is causing changes to the operations:

→ some measurements happen at random due to the environment and destroy «partially» the wavefunction (decoherence)

→ Some control parameters are imprecise and $y_n \approx y_n + d$

→ Theory of modeling of noise is very phenomenological

Noisy Intermediate-Scale Quantum era

Decoherence and Noise:

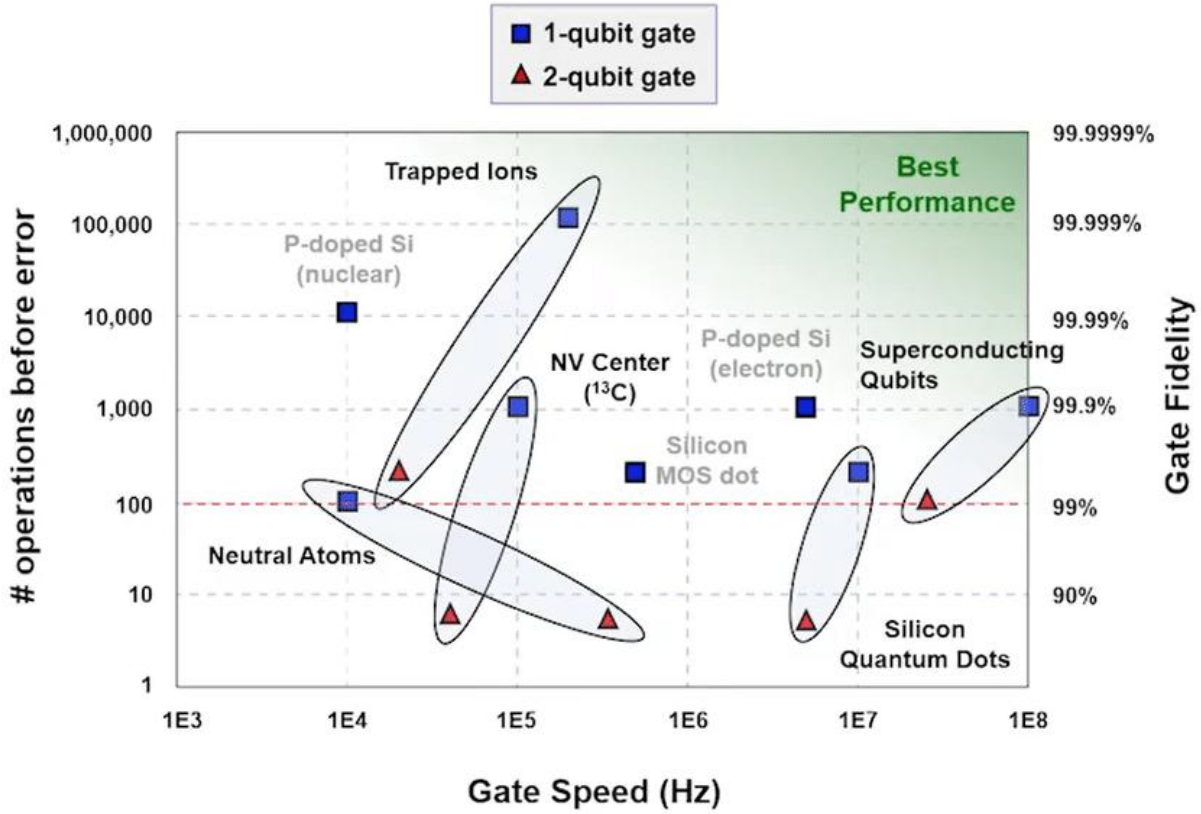
- Quantum information is sensitive to its environment, leading to decoherence and noise that can cause errors.

Error Rates:

- The high error rates in NISQ devices pose challenges for performing reliable and accurate quantum computations.

Limited Gate Fidelity:

- The fidelity of quantum gates may be limited, impacting the overall performance of quantum circuits.



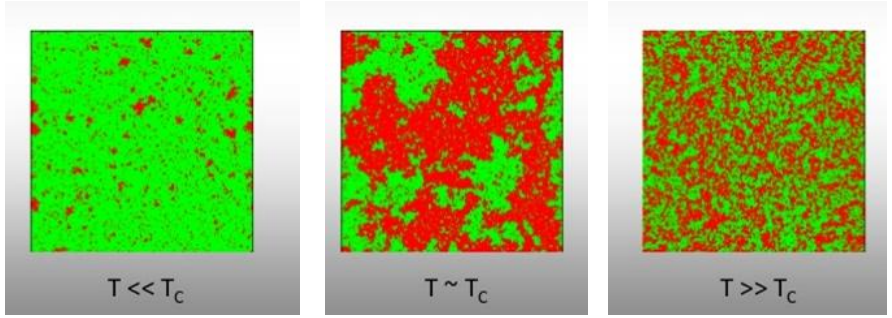
Quadratic Unconstrained Binary Optimization and Ising Model: History

What is an Ising problem? Why is it relevant here?

Discovery by Curiosity

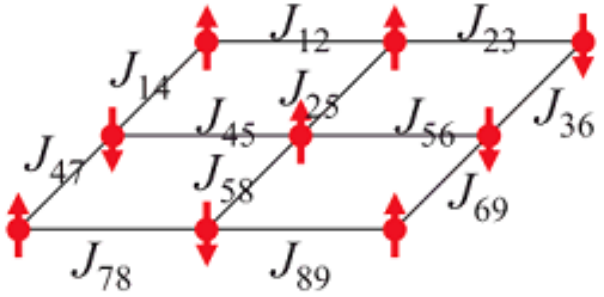
In 1895, Pierre Curie (Nobel Prize 1903) found that heating a magnet can cause it to lose its magnetism, i.e., cause a "phase transition".

- But **Why? Nature prefers the least energy states.**
  - **Why such a sudden change?**



Model

1920 - Lenz presents model for energy states of discrete particles
1925 - Lenz's student, Ising, solved a special 1-D case of the model
1940 - Onsager (*Nobel Prize 1968*) solved the 2-D case
2000 - Istrail shows that 3-D case is NP-Hard



$$\min_{\sigma\in\{-1,+1\}^n} \sum_{(ij)\in E(G)} J_{ij}\sigma_i\sigma_j + \sum_{i\in V(G)} h_i\sigma_i =$$
$$\min_{\mathbf{x}\in\{0,1\}^n} \sum_{(ij)\in E(G)} x_i Q_{ij} x_j + \sum_{i\in V(G)} Q_{ii}x_i + c$$

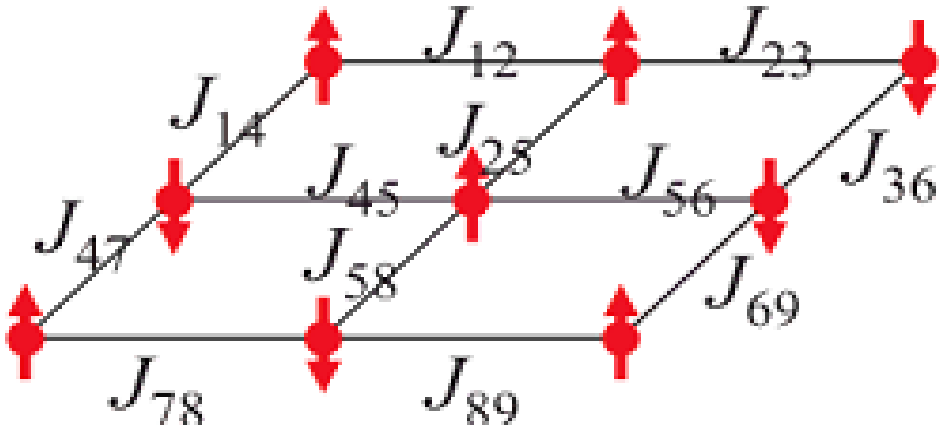
Quadratic unconstrained binary opt (QUBO) is a discrete nonlinear opt subcase

General lesson

1971 - Wilson (*Nobel Prize 1982*), **universality**:
systems with the same number of dimensions and symmetries go through the same phase transitions

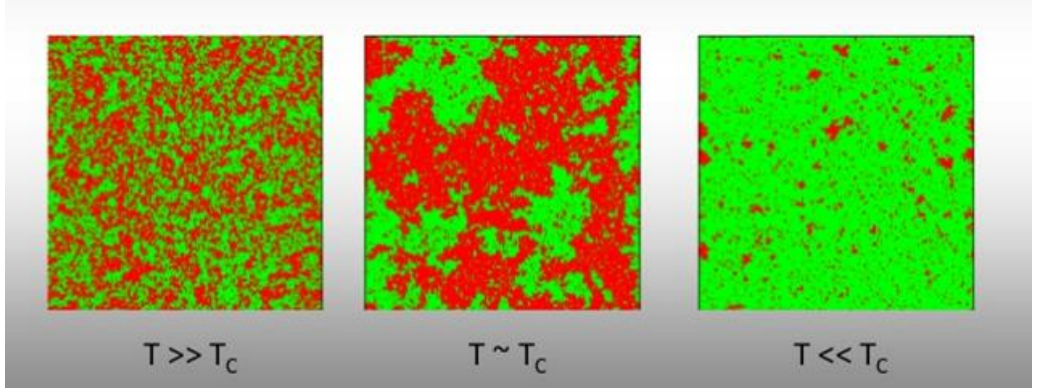
The Ising model is the simplest model in theory space that captures properties of several interacting systems like magnets, water etc.

Ising Model Magnetic Phenomenology



$$H = -\sum_{\langle i,j \rangle} J_{i,j}\,\sigma_i\sigma_j - \sum_j h_j\sigma_j$$

Spin are subject to thermal fluctuations and can be up or down

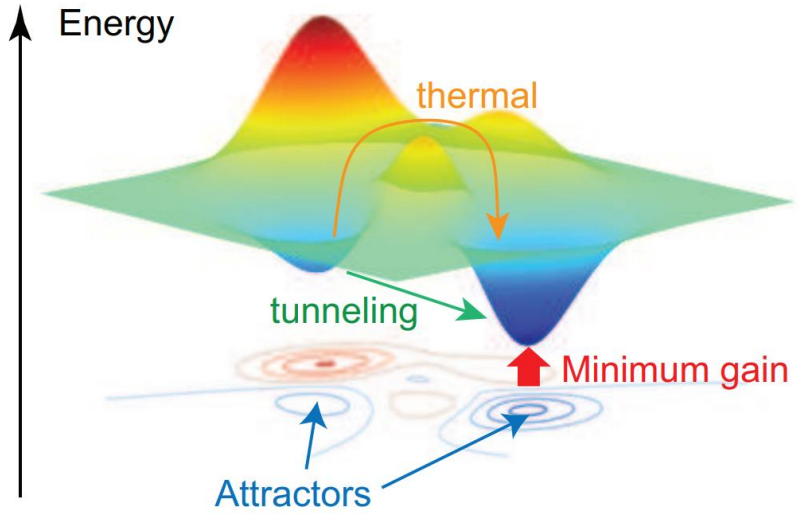


Depending on the temperature the fluctuations have macroscopic properties of symmetry (“thermodynamic phases”) which are encapsulated in the value of an order parameter (“magnetization”)

Quadratic Unconstrained Binary Optimization and Ising Model: Solvers

Solving an ISING/QUBO problem

What is it? How?

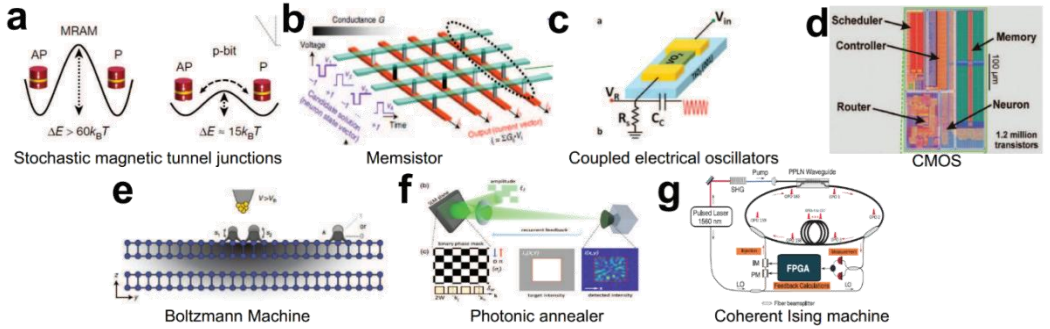


"Solving the Model"

- **Physicists:** identifying thermodynamics expectation values (find the distribution of solutions/energies)
- **Optimizers:** find the actual variable values that minimizes the objective function (find the ground state)

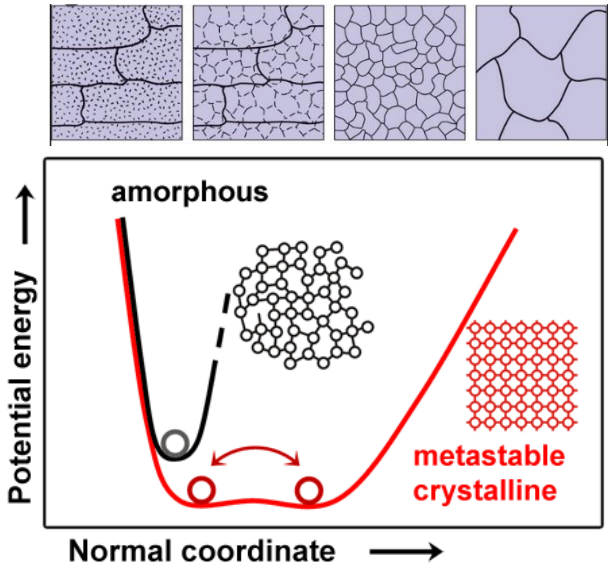
How to solve the Ising model? With an Ising solver!

- **Ising solver?** End-to-end solution method including software/algorithms and hardware/machines
  - **Classical Thermal Annealing**
  - Other classical algorithms
    - MIP or Physics-inspired classical algorithms
  - Hardware solution-based methods
    - **Coherent Ising Machines,** GPU, CMOS, …
  - Quantum Approaches
    - **Quantum Annealing**
    - **Hybrid Quantum-Classical Algorithms**
      - **Variational Quantum Algorithms – QAOA, VQE**



1. Mohseni, Naeimeh, Peter L. McMahon, and Tim Byrnes. "Ising machines as hardware solvers of combinatorial optimization problems." Nature Reviews Physics 4.6 (2022): 363-379.

Simulated Annealing



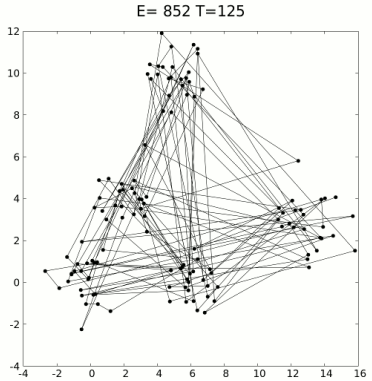
Concept coming from annealing in metallurgy

Slow cooling allows for perfect crystals (minimizing energy)

Start at effective high temperature and gradually decrease the temperature by increments until is slightly above zero

Interesting behavior:

- “Divide-and-conquer”: Big features are solved early in the search and small features later while refining
- Ability to escape local-minima
- Guaranteed to reach lowest energy if temperature is lowered slowly enough



Ising Model – Monte Carlo, Physics Inspired Methods

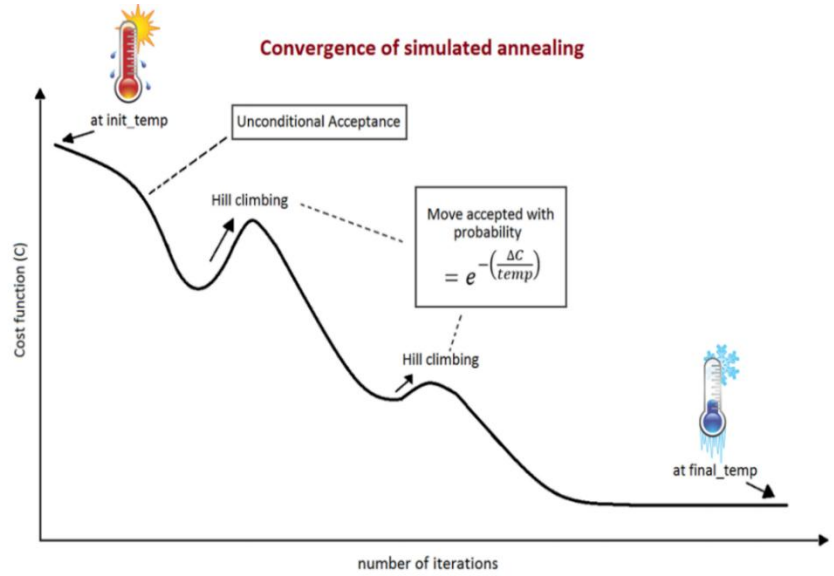
Ising model as Markov-Chain

The immediate probability $P(\boldsymbol{\sigma}^c, \beta) = e^{-\beta H(\boldsymbol{\sigma}^c)}/Z(\beta)$ of transitioning to a future state $\boldsymbol{\sigma}^f$ depends only in the current state $\boldsymbol{\sigma}^c = [\sigma_1^c, \cdots, \sigma_N^c]$

Given single flip dynamics, we can jump from any state to another.

Metropolis-Hastings Monte Carlo Algorithm for Ising Models

1) Start with a known configuration, $\boldsymbol{\sigma}^i = [\sigma_1^i, \cdots, \sigma_N^i]$ corresponding energy, $H(\boldsymbol{\sigma}^i)$ and temperature value $T = (k_B\beta)^{-1}$
2) Randomly change the configuration
   - Flip some spins $\boldsymbol{\sigma}^i \rightarrow \boldsymbol{\sigma}^j$
3) Calculate new energy value $H(\boldsymbol{\sigma}^j)$
4) Compare to energy at previous position
   - If , $H(\boldsymbol{\sigma}^j) < H(\boldsymbol{\sigma}^i)$ keep new position
   - If , $H(\boldsymbol{\sigma}^j) > H(\boldsymbol{\sigma}^i)$ keep new position if Boltzmann factor for transition satisfies $\exp\left[-\frac{H(\boldsymbol{\sigma}^j)-H(\boldsymbol{\sigma}^i)}{k_B T}\right] \geq \text{Rand}[0,1]$
5) Repeat 2) - 4) K times

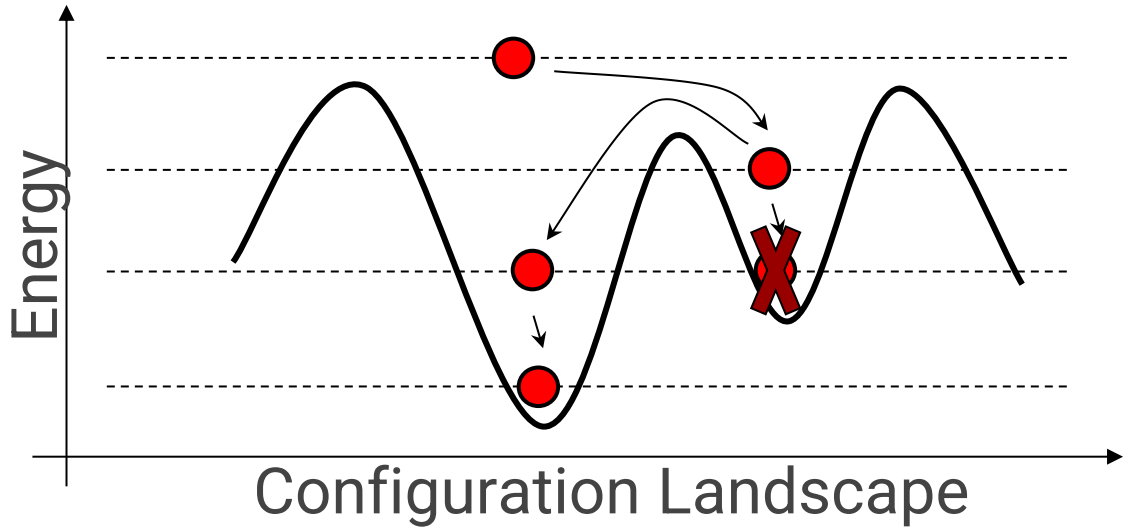


[1] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing.Science,220(4598):671–680, 1983.

Simulated Annealing - Details

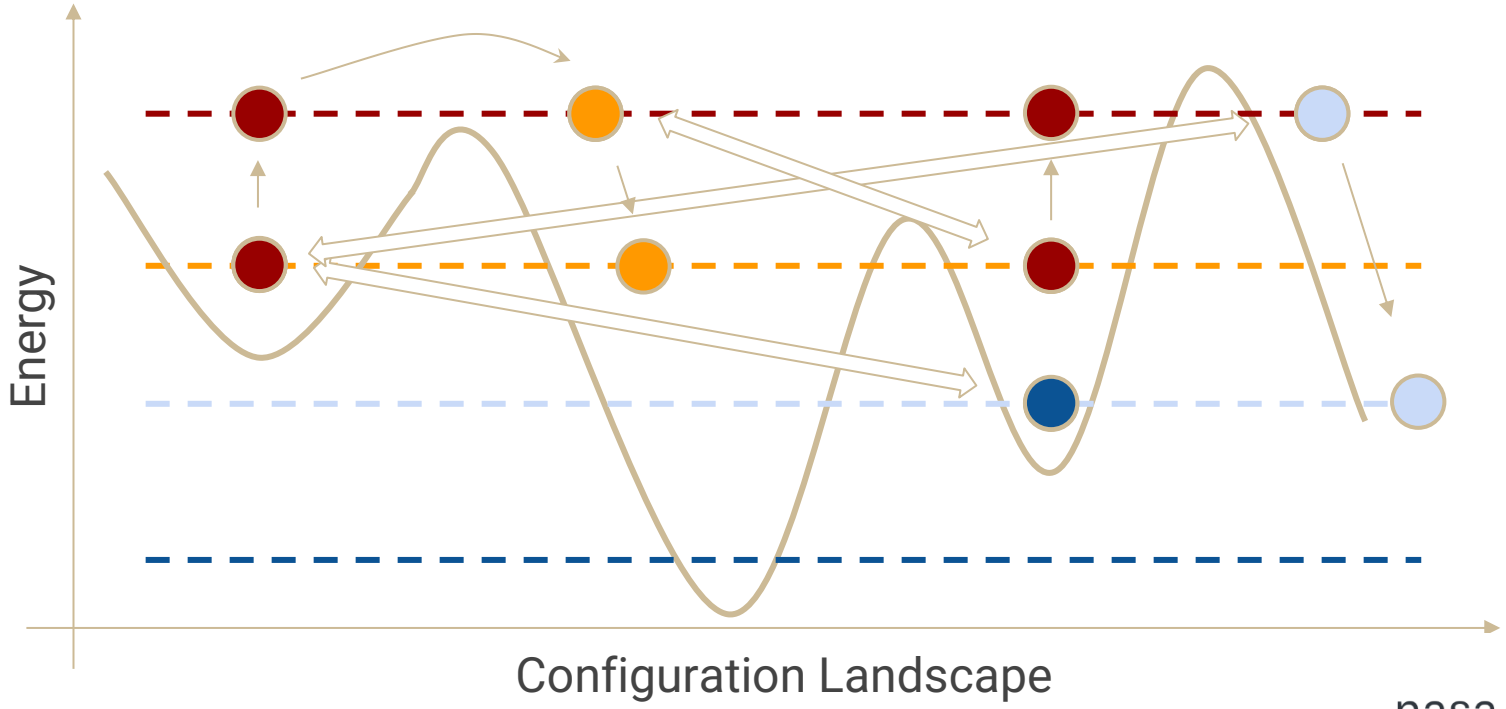
- For a sufficiently **large number of steps**, the spin system will reach "**thermalization**" and therefore, configurations are sampled accordingly to the **Gibbs distribution**.
- Starting from **high temperature** and then **slowly reducing it**, the system will **eventually thermalize** at any given temperature and therefore, **sample the ground state**.

Using the Metropolis-Hasting algorithm to compute the transition probability $p(\sigma'|\sigma)$, **preserving both the detailed balance and ergodicity of the system,** would eventually **drive the system** to a given stationary distribution $p(\sigma)$.



[1]Mandra, S. Lecture on Ising Models, 2021.

Advaced Simulated Annealing – Parallel Tempering



nasa/**PySA**

4 Contributors · 1 Issue · 23 Stars · 5 Forks

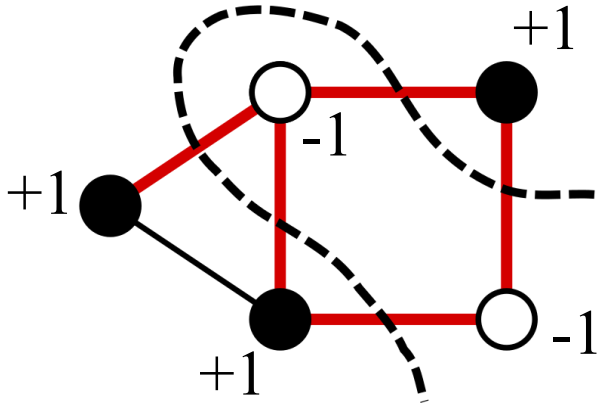
$$A_{\text{exchange}} = \min\{1, e^{\Delta\beta\Delta\mathcal{H}}\}$$

Quadratic Unconstrained Binary Optimization and Ising Model: Equivalences

Ising for Combinatorial Optimization: MaxCut

Starting from Ising Problem without external field

$$H(\sigma) = -\sum_{(ij)\in E(G)} J_{ij}\sigma_i\sigma_j$$
$$= -\sum_{(ij)\in E(V^+)} J_{ij} - \sum_{(ij)\in E(V^-)} J_{ij} + \sum_{(ij)\in \delta(V^+)} J_{ij}$$
$$= -\sum_{(ij)\in E(G)} J_{ij} + 2\sum_{(ij)\in \delta(V^+)} J_{ij}$$



where the set V^+ (V^-) are all the vertices with $\sigma = +1$ ($\sigma = -1$) and their boundary (cut) is denoted by $\delta(V^+)$

Now consider that the graph has weighted edges W_{ij}

Then the size of the cut is $|\delta(V^+)| = \frac{1}{2}\sum_{(i,j)\in\delta(V^+)} W_{ij}$

Therefore we obtain $H(\sigma) = \sum_{(ij)\in E(G)} W_{ij} - 4|\delta(V^+)|$

When minimizing the Ising model, we are finding the maximum cut of the graph

$$\min_\sigma H(\sigma) = \sum_{(ij)\in E(G)} W_{ij} + 4\max_\sigma |\delta(V^+)|$$

Optimization value of the choice made for the spins

Constant of the problem, easy to compute

[1]https://www.electronics-tutorials.ws/electromagnetism/magnetism.html
[2]http://www.irm.umn.edu/hg2m/hg2m_b/hg2m_b.html

From Ising to Quadratic Binary Unconstrained (QUBO)

Starting from the minimization of the Ising Model

$$\min_{\sigma\in\{-1,+1\}^n} H(\sigma) = \min_{\sigma\in\{-1,+1\}^n} \sum_{(ij)\in E(G)} J_{ij}\sigma_i\sigma_j + \sum_{i\in V(G)} h_i\sigma_i$$

We can directly pose this problem as a Quadratic Unconstrained Binary Optimization (QUBO).

$$\min_{\sigma\in\{-1,+1\}^n} \sum_{(ij)\in E(G)} J_{ij}\sigma_i\sigma_j + \sum_{i\in V(G)} h_i\sigma_i =$$
$$\min_{\mathbf{x}\in\{0,1\}^n} \sum_{(ij)\in E(G)} x_i Q_{ij} x_j + \sum_{i\in V(G)} Q_{ii}x_i + c$$

EXERCISE: put this in ILP form introducing the variable $x_{ij} = x_i x_j$

$$\sigma_i = 2x_i - 1 \qquad x_i = (\sigma_i+1)/2$$
$$\sigma_i\sigma_j = 4x_ix_j - 2x_i - 2x_j + 1 \iff x_ix_j = (\sigma_i\sigma_j + \sigma_i + \sigma_j + 1)/4$$

With $Q_{ij} = 4J_{ij}, Q_{ii} = 2h_i - \sum_{j\in V(G)}(2J_{ij} + 2J_{ji}), c = \sum_{i<j} J_{ij} - \sum_{i\in V(G)} h_i$

$$J_{ij} = Q_{ij}/4, h_i = Q_{ii}/2 + \sum_{j\in V(G)}(Q_{ij}/4 + Q_{ji}/4), c_I = c_Q + \sum_{i<j} Q_{ij}/4 - \sum_{i\in V(G)} Q_{ii}/2$$

Quadratic Unconstrained Binary Optimization model

Quadratic Unconstrained Binary Optimization

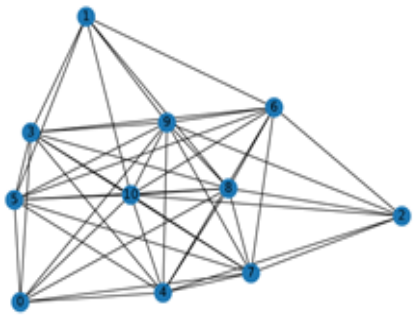
$$\min_{\mathbf{x}\in\{0,1\}^n} \sum_{(ij)\in E(G)} x_i Q_{ij} x_j + \sum_{i\in V(G)} Q_{ii} x_i + c$$
$$= \min_{\mathbf{x}\in\{0,1\}^n} \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + c$$

Offset $\boldsymbol{c}$
- Irrelevant for optimization

Quadratic coefficient matrix $\mathbf{Q}$
- Can be either upper triangular or complete $x_i x_j = x_j x_i$
- Elements on diagonal can be linearized $x_i^2 = x_i$ if $x_i \in \{0, 1\}$
- Represents adjacency matrix of a problem

In terms of IP it would be a
Non-convex Integer Nonlinear Program

$$\begin{bmatrix}
-46. & 0. & 0. & 48. & 48. & 48. & 0. & 48. & 48. & 48. & 48. \\
0. & -44. & 0. & 48. & 0. & 48. & 48. & 0. & 48. & 48. & 48. \\
0. & 0. & -44. & 0. & 48. & 0. & 48. & 48. & 48. & 48. & 48. \\
48. & 48. & 0. & -92. & 48. & 96. & 48. & 48. & 96. & 96. & 96. \\
48. & 0. & 48. & 48. & -92. & 48. & 48. & 96. & 96. & 96. & 96. \\
48. & 48. & 0. & 96. & 48. & -92. & 48. & 48. & 96. & 96. & 96. \\
0. & 48. & 48. & 48. & 48. & 48. & -91. & 48. & 96. & 96. & 96. \\
48. & 0. & 48. & 48. & 96. & 48. & 48. & -92. & 96. & 96. & 96. \\
48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & -139. & 144. & 144. \\
48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & 144. & -138. & 144. \\
48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & 144. & 144. & -139.
\end{bmatrix}$$



Quadratic Unconstrained Binary Optimization and Ising Model: Mapping constrained Optimization

QUBO Mappings Examples: Linear Equality Constraints

Example: Binary Linear Programming

$$\min_{\mathbf{x}\in\{0,1\}^n} \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + c$$
$$s.t.\ \mathbf{A}\mathbf{x} = \mathbf{b}$$

Hard Constraint

$\min(H) \Longleftrightarrow H_A = 0$

Objective Function to be optimized

We can write the energy function in two terms $H = H_A + H_B$

- Constraint satisfaction $(H_A = 0)$

$$H_A = \rho \sum_{j=1}^{m} \left(\sum_{i=1}^{n} \left(A_{ij}x_i - b_j\right)^2\right)$$

Penalty Coefficient

- Objective function

$$H_B = \sum_{i=1}^{n} c_i x_i$$

How to determine the penalty? Guarantee that the minimal >0 value for H_A is larger than the gap between the ground state of H_B and the minimum energy that satisfies the constraint (unknown – but can be bound)

Note: This is known as augmented Lagrangian (see 1st lecture)

[1] Bian, Zhengbing, et al. "Discrete optimization using quantum annealing on sparse Ising models." Frontiers in Physics 2 (2014): 56.
[2] Lucas, Andrew. "Ising formulations of many NP problems." Frontiers in Physics 2 (2014): 5.

QUBO Mappings Examples: Polynomial Unconstrained Binary

Any pseudo-Boolean function defined as

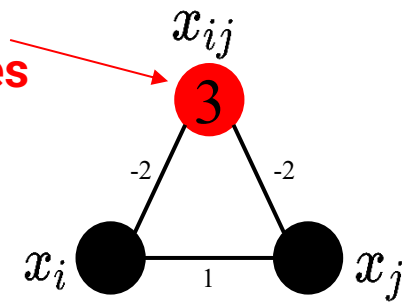
$$f: \{0,1\}^n \longmapsto \mathbb{R}$$

Can be written uniquely as a sum of multilinear functions

$$x_{ij} = x_i x_j \qquad f(x) = a_0 + \sum_i a_i x_i + \sum_{ij} a_{ij} x_i x_j + \sum_{ijk} a_{ijk} x_i x_j x_k + \cdots$$

We can transform any binary polynomial into a quadratic polynomial by introducing new variables as we

Usually called ancillary variables



Naive example:

- Using linear inequalities

  $x_{ij} \geq x_i + x_j - 1, x_{ij} \leq x_i, x_{ij} \leq x_j$

- Using a polynomial expression as penalty (gadget)

  $H(\mathbf{x}) = 3x_{ij} + x_i x_j - 2x_{ij} x_i - 2x_{ij} x_j$

x_i	x_j	x_{ij}	$x_i x_j$	$H(x)$
1	1	1	1	0
1	1	0	1	1
1	0	1	0	1
1	0	0	0	0
0	1	1	0	1
0	1	0	0	0
0	0	1	0	3
0	0	0	0	0

[1] Babbush, Ryan, Bryan O'Gorman, and Alán Aspuru-Guzik. "Resource efficient gadgets for compiling adiabatic quantum optimization problems." Annalen der Physik 525.10-11 (2013): 877-888.
[2] Boros, Endre, and Peter L. Hammer. "Pseudo-boolean optimization." Discrete applied mathematics 123.1-3 (2002): 155-225.

QUBO Mappings Examples: From Integers to Bits

How to transform general Integer Programs into Binary programs?

$$y \in \{0, \cdots, \bar{y}\} \subseteq \mathbb{Z}$$

In general, with a width of the integer encoding $\boldsymbol{d}$

$$y = \sum_{j=1}^{d} k_j x_j = \mathbf{k}^{\mathrm{T}}\mathbf{x}, k_j \in \mathbb{Z}_+, x_j \in \{0,1\}$$

- Unary encoding $k_j = 1, d = \bar{y}$
- Binary encoding $k_j = 2^{j-1}, d = \lfloor \log_2(\bar{y}) \rfloor$
- Bounded encoding

We can find an encoding with an upper bound for the coefficients $\mu \ll \bar{y}$

$$\text{if } \bar{y} < 2^{\lfloor \log(\mu) \rfloor} \qquad \text{if } \bar{y} > 2^{\lfloor \log(\mu) \rfloor}$$

N bits for 1…N
Min precision req

$\log_2$(N) bits for 1…N
max precision req

[1] Karimi, Sahar, and Ronagh, Pooya. "Practical integer-to-binary mapping for quantum annealers." Quantum Information Processing 18.4 (2019): 94.

QUBO Mappings Examples: Linear Inequality Constraints

For linear equality constraints

$$s.t.\ \mathbf{Ax} = \mathbf{b} \longrightarrow H_A = \rho \sum_{j=1}^{m} \sum_{i=1}^{n} \left(A_{ij} x_i - b_j\right)^2$$

Inequality constraints:

$$\mathbf{Ax} \leq \mathbf{b}$$
$$\mathbf{Ax} - \mathbf{b} \leq \mathbf{0}$$
$$\mathbf{Ax} + \mathbf{s} - \mathbf{b} = \mathbf{0}$$

"slack variable"
Free variable that can take any value from zero to b

⟺

As long as $\mathrm{Ax} \leq \mathrm{b}$ there is one value of s that satisfies the equality constraint

Typical situation in mappings:
(e.g. knapsack problem)

$$\sum_i x_i \leq \mathrm{M}$$
$$\sum_i x_i + \sum_j m_j y_j = \mathrm{M}$$
$$H_A = \rho\left(\sum_i x_i + \sum_j m_j y_j - \mathrm{M}\right)^2$$

From integers to bits

QUBO.jl

A Julia Ecosystem for Ising and QUBO optimization


Pedro Maciel Xavier


Pedro Ripper


Tiago Andrade


Joaquim Dias Garcia


Nelson Maculan

Pedro Maciel Xavier[1,2], Pedro Ripper[1], Tiago Andrade[1], Joaquim Dias Garcia[1], Nelson Maculan[2], DEBN

[1]Energy Consulting and Analytics, PSR Inc.

[2]Department of Systems Engineering and Computer Science, UF Rio de Janeiro

https://github.com/JuliaQUBO/QUBO.jl

1. Xavier, Pedro Maciel, et al. "Qubo. jl: A julia ecosystem for quadratic unconstrained binary optimization." arXiv preprint arXiv:2307.02577 (2023).

QUBO.jl
What do our Julia packages do?

Table 2 List of supported variable encodings and constraint penalization methods across QUBO modelling platforms

	ToQUBO.jl	PyQUBO	qubovert	Qiskit	OpenQAOA	Amplify
Automatic variable encoding methods implemented						
Binary [1]	■ ★	■	■			■ !
Unary [1]	■ ★	■	■	■	■	■ !
One-Hot [1]	■ ★	■				
Domain-Wall [2]	■ ★	■				
Bounded-Coefficient [3]	■ ★	■				
Arithmetic Progression	■ ★					■ !
Supported constraints						
$\mathbf{a}'\mathbf{x} \leq b$	■		■	■	■	■
$\mathbf{a}'\mathbf{x} = b$	■		■	■	■	■
$\mathbf{x}'Q\,\mathbf{x} + \mathbf{a}'\mathbf{x} \leq b$	■					■
$\mathbf{x}'Q\,\mathbf{x} + \mathbf{a}'\mathbf{x} = b$	■					■
SOS1	■					
$\bigwedge_i x_i$ †			■			■
$\bigvee_i x_i$ †			■			■
$\bigoplus_i x_i$ †						■

[1] (Tamura et al. 2021); [2] (Chancellor 2019); [3] (Karimi and Ronagh 2019);

† For logical constraints, it is assumed that $x_i \in \mathbb{B}$.

★ `ToQUBO.jl` is the only platform in the board whose automatic encoding routines are also applicable to continuous variables. Furthermore, its *Bounded-Coefficient* implementation allows the technique to be used not just with the *Binary* encoding, as proposed in the original paper, but also with the *Unary* and *Arithmetic Progression* modes.

! `Amplify` only supports these encoding methods when introducing integer slack variables for mapping inequality constraints. They are not applicable to regular variables, as their models only accept binary or spin sites so far.

Table 3 External samplers coverage

Sampler	Hardware	QUBODrivers.jl	PyQUBO	qubovert	Qiskit	OpenQAOA	Amplify
Simulated Annealing	CPU/ GPU	■ [3]	■	■			■
Quantum Annealing	QA[1]	■ [4]	■		■ [9]		■
Simulated Quantum Annealing	CPU/ GPU	■ [5]					■
VQE	QGC [2]	■ [6]			■	■	
QAOA	QGC [2]	■ [6]			■	■	■
MQLib Heuristics Library	CPU	■ [7]					
Parallel Tempering	CPU	■ [8]					
Simulated Bifurcation Machine	GPU/FPGA						■
Digital Annealing	ASIC						■
CMOS Annealing	CMOS						■

[1] Quantum Annealer; [2] Quantum Gate Circuit; [3] `DWave.jl` (Xavier and Ripper 2022a); [4] `DWaveNeal.jl` (Xavier and Ripper 2023a); [5] `QuantumAnnealingInterface.jl` (Morrell and Coffrin 2022, Xavier and Ripper 2022b); [6] `QiskitOpt.jl` (Xavier and Ripper 2023c); [7] `MQLib.jl` (Dunning et al. 2018, Xavier and Ripper 2023b); [8] `PySA.jl` (Xavier 2023); [9] `Dwave-Qiskit-plugin` (D-Wave 2020);

1. Xavier, Pedro Maciel, et al. "Qubo. jl: A julia ecosystem for quadratic unconstrained binary optimization." arXiv preprint arXiv:2307.02577 (2023).

How do our Julia packages perform?

Outcome:
We provide the **fastest and most complete reformulator** of traditional optimization problems **to Ising/QUBO** as **an open-source** software

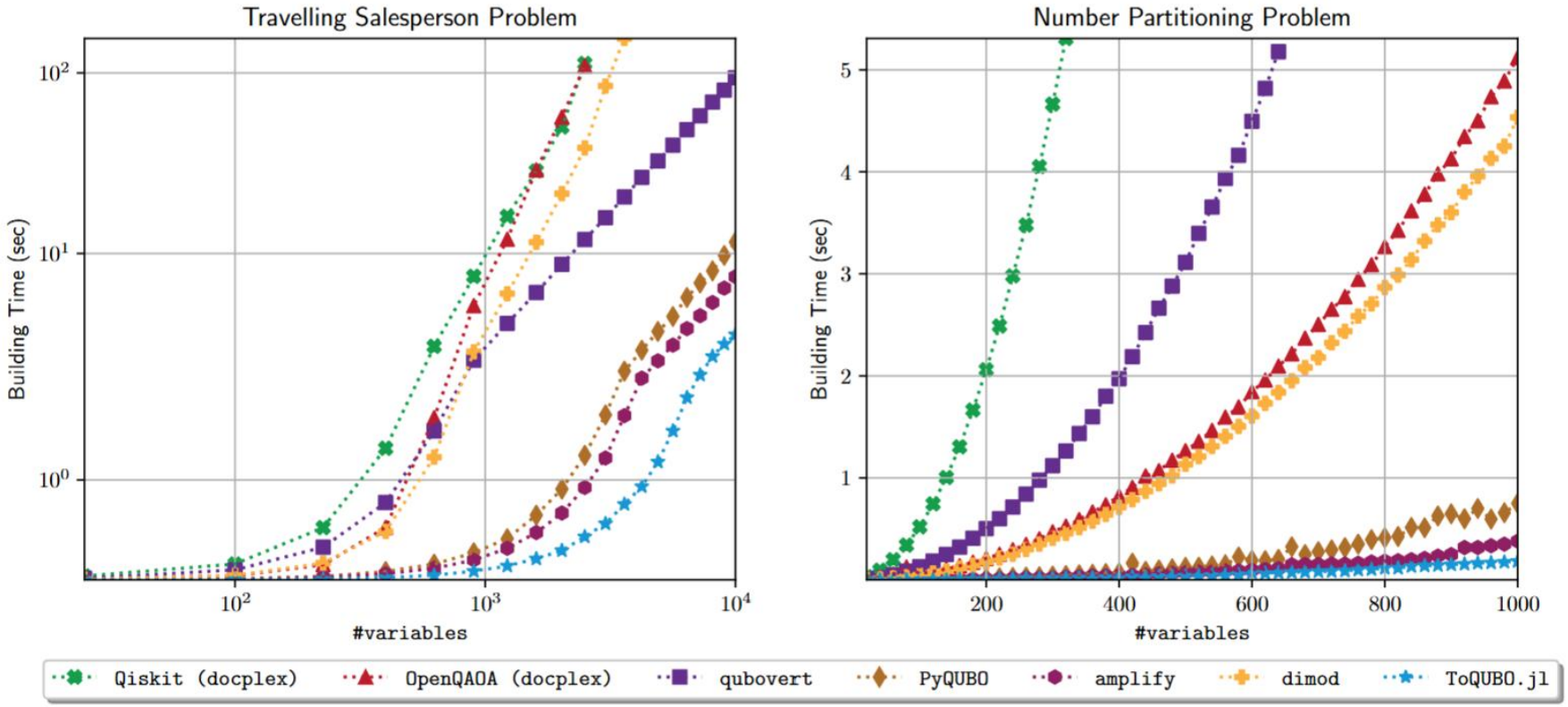


Figure 3.: Benchmark results for the Travelling Salesperson Problem (TSP) and Number Partitioning Problem (NPP) QUBO building time. Each TSP instance is comprised of $\sqrt{\#\texttt{variables}}$ cities. The partitioned sets in the NPP are of the form $[\#\texttt{variables}] = \{1, \ldots, \#\texttt{variables}\}$.

Let’s jump back to the Colab

We’re using JuPyteR notebook hosted in GitHub

https://github.com/SECQUOIA/QUBONotebooks

You can access them directly using Google Colab.

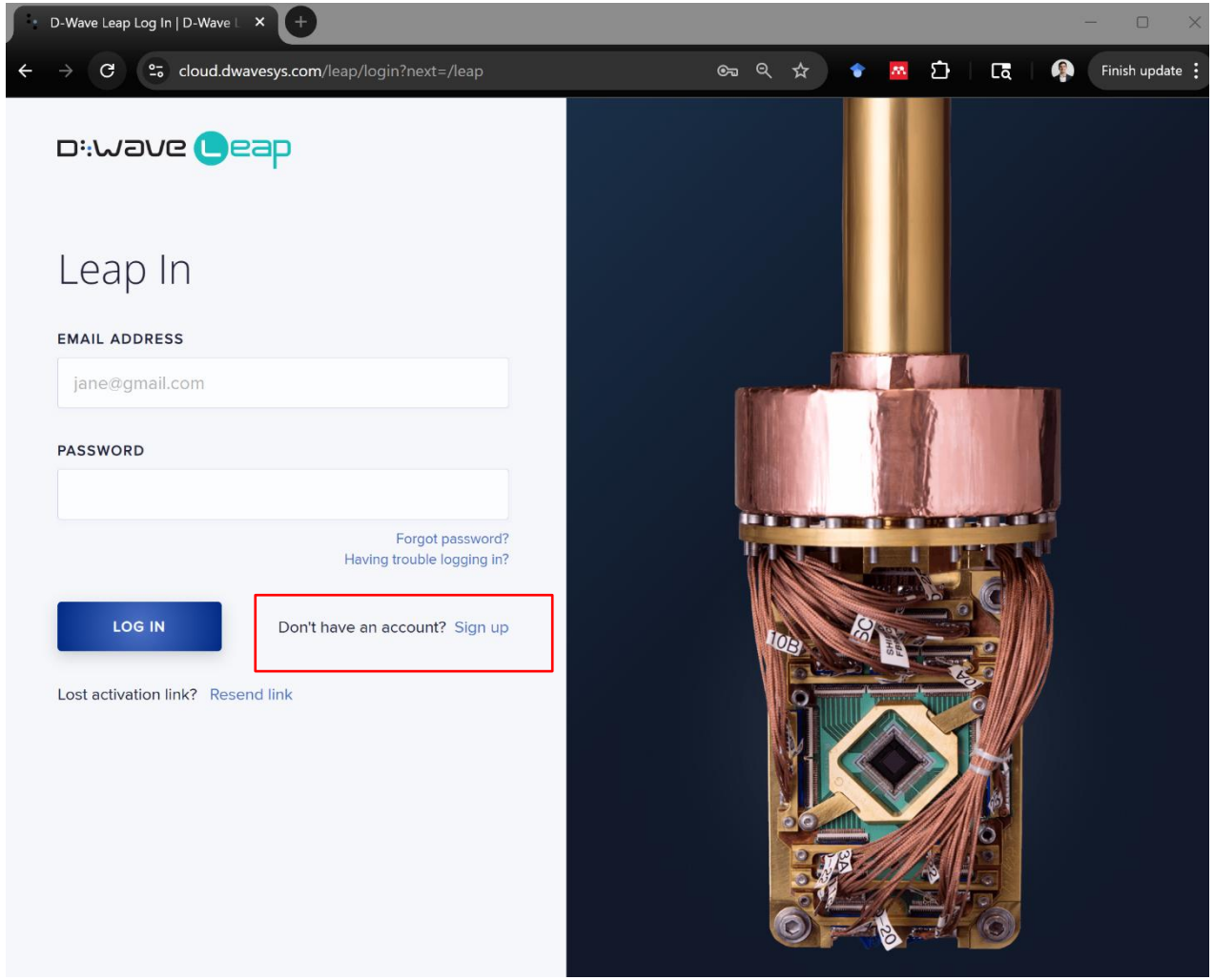
For code written in Julia and that uses the JuMP package

https://colab.research.google.com/github/JuliaQUBO/QUBONotebooks/blob/main/notebooks_jl/2-QUBO.ipynb

If you feel strongly against Julia, there is a second notebook using Python and the package Pyomo

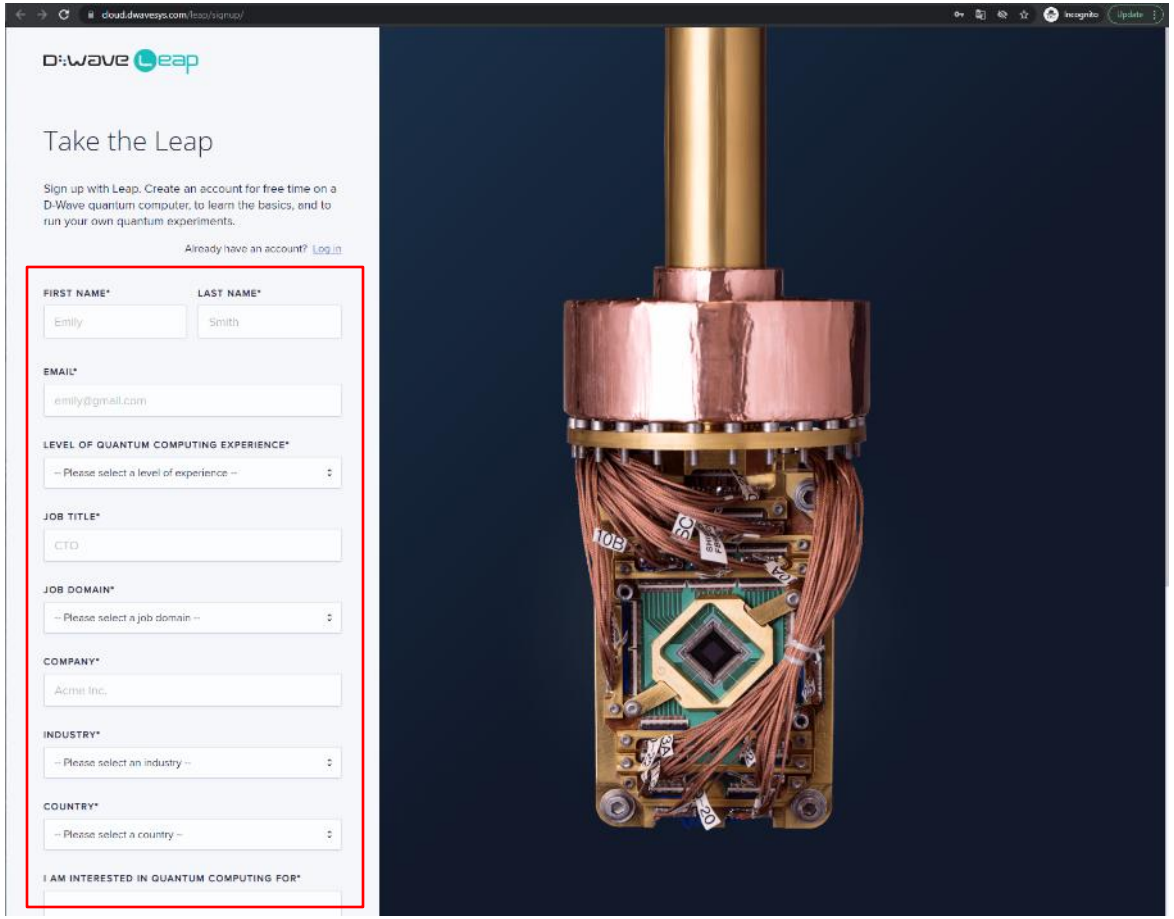
https://colab.research.google.com/github/JuliaQUBO/QUBONotebooks/blob/main/notebooks_py/2-QUBO_python.ipynb

Create D-WAVE (Leap) account



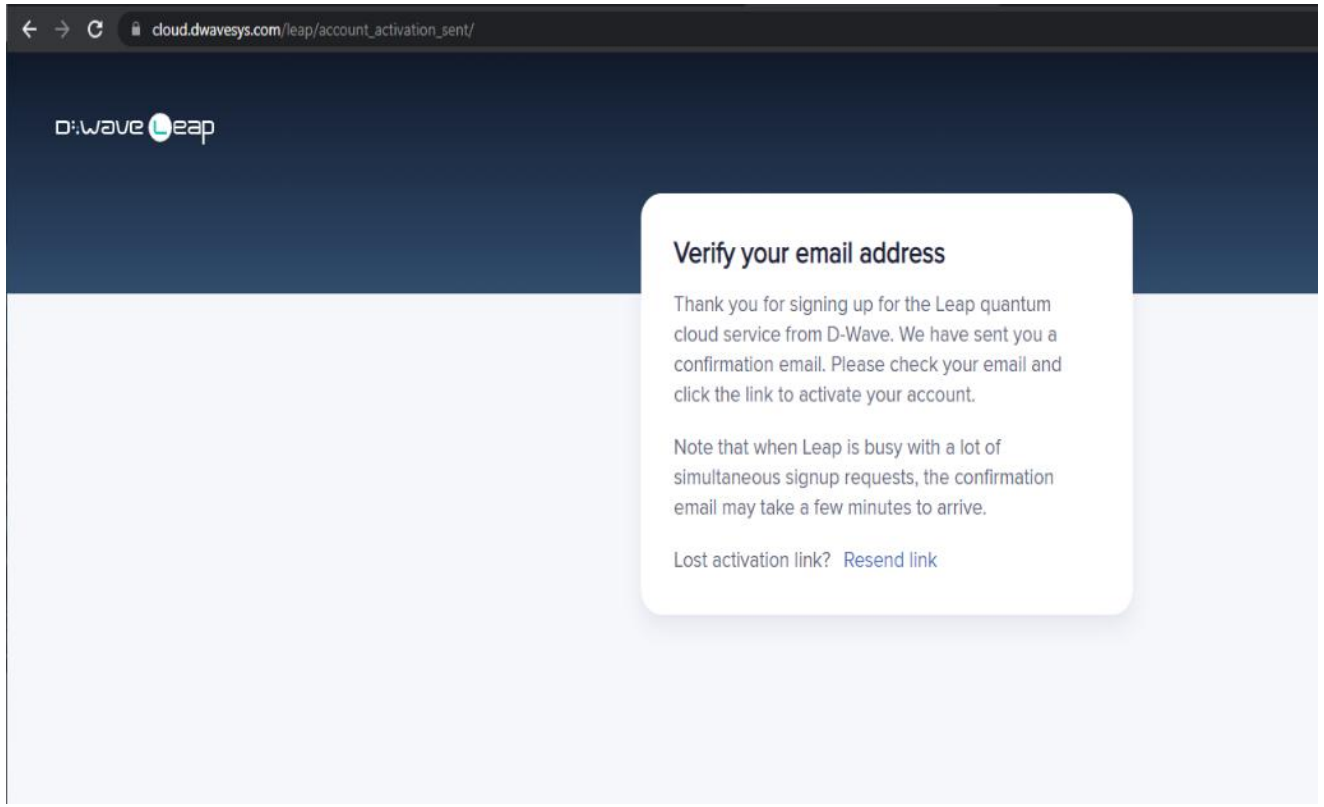
1- Click on “Sign up”

Create D-WAVE (Leap) account



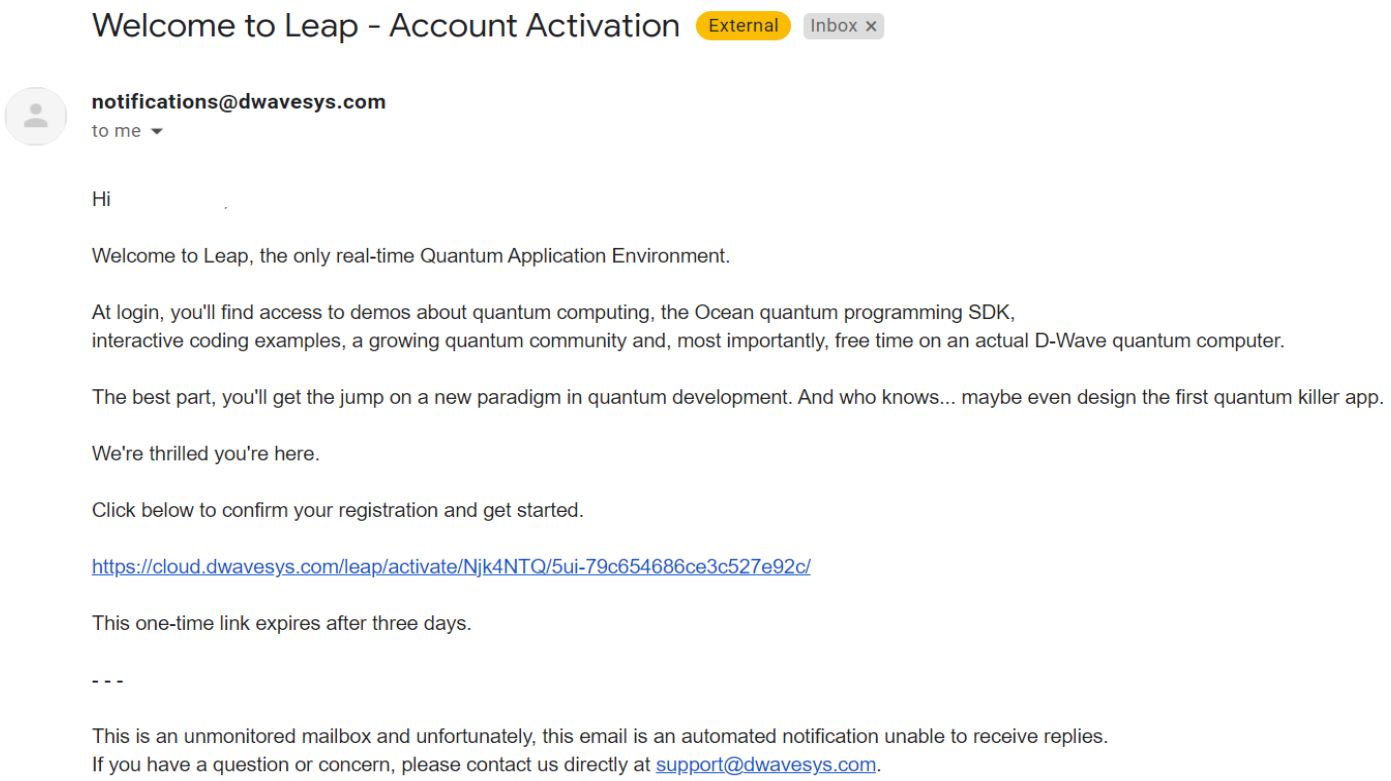
2. Follow the steps

Create D-WAVE (Leap) account



7- Verify your email address

Create D-WAVE (Leap) account



Welcome to Leap - Account Activation External Inbox ×

notifications@dwavesys.com
to me

Hi

Welcome to Leap, the only real-time Quantum Application Environment.

At login, you'll find access to demos about quantum computing, the Ocean quantum programming SDK,
interactive coding examples, a growing quantum community and, most importantly, free time on an actual D-Wave quantum computer.

The best part, you'll get the jump on a new paradigm in quantum development. And who knows... maybe even design the first quantum killer app.

We're thrilled you're here.

Click below to confirm your registration and get started.

https://cloud.dwavesys.com/leap/activate/Njk4NTQ/5ui-79c654686ce3c527e92c/

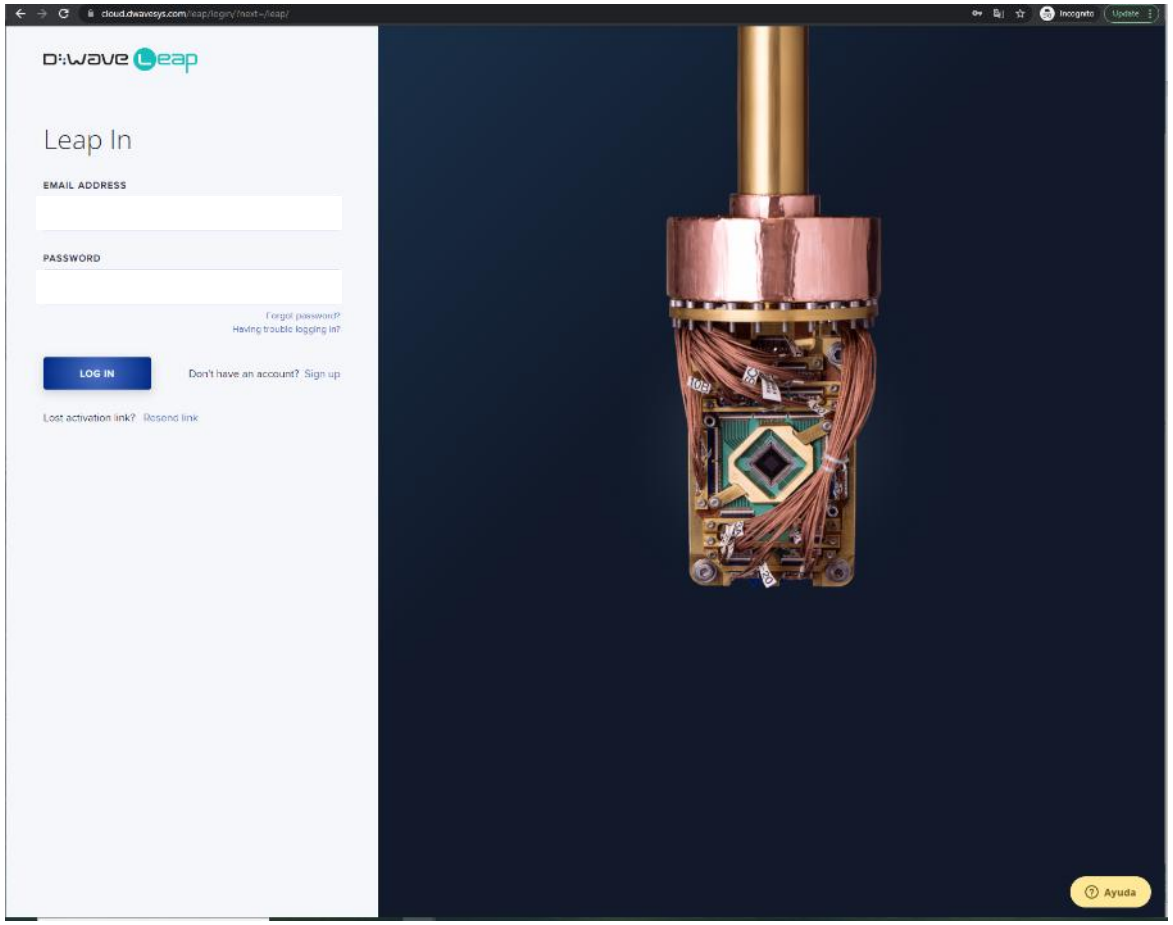
This one-time link expires after three days.

- - -

This is an unmonitored mailbox and unfortunately, this email is an automated notification unable to receive replies.
If you have a question or concern, please contact us directly at support@dwavesys.com.

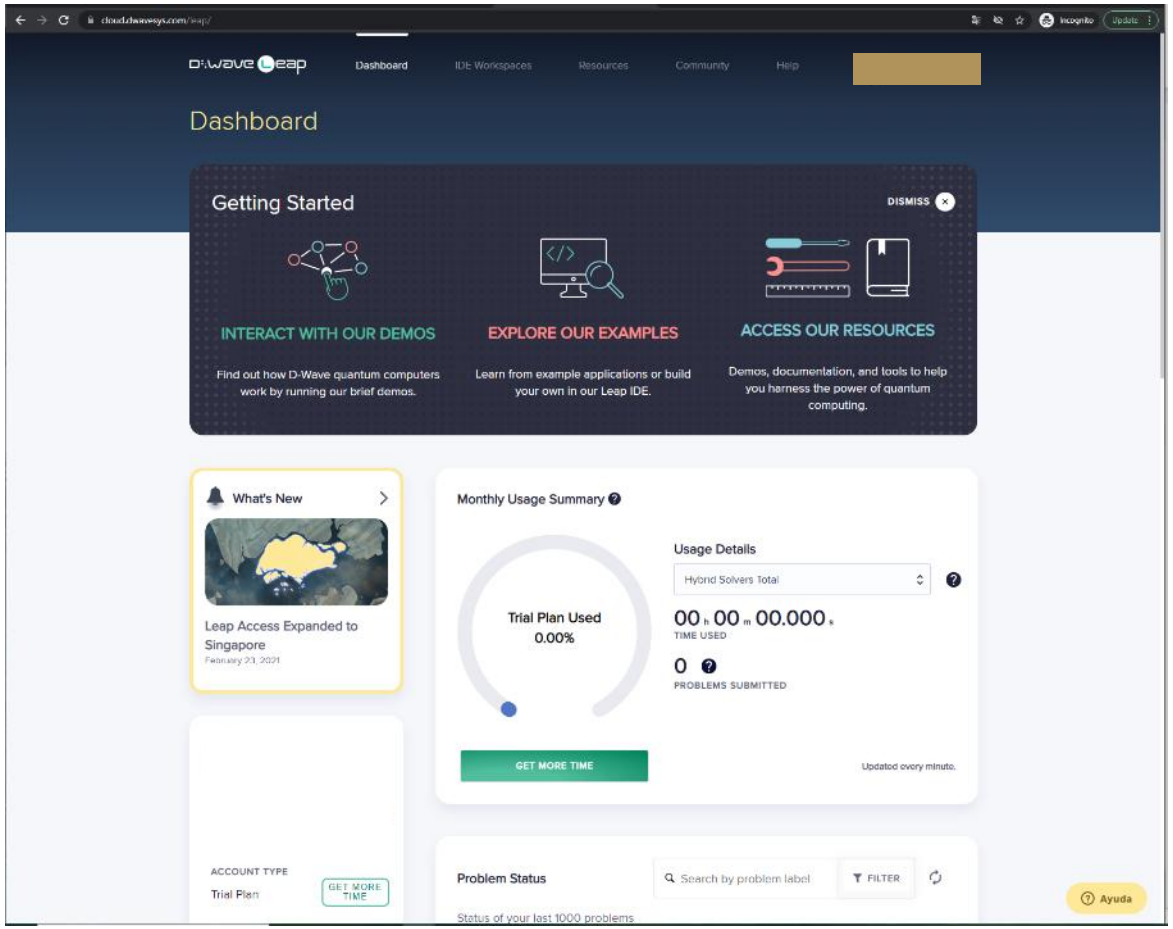
8- Confirm your registration

Create D-WAVE (Leap) account



9- Log in

Create D-WAVE (Leap) account

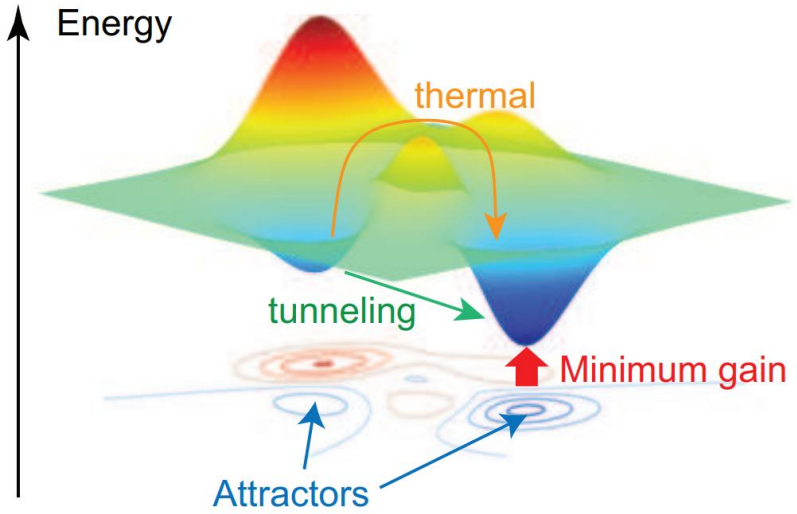


10- Enjoy!

Quantum Methods for QUBO: Why?

Solving an ISING/QUBO problem

What is it? How?

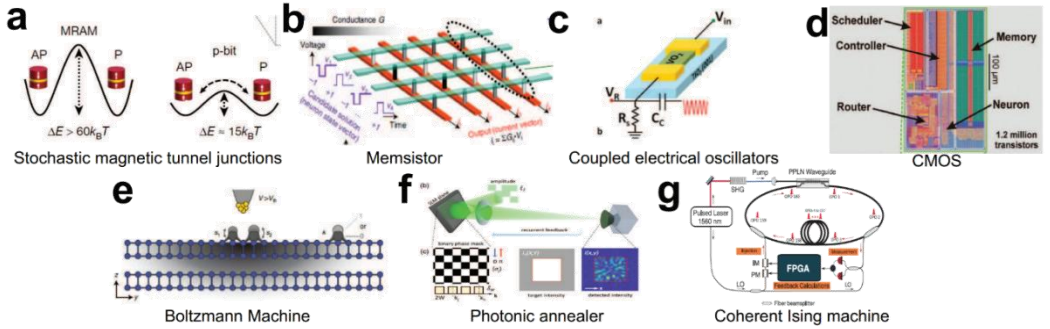


"Solving the Model"

- **Physicists:** identifying thermodynamics expectation values (find the distribution of solutions/energies)
- **Optimizers:** find the actual variable values that minimizes the objective function (find the ground state)

How to solve the Ising model? With an Ising solver!

- **Ising solver?** End-to-end solution method including software/algorithms and hardware/machines
  - **Classical Thermal Annealing**
  - Other classical algorithms
    - MIP or Physics-inspired classical algorithms
  - Hardware solution-based methods
    - **Coherent Ising Machines,** GPU, CMOS, …
  - Quantum Approaches
    - **Quantum Annealing**
    - **Hybrid Quantum-Classical Algorithms**
      - **Variational Quantum Algorithms – QAOA, VQE**



1. Mohseni, Naeimeh, Peter L. McMahon, and Tim Byrnes. "Ising machines as hardware solvers of combinatorial optimization problems." Nature Reviews Physics 4.6 (2022): 363-379.

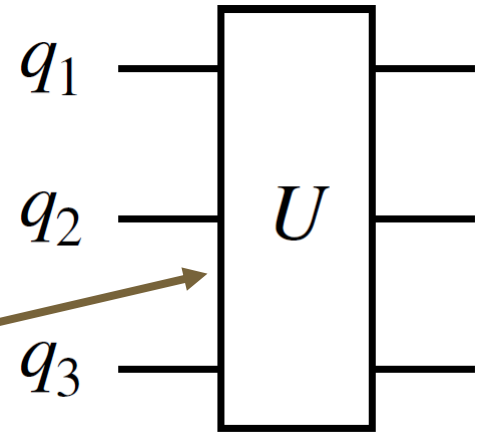
Why Binary Unconstrained in Quantum?

Let's say I want to solve the next problem (PUBO)

$$\min_x E(x)$$

$$= \min_x \sum_i c_i x_i + \sum_{i,j} c_{ij} x_i x_j + \sum_{i,j,k} c_{i,j,k} x_i x_j x_k + \cdots$$

Convert this classical function $E(x)$ into a unitary operator U



$$\text{s.t. } x_i \in \{0,1\}, \forall i \in \{1,\dots,n\} = [\![n]\!]$$

Solution space of the optimization problem maps directly into the basis vectors of quantum state

$$|\psi\rangle = \begin{matrix}|00\rangle\\|01\rangle\\|10\rangle\\|11\rangle\end{matrix}\begin{pmatrix}\psi_{00}\\\psi_{01}\\\psi_{10}\\\psi_{11}\end{pmatrix} =_{eg} \begin{pmatrix}0.0+0.0i\\0.0+0.0i\\1.0+0.0i\\0.0+0.0i\end{pmatrix} = |10\rangle$$

Why QUBO in Quantum?

We have our QUBO

$$\min_x E(x) = \min_x \sum_i c_i x_i + \sum_{i,j} c_{ij} x_i x_j$$

$$\text{s.t. } x_i \in \{0,1\}, \forall i \in \{1, \dots, n\} = [\![n]\!]$$

We only need nearest neighbor interaction for representing the energy:

- “Easy” in current quantum hardware, e.g., 1 and 2 qubit gates

Quantum Methods for QUBO: Quantum Adiabatic Algorithm

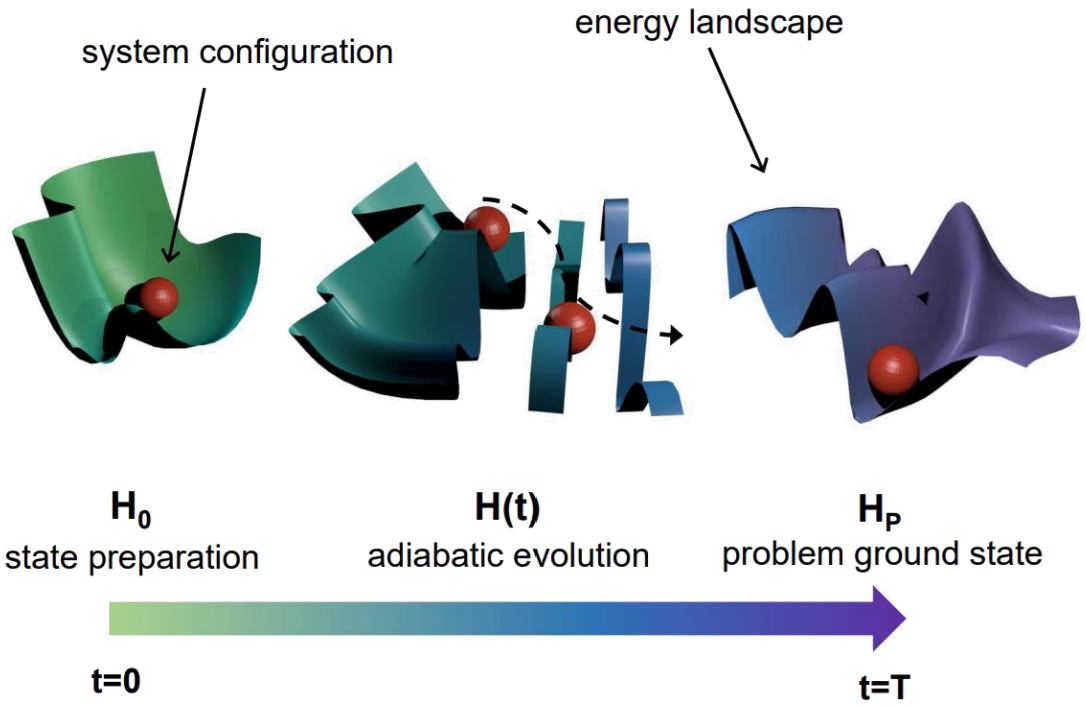
Quantum Adiabatic Algorithm

Quantum Adiabatic Algorithm

1. Switch on a quantum interaction in your system
2. From possible energies of your quantum system set the state to a well-defined energy ranked n^{th} in order of magnitude (e.g. the smallest)
3. Allow the system to evolve (aka Schrödinger evolution no measurement! no noise! T=0K !) that changes the energy states «sufficiently slow».
4. Measure the energy of the state. You will find with 100% probability that the energy is ranked also n^{th}

- **Adiabatic evolution (e.g. slow Schrödinger) preserves the energy ranking of your system.**
- **The smallest energy state (ground state) also maps into the ground state at the end.**

"Adiabaten hypothese": "If a system be affected in a reversible adiabatic way, allowed motions are transformed into allowed motions" (Einstein, 1914).



Solving ISING/QUBOs using Quantum Computing

How?

Quantum Adiabatic Algorithm (QAA)

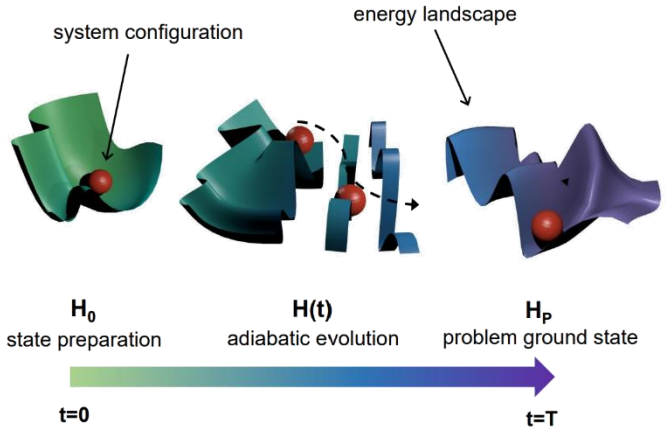
- Adiabatic evolution preserves energy ranking of states
- Start evolution from system at ground-state
- Evolve system adiabatically to one that encodes optimization problem

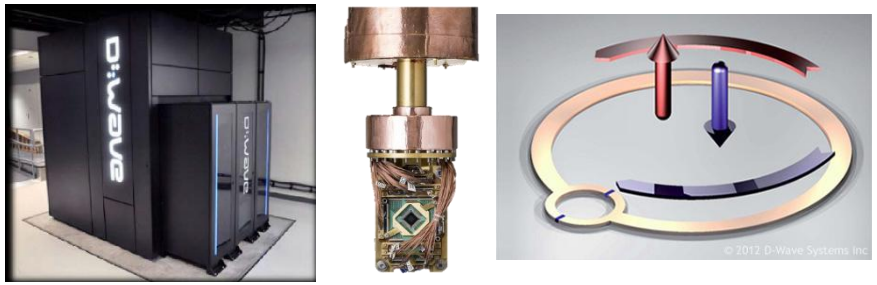
Quantum Annealing

- They run a single quantum algorithm
- Physical implementation of idealized QAA
- Analog computation
- D-Wave quantum annealer is the best-known example

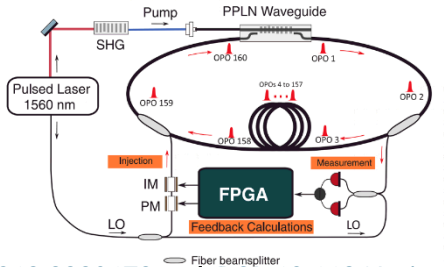
Myriad of physical or Physics-inspired methods

- Coherent Ising Machines, Simulated Bifurcation Machines, Digital annealers, …





Retrieved from www.dwavesys.com/



Retrieved from 10.1109/MSPEC.2018.8389173 and DOI: 10.1126/science.aah5178

1. Pierangeli, Davide, Giulia Marcucci, and Claudio Conti. "Adiabatic evolution on a spatial-photonic Ising machine." Optica 7.11 (2020): 1535-1543.
2. Salloum, Hadi, et al. "Integration of machine learning with quantum annealing." International Conference on Advanced Information Networking and Applications. Cham: Springer Nature Switzerland, 2024.

Mapping classical problems to Quantum Hamiltonians

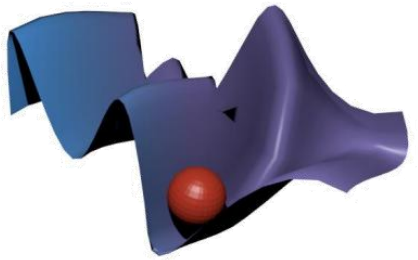
Problem Hamiltonian H_C

From an objective function we can construct a Hamiltonian on n qubits by replacing $\mathbf{x} \in \{0,1\}^n$ with $\mathbf{z} \in \{-1,+1\}^n$ obtaining a polynomial

$$f(\mathbf{z}) = \sum_{\mathcal{C} \subset \{1,\ldots,n\}} \alpha_{\mathcal{C}} \prod_{j \in \mathcal{C}} z_j$$

We replace the Pauli Z operator on qubit i σ_i^z for each z_i variable leading to

$H_C = \sum_{\mathcal{C} \subset \{1,\ldots,n\}} \alpha_{\mathcal{C}} \bigotimes_{j \in \mathcal{C}} \sigma_j^z$:



$\mathbf{H_P}$
problem ground state

$$\begin{bmatrix} 3/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1/4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3/4 \end{bmatrix}$$

Quantum Methods for QUBO: Quantum Annealing

The Quantum Adiabatic Algorithm for Ising Machines

(1) Map objective function into energy of a quantum Ising system

$$H_p = \sum_{ij} J_{ij}\sigma_Z^i \otimes \sigma_Z^j + \sum_i h_i \sigma_Z^i$$

(2)Start from easy problem to solve with known solution

$$H_D = \Gamma \sum_i \sigma_X^i \qquad |\Psi\rangle_{Nqubits} = \frac{1}{\sqrt{2^N}} \sum_{n=1}^{2^N} |solution(n)\rangle$$

(transverse field)

(3) Do any Schrödinger evolution (no measurement! No noise!)
that changes the energy states «sufficiently slow».

How slow? It depends on the problem, on H_D and on the <u>Annealing Schedule</u>
No way to predict efficiently. Try!

$$H|s_1 s_2 \dots s_N\rangle = E_N|s_1 s_2 \dots s_N\rangle$$

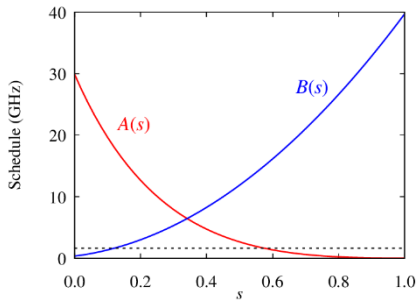
$$\exp(iH)\ |s_1 s_2 \dots s_N\rangle = e^{iE_N}|s_1 s_2 \dots s_N\rangle$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

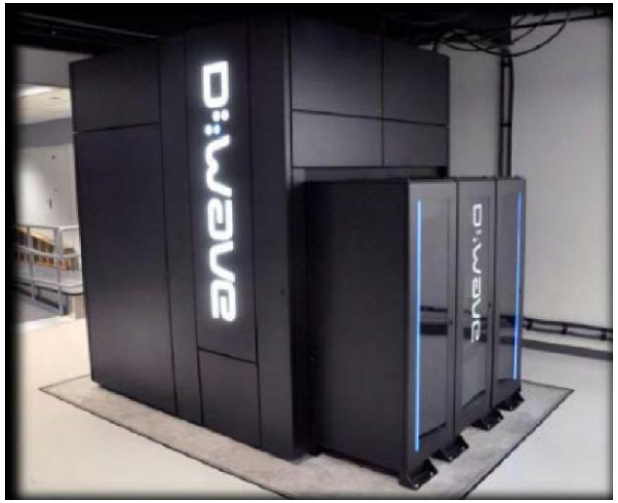
$$R_x(\pi/2)|0\rangle = |1\rangle$$
$$R_x(\pi/2)|1\rangle = |0\rangle$$

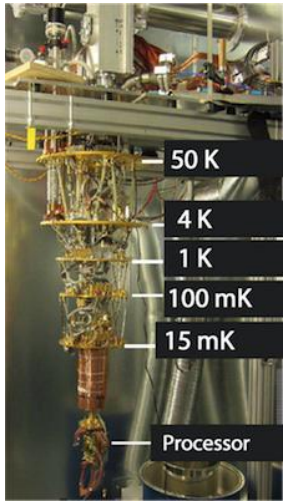
If this field is always on and constant the minimum energy state is the **all-superposed state**

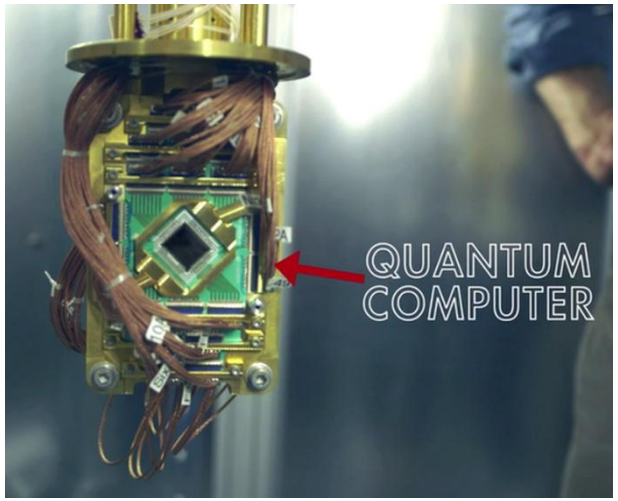
$$\boldsymbol{H = A(t)H_D + B(t)H_p}$$

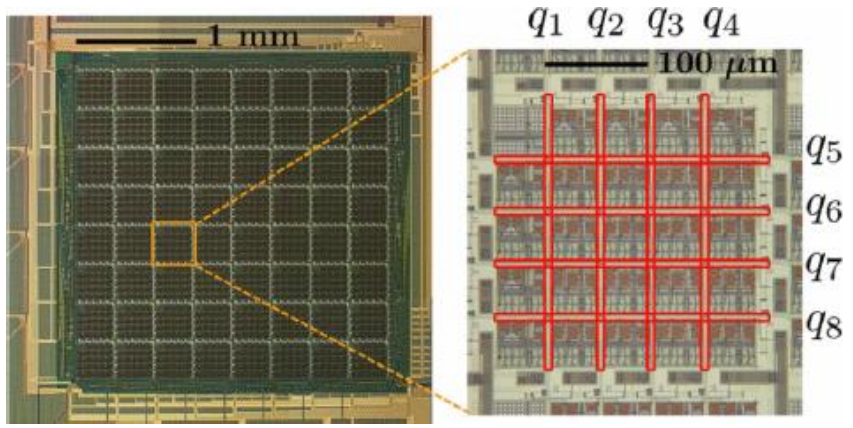


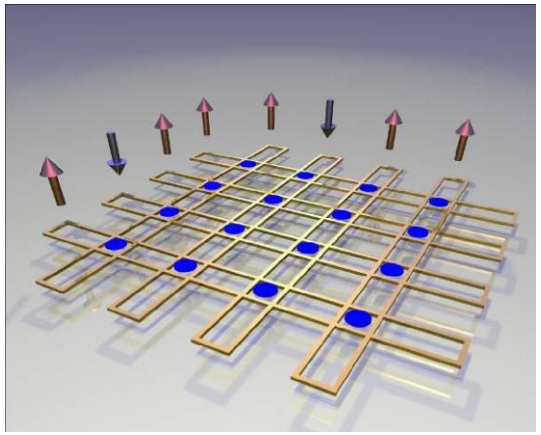
Quantum annealing à la D-Wave

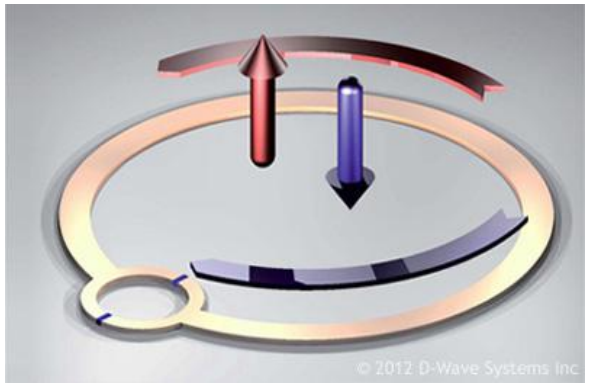


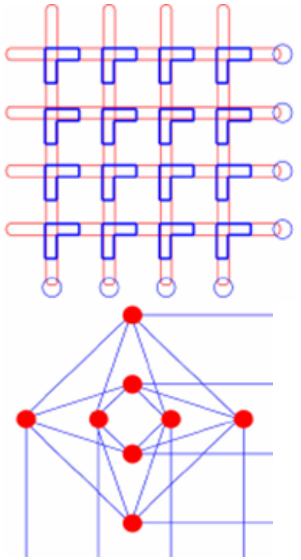


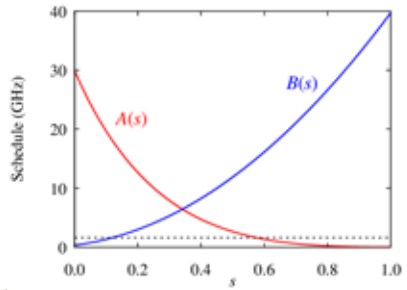




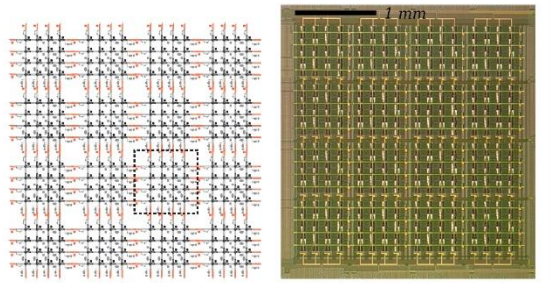








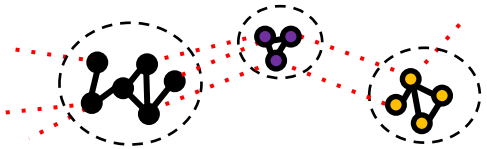
A 128-qubit chip composed of a 4 × 4 array of eight-qubit unit cells.

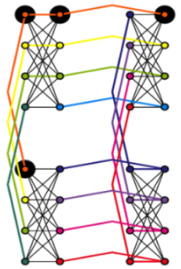


Minor Embedding

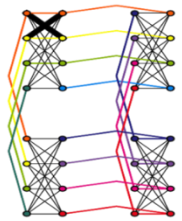
To solve an arbitrary Ising problem its interaction graph must be embedded in a hardware graph.
Why? Quantum Annealer minimizes energy of an Ising spin problem whose interactions lie in the edges of a graph limited by hardware

- Parameter Setting

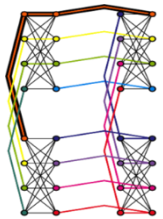




$$\sum_{j\in\varepsilon(i)} h'_j = h_i$$

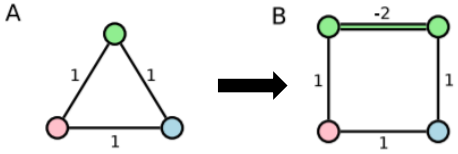


$$\sum_{j_1\in\varepsilon(i_1)}\sum_{j_2\in\varepsilon(i_2)} J'_{j_1j_2} = J_{i_1i_2}$$

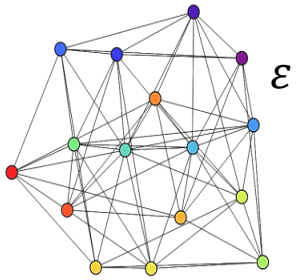


$$J'_{j_1j_2} < |h_i| - \sum_{k=1}^{n} |J_{ij}|$$

- Topological Embedding

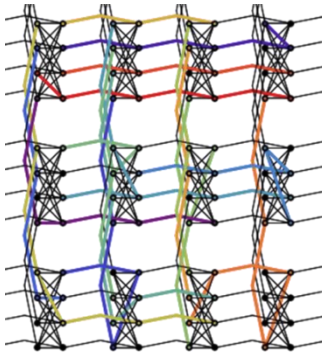


(n_H hardware qubits) (n_P logical bits)

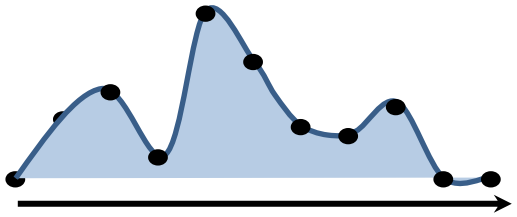


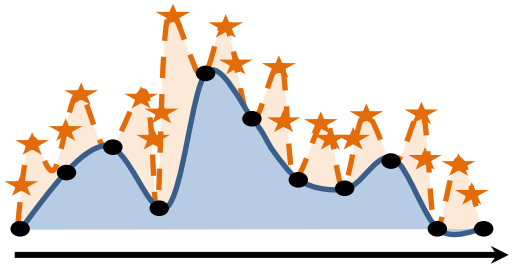
$$\varepsilon(i)\colon \{1,\dots,n_L\} \to 2^{\{1,\dots,n_P\}}$$

Assign "colors" to connected sets of qubits



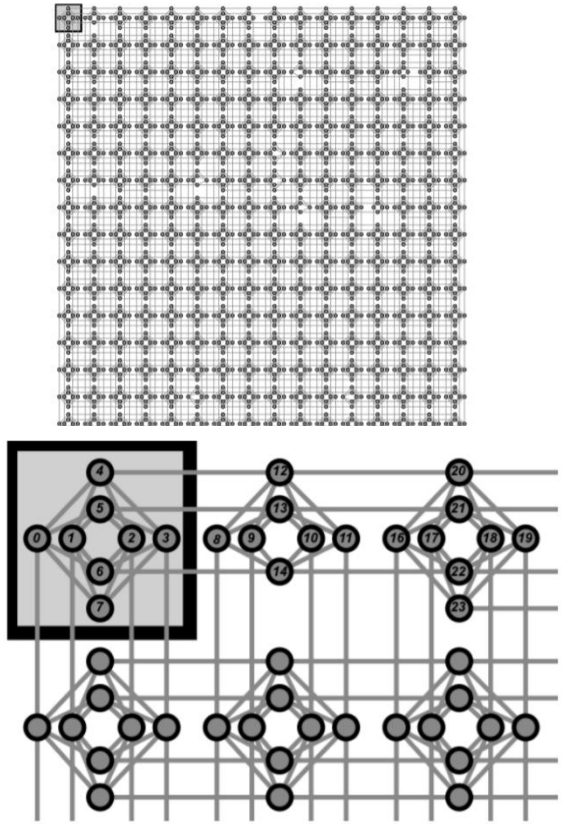
Energy Landscape Before embedding



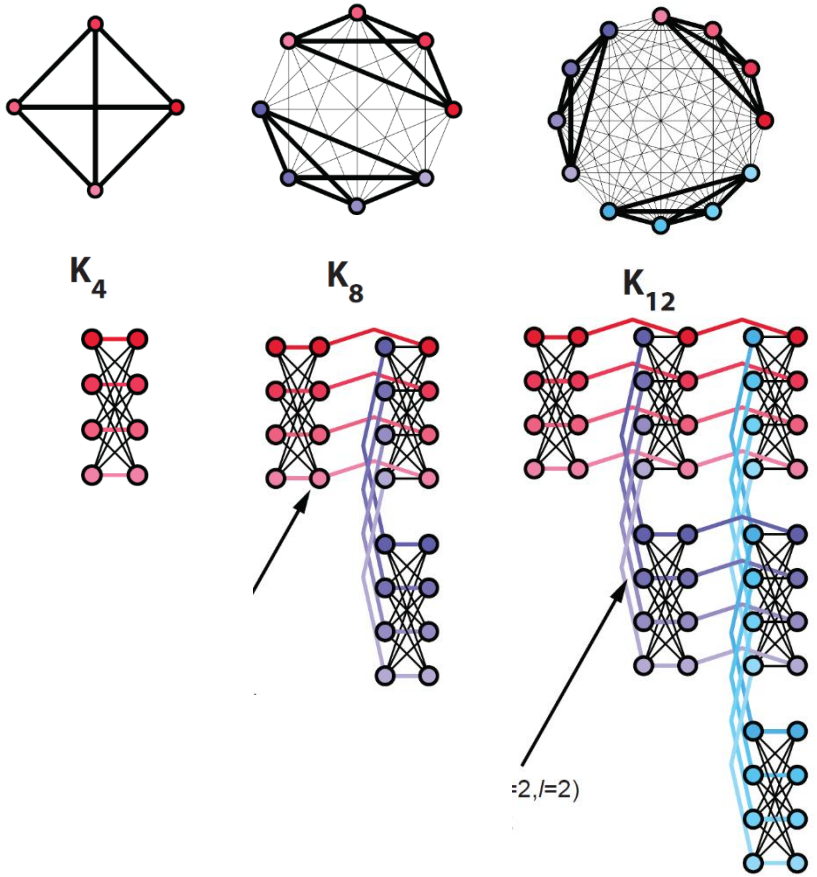


Minor Embedding of a fully connected graph

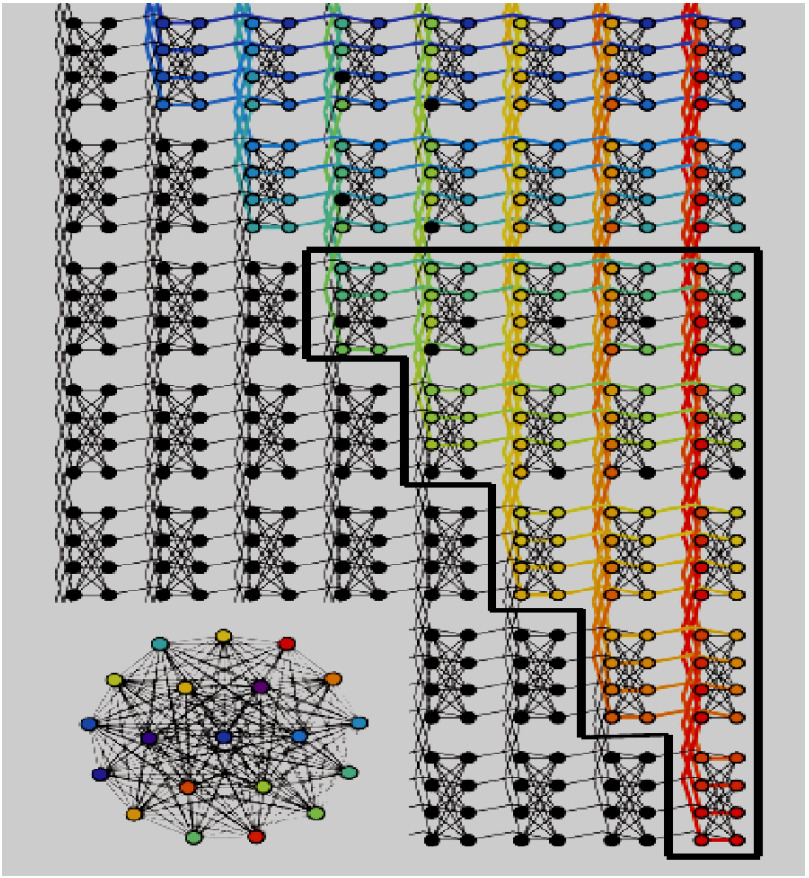
C16 Chimera graph in D-Wave 2000Q. 16x16 unit cells, each cell is a K4,4 bipartite graph



- Systematic Rule for Embedding



- Quadratic overhead



[1] https://arxiv.org/pdf/1810.01440.pdf

Minor Embedding - Example

$$\min_{\mathbf{z}\in\{-1,+1\}^3} z_1z_2 + z_2z_3 + z_3z_1$$

$$\min_{\mathbf{y}\in\{-1,+1\}^3} y_1y_2 + y_2y_3 + y_3y_4$$

$$\text{s.t. } y_1 = y_4$$

Penalizing the new constraint we obtain

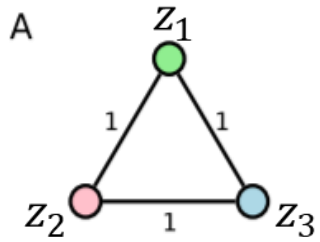
$$\min_{\mathbf{y}\in\{-1,+1\}^3} y_1y_2 + y_2y_3 + y_3y_4 + \rho(y_1 - y_4)^2$$

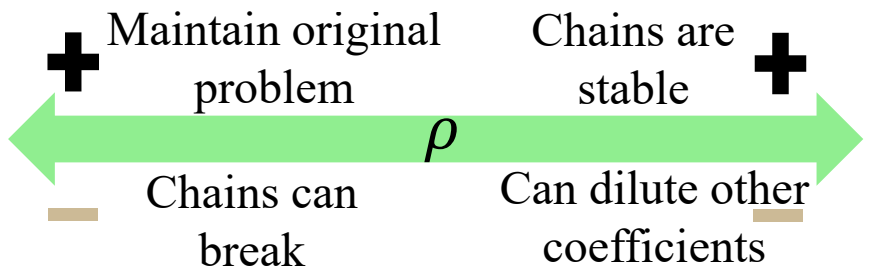
$$\min_{\mathbf{y}\in\{-1,+1\}^3} y_1y_2 + y_2y_3 + y_3y_4 - \rho y_1y_4$$

ρ multiplier known as Chain strength

- **Proposal:** Make it a factor of the Q coefficients

Classical example of minor-embedding of triangular graph into square-graph [2]



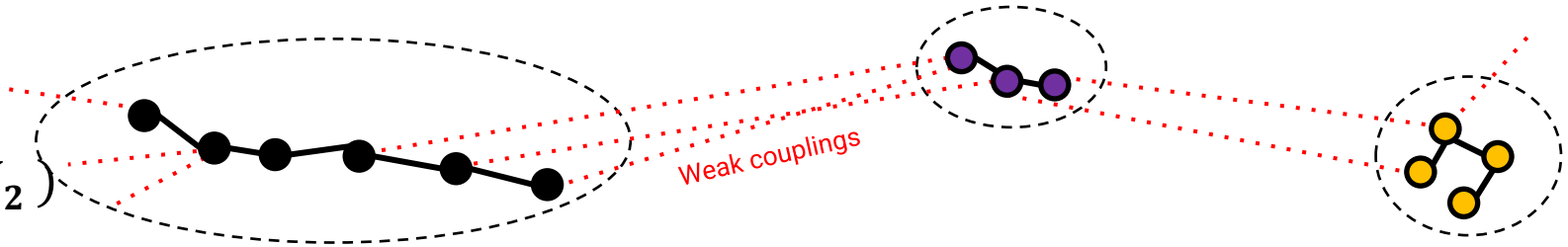


[1] https://arxiv.org/pdf/1810.01440.pdf
[2] https://www.dwavesys.com/sites/default/files/14-1020A-A_Virtual_Graphs_for_High_Performance_Embedded_Topologies.pdf

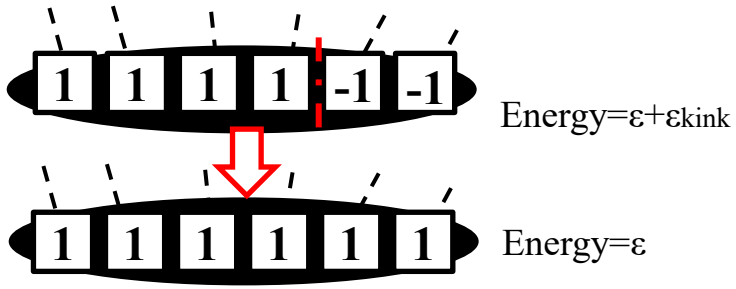
Unembedding

Ferromagnetic Coupling

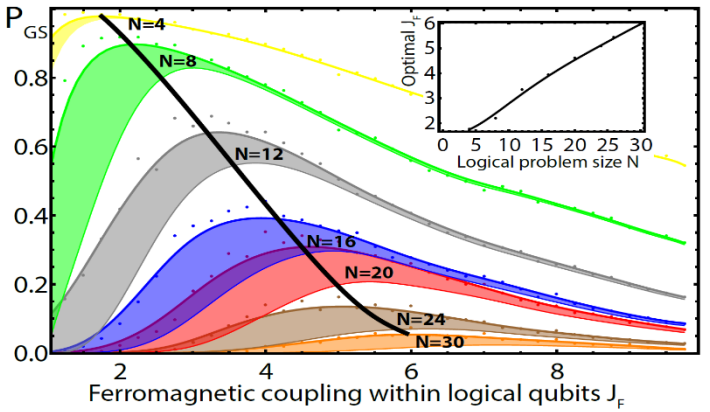
$$f(S_1, S_2) = -J_F s_1 s_2$$
$$f(X_1, X_2) = -4J_F(-X_1 - X_2 + X_1X_2)$$

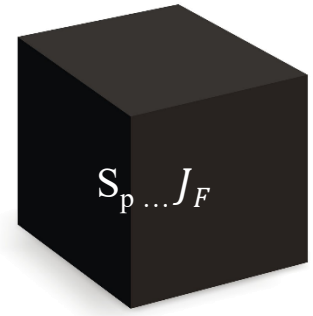


Majority Voting



What is the correct J_F ?





Not too large, not too small. Trial and error.
(See Venturelli et al.
https://journals.aps.org/prx/abstract/10.1103/PhysRevX.5.031040)

Minor Embedding - Example

From our main example

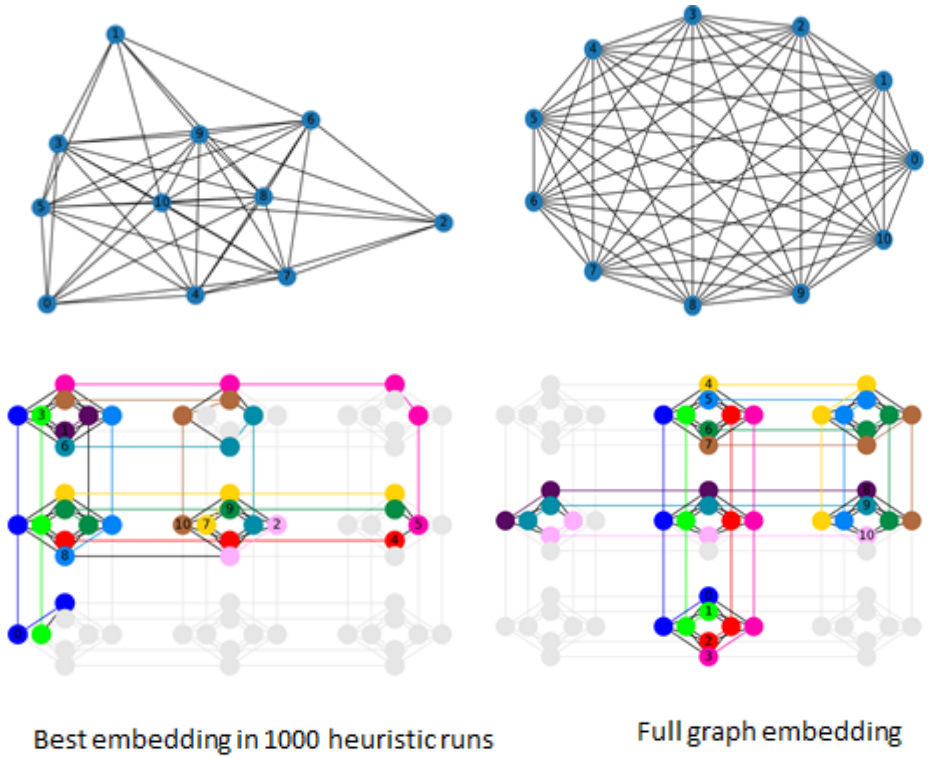
$$\begin{bmatrix}
-46. & 0. & 0. & 48. & 48. & 48. & 0. & 48. & 48. & 48. & 48. \\
0. & -44. & 0. & 48. & 0. & 48. & 48. & 0. & 48. & 48. & 48. \\
0. & 0. & -44. & 0. & 48. & 0. & 48. & 48. & 48. & 48. & 48. \\
48. & 48. & 0. & -92. & 48. & 96. & 48. & 48. & 96. & 96. & 96. \\
48. & 0. & 48. & 48. & -92. & 48. & 48. & 96. & 96. & 96. & 96. \\
48. & 48. & 0. & 96. & 48. & -92. & 48. & 48. & 96. & 96. & 96. \\
0. & 48. & 48. & 48. & 48. & 48. & -91. & 48. & 96. & 96. & 96. \\
48. & 0. & 48. & 48. & 96. & 48. & 48. & -92. & 96. & 96. & 96. \\
48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & -139. & 144. & 144. \\
48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & 144. & -138. & 144. \\
48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & 144. & 144. & -139.
\end{bmatrix}$$

And embed it into a Chimera graph (subgraph of the Chip)

Notice that we need to “duplicate” certain variables into several qubits

This step is non-trivial:

Either use heuristic methods or solve highly constrained problem



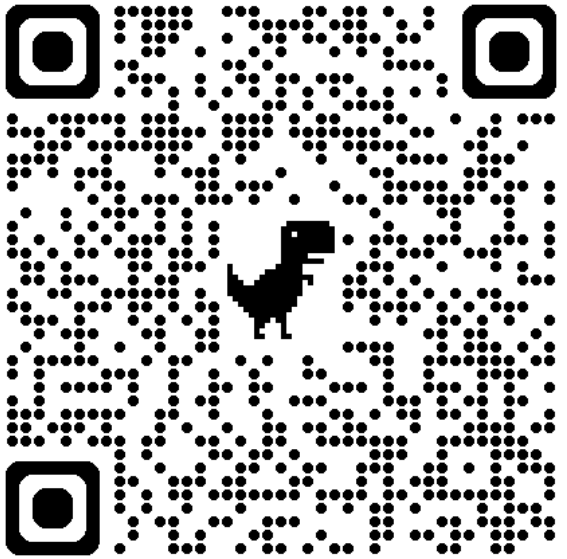
Best embedding in 1000 heuristic runs

Full graph embedding

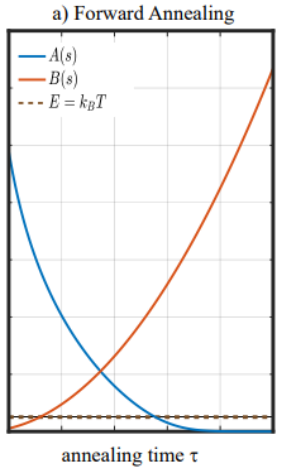
First things first – Colab notebook

We are finally going to run on Quantum devices!

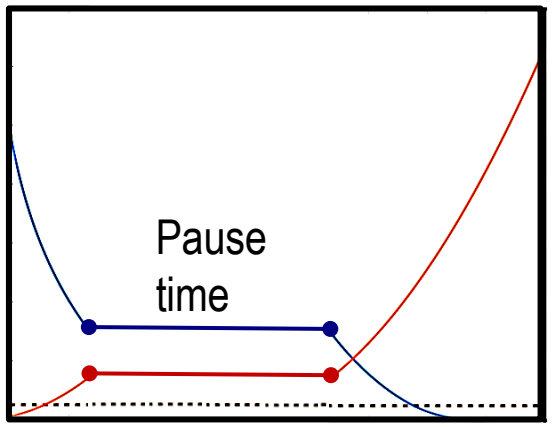
https://github.com/SECQUOIA/QUBONotebooks/blob/main/notebooks_py/4-DWAVE_python.ipynb



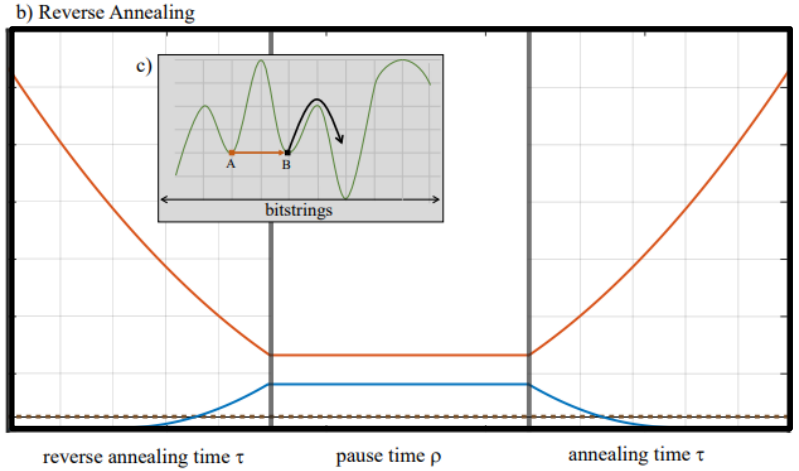
Annealing Schedule Parameters



Time for annealing



Paused annealing



First pause time

reversal time

These are all parameters that influence performance.

Only for elegant problems they can be derived ab-initio. In the real world you have some physics guidance for best guess then you use a heuristics to find them

Power of Pausing: Advancing Understanding of Thermalization in Experimental Quantum Annealers

Jeffrey Marshall,[1,2,3] Davide Venturelli,[1,4] Itay Hen,[3,5] and Eleanor G. Rieffel[4]

Why and when is pausing beneficial in quantum annealing?

Huo Chen[1,2] and Daniel A. Lidar[1,2,3,4]

Ferromagnetically shifting the power of pausing

Zoe Gonzalez Izquierdo,[1,2,3] Shon Grabbe,[2] Stuart Hadfield,[2,3] Jeffrey Marshall,[2,3] Zhihui Wang,[2,3] and Eleanor Rieffel[2]

Search range in experimental quantum annealing

Nicholas Chancellor and Viv Kendon
Department of Physics; Joint Quantum Centre (JQC) Durham-Newcastle Durham University, South Road, Durham, DH1 3LE, UK
(Dated: August 26, 2020)

Reverse Quantum Annealing Approach to Portfolio Optimization Problems

Davide Venturelli[1] Alexei Kondratyev[2]

[1] *USRA Research Institute for Advanced Computer Science, Mountain View, CA 94035, USA*
[2] *Data Analytics Group, Financial Markets, Standard Chartered Bank, London EC2V 5DD, UK*

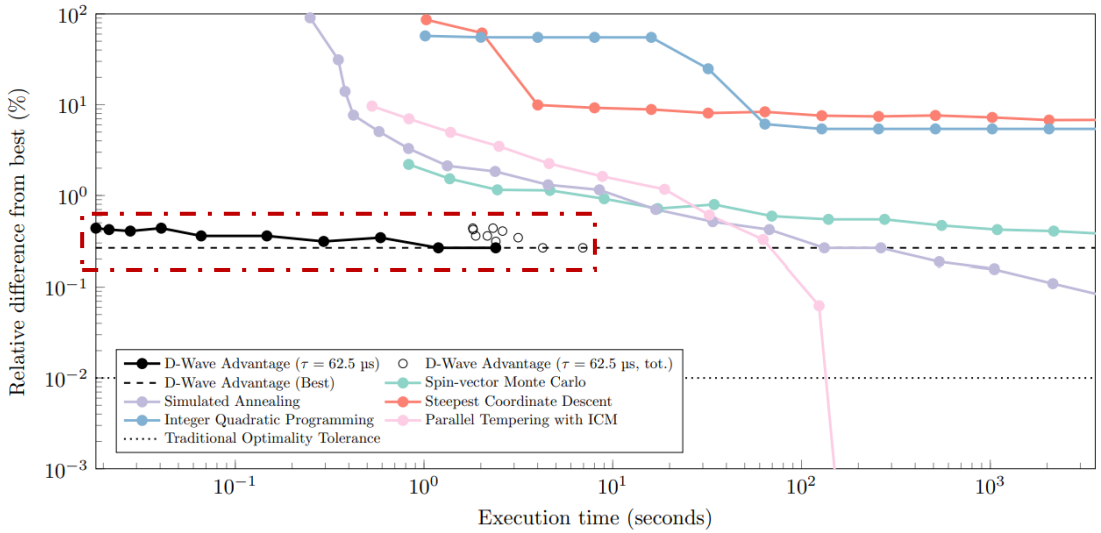
Comparing relaxation mechanisms in quantum and classical transverse-field annealing

Tameem Albash[1,2] and Jeffrey Marshall[3,4,*]

Relevant literature to understand performance on pause parameters.

Solving ISING/QUBOs via Quantum Computing

Quantum Annealing



Empirical performance advantages [1]

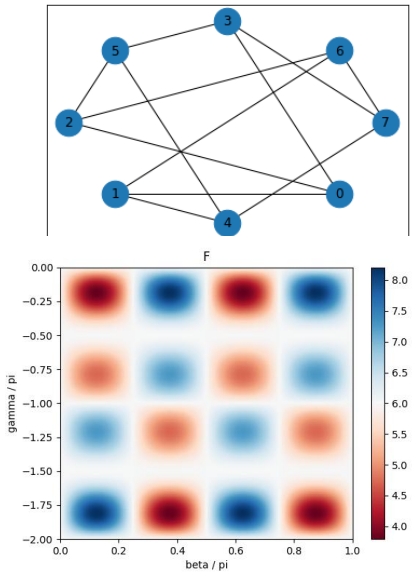
Limited by size and connectivity

Understood as Heuristic methods

Gate-based computing (QAOA)

For quite general problems (MAXCUT 3-reg-degree graphs) provides provably better approximation ratio than random guessing [2].

For a particular problem (SAT E3Lin2), QAOA provided the best approximation ratio [3].



Interesting theoretical guarantees

Even more limited by size and noise

Approximation algorithms

1. https://arxiv.org/pdf/2210.04291.pdf
2. https://arxiv.org/pdf/1411.4028.pdf
3. https://arxiv.org/pdf/1709.03489.pdf

Quantum Methods for QUBO: Gate-based computers QAOA

Thanks to Davide Venturelli from USRA and NASA QuAIL and many more for inspiration for these slides!

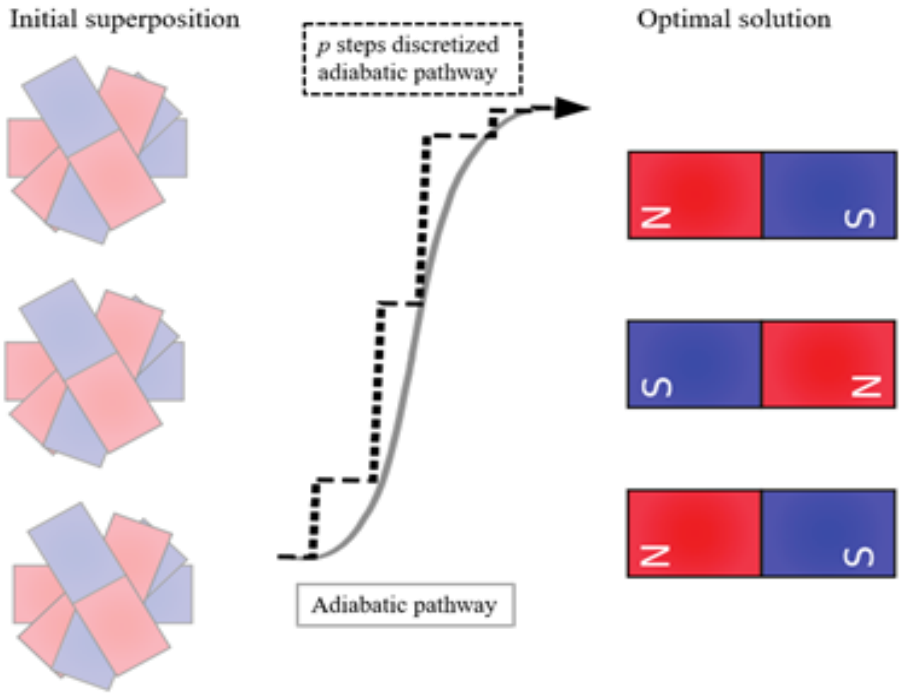
Quantum Optimization for discrete optimization

Recipe for solving discrete optimization problems with gate-based computers

1. Map the optimization problem to QUBO/Ising
2. Assign variable in QUBO/Ising to a qubit in a system.
3. Apply a circuit, aiming to maximize the probability of measuring the optimal solution of the problem.
4. Measure the state with output as qubit values.
   - These values yield the optimal solution with some probability
5. Repeat this procedure several times and return the best-found solution

Proposal: Quantum Alternating Optimization Ansatz (QAOA)

Variational algorithm for the QUBO/Ising problem and inspired in QAA



QAOA Tutorial Outline

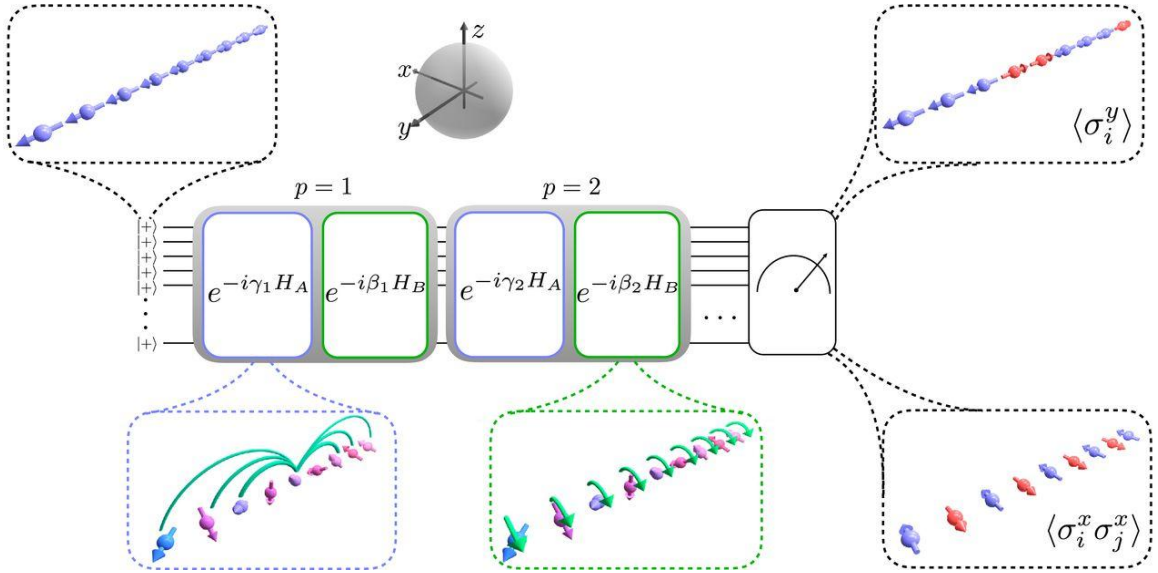
- Quantum Approximate Optimization Algorithm: review and status
- The «Quantum Alternating Operator Ansatz»
  - Mixing Operators
  - Examples
- Compiling and Executing
  - The gate synthesis problem
  - Review of compilation methods
  - Compiling framework in nearest-neighbor architectures

READING LIST

- **Quantum Approximate Optimization with Hard and Soft Constraints.** Hadfield, S., Wang, Z., Rieffel, E. G., O'Gorman, B., Venturelli, D., & Biswas, R. (2017, November). In *Proceedings of the Second International Workshop on Post Moores Era Supercomputing* (pp. 15-21). ACM.
- **From the quantum approximate optimization algorithm to a quantum alternating operator Ansatz** Hadfield, S, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas. *arXiv preprint arXiv:1709.03489 (2017)*. Algorithms (2019).
  - **Best Paper Award MDPI Algorithms Journal**

Quantum Approximate Optimization Algorithm

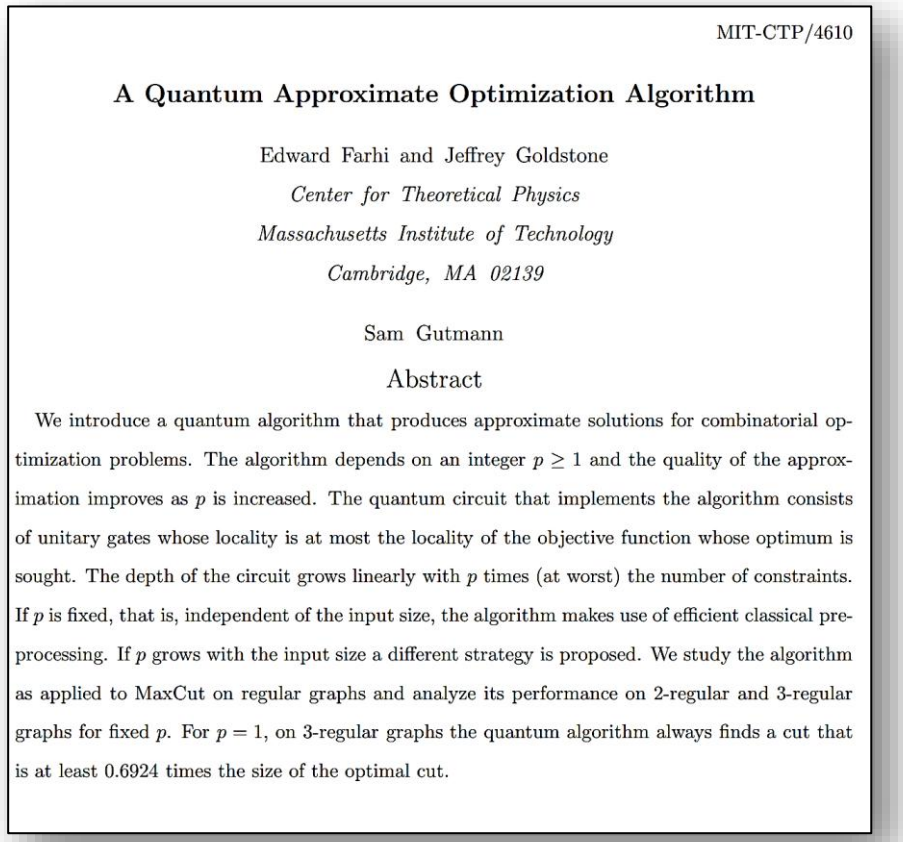
- Gate-based quantum algorithm for QUBO optimization
- Iteratively alternates p times between applying two sets of operators: Mixing and Phase Shifting/Driving
  - Induce entanglement and the objective function
- Requires as many qubits as the size of the problem
- Requires polynomially many gates compared to the problem size
- Is an approximation algorithm:
  - One can theoretically prove that solution to any problem within a certain class using this algorithm will always be in a range (approximation ratio) of the true optimal



Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator
Guido Pagano, Aniruddha Bapat, Patrick Becker, Katherine S. Collins, Arinjoy De, Paul W. Hess, Harvey B. Kaplan, Antonis Kyprianidis, Wen Lin Tan, Christopher Baldwin, Lucas T. Brady, Abhinav Deshpande, Fangli Liu, Stephen Jordan, Alexey V. Gorshkov, Christopher Monroe
Proceedings of the National Academy of Sciences Oct 2020, 117 (41) 25396-25401; DOI: 10.1073/pnas.2006373117

- For MAXCUT of regular 3-degree graphs QAOA with p=1 has approximation ratio of 0.6942 vs. 2/3 of random guessing.
- For a satisfiability problem E3Lin2, QAOA with p=1 gave the best approximation ratio at the point.

Origins of the QAOA



MIT-CTP/4610

A Quantum Approximate Optimization Algorithm

Edward Farhi and Jeffrey Goldstone
Center for Theoretical Physics
Massachusetts Institute of Technology
Cambridge, MA 02139

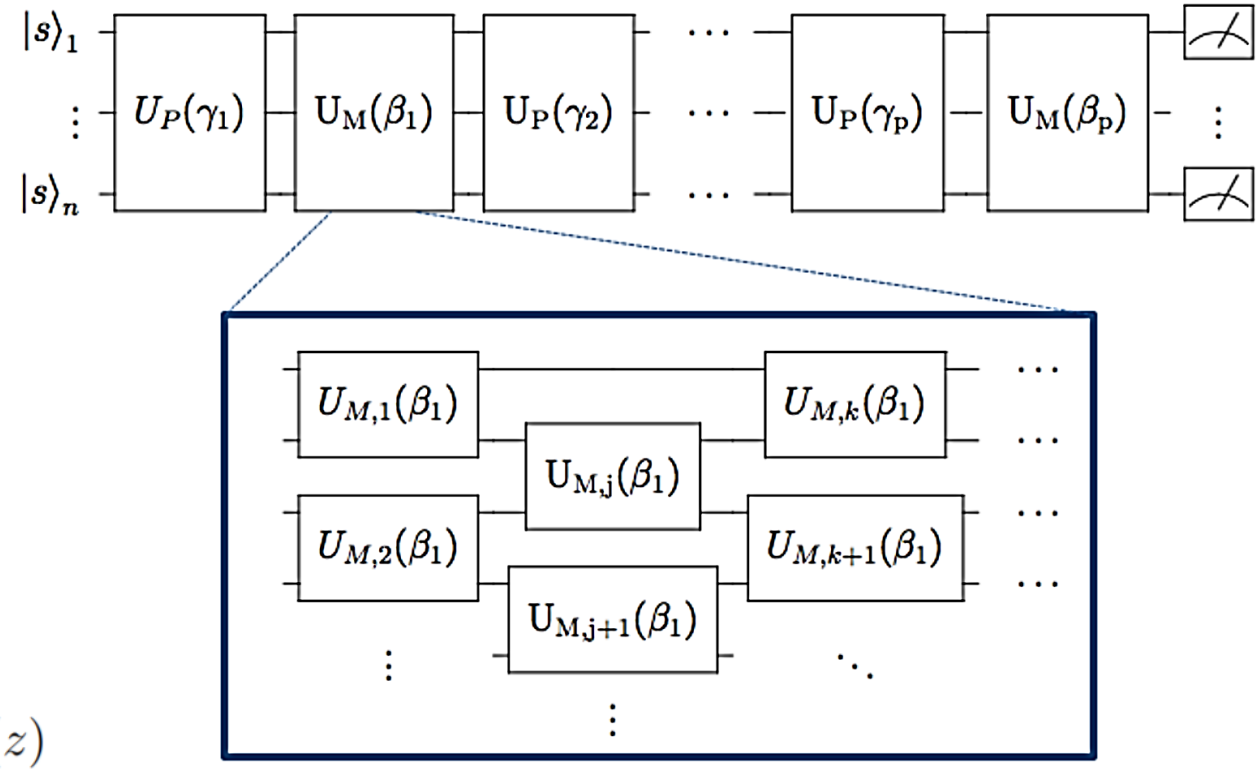
Sam Gutmann

Abstract

We introduce a quantum algorithm that produces approximate solutions for combinatorial optimization problems. The algorithm depends on an integer $p \geq 1$ and the quality of the approximation improves as p is increased. The quantum circuit that implements the algorithm consists of unitary gates whose locality is at most the locality of the objective function whose optimum is sought. The depth of the circuit grows linearly with p times (at worst) the number of constraints. If p is fixed, that is, independent of the input size, the algorithm makes use of efficient classical preprocessing. If p grows with the input size a different strategy is proposed. We study the algorithm as applied to MaxCut on regular graphs and analyze its performance on 2-regular and 3-regular graphs for fixed p. For $p = 1$, on 3-regular graphs the quantum algorithm always finds a cut that is at least 0.6924 times the size of the optimal cut.

$$|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle = Q_p(\boldsymbol{\beta}, \boldsymbol{\gamma}) \, |s\rangle$$

$$Q_p(\boldsymbol{\beta}, \boldsymbol{\gamma}) = U_{\mathrm{M}}(\beta_p) U_{\mathrm{P}}(\gamma_p) \cdots U_{\mathrm{M}}(\beta_1) U_{\mathrm{P}}(\gamma_1)$$



$$F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \boldsymbol{\gamma}, \boldsymbol{\beta}| \, C \, |\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle \qquad M_p \geq M_{p-1}$$

$$M_p = \max_{\boldsymbol{\gamma}, \boldsymbol{\beta}} F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) \qquad \lim_{p \to \infty} M_p = \max_z C(z)$$

QAOA

- Design a binary optimization classical Hamiltonian (“phase separation”)
- Design a unitary operator that can connect and allow jumps between different states (“mixing”)
- Prepare a QAOA state for some parameters

$$|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle = Q_p(\boldsymbol{\beta}, \boldsymbol{\gamma}) \, |s\rangle$$

$$Q_p(\boldsymbol{\beta}, \boldsymbol{\gamma}) = U_{\mathrm{M}}(\beta_p) U_{\mathrm{P}}(\gamma_p) \cdots U_{\mathrm{M}}(\beta_1) U_{\mathrm{P}}(\gamma_1)$$

- Measure the state in the computational value and compute the exp. value of C(z)

$$F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \boldsymbol{\gamma}, \boldsymbol{\beta}| \, C \, |\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle \qquad M_p \geq M_{p-1}$$

$$M_p = \max_{\boldsymbol{\gamma}, \boldsymbol{\beta}} F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) \qquad \lim_{p \to \infty} M_p = \max_{z} C(z)$$

- Change the parameters if they are not proven optimal and repeat 3-4

Vanilla QAOA

$$O_{\text{QUBO}}(q) = \sum_{i=1}^{N} a_i q_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} b_{ij} q_i q_j$$

Associate one qubit to each q_i

Initialize the registers in a superposition of all possible bitstring

$$|\psi\rangle_{in} = \left(2^{N/2}\right)^{-1} \sum_s |s\rangle$$

Assign to each superposed solution a phase proportional (arbitrary parameter γ_1) to its objective function value

$$|\psi\rangle_{ps(1)} = \left(2^{N/2}\right)^{-1} \sum_s e^{i\gamma_1 E_s} |s\rangle$$

Mix the amplitudes by a transverse field rotation $\exp(i\beta X)$ on each qubit (arbitrary parameter)

$$|\psi\rangle_{mix(1)} = \left(2^{N/2}\right)^{-1} \sum_s \mathrm{B}_{1s}(\beta_1, \gamma_1) e^{i\Gamma_{1s}(\beta_1, \gamma_1)} |s\rangle$$

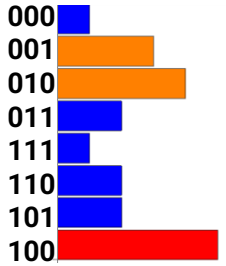
Phase separate again with new γ_2

$$|\psi\rangle_{ps(2)} = \left(2^{N/2}\right)^{-1} \sum_s \mathrm{B}_{1s}(\beta_1, \gamma_1) e^{i(\Gamma_{1s}(\beta_1, \gamma_1) + \gamma_2 E_s)} |s\rangle$$

$$|\psi\rangle_{mix(2)} = \left(2^{N/2}\right)^{-1} \sum_s \mathrm{B}_{2s}(\beta_1, \gamma_1, \beta_2, \gamma_2) e^{i(\Gamma_{2s}(\beta_1, \gamma_1, \beta_2, \gamma_2))} |s\rangle$$

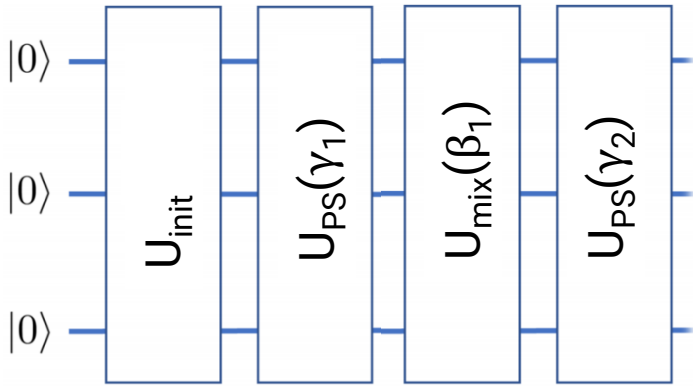
After having repeated the algorithm p times do measure in the computational base the expectation value of the objective function

$$\langle\psi|\mathrm{O}|\psi\rangle_{out} = \sum_s \mathrm{O}_s \left|\mathrm{B}_{ps}\right|^2$$



Quantum Approximate Optimization Algorithm

Example



Initialization operator

Mix the amplitudes by a transverse field rotation $\exp(i\beta X)$ on each qubit (arbitrary parameter)

$$|\psi\rangle_{mix(1)} = \left(2^{N/2}\right)^{-1} \sum_s \mathrm{B}_{1s}(\beta_1, \gamma_1) e^{i\Gamma_{1s}(\beta_1,\gamma_1)} |s\rangle$$

Now if you measure, the probability of a bitstring depends both on γ and β in a non-linear way.

Phase separation operator dependent on a parameter

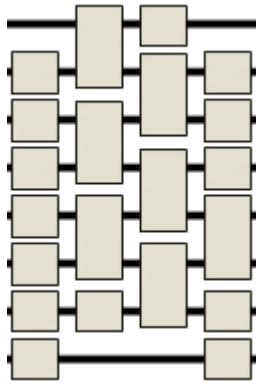
$$\underbrace{\exp(i\beta Z_1 Z_2)}|s_1 s_2\rangle = e^{i\gamma_1 s_1 s_2}|s_1 s_2\rangle$$

Logical 2-qubit gate representing the Ising interaction

Hadamard Gates

$$|\psi\rangle_{in} = \frac{1}{\sqrt{2^N}} \sum_{n=1}^{2^N} |\text{solution(n)}\rangle$$
$$= \left(2^{N/2}\right)^{-1} \sum_s |s\rangle$$

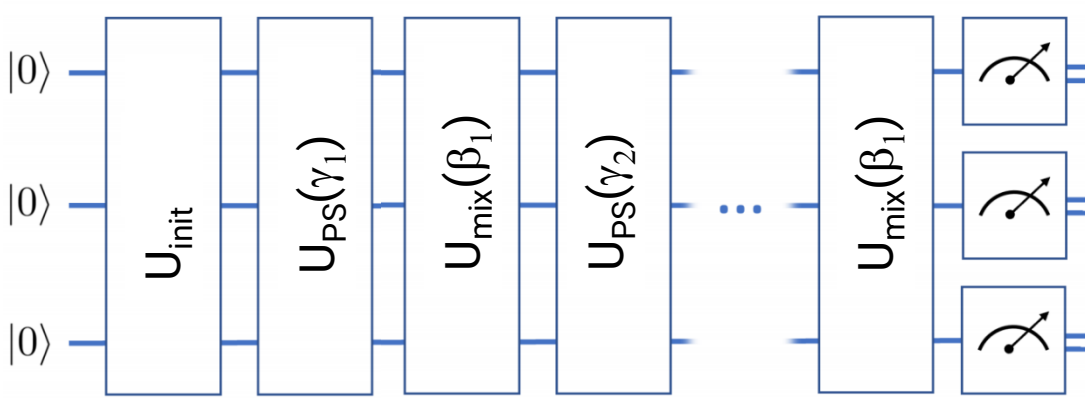
$$|\psi\rangle_{ps(1)} = \frac{1}{\sqrt{2^N}} \sum_{n=1}^{2^N} e^{iE_n} |solution(n)\rangle$$



You need to schedule the gates for every term of the objective function !

Quantum Approximate Optimization Algorithm

Example



Now if you measure, the probability of a bitstring depends both on γ and β in a non-linear way.
It is exponentially difficult to predict or simulate the probability $\left|\mathrm{B}_{2s}(\beta_1,\gamma_1,\beta_2,\gamma_2,\ldots,\beta_p,\gamma_p)\right|^2$ to find the optimal unknown solution s^*

$$|\psi\rangle_{\mathrm{QAOA}(p)} = \left(2^{N}/_{2}\right)^{-1}\sum_s \mathrm{B}_{2s}(\beta_1,\gamma_1,\beta_2,\gamma_2,\ldots,\beta_p,\gamma_p)e^{i\Gamma_{1s}(\beta_1,\gamma_1,\beta_2,\gamma_2,\ldots,\beta_p,\gamma_p)}|s\rangle$$

For $p \to \infty$ you can map this evolution to AQC; discrete becomes continuous; so, you know how to do it. For finite p there is currently not a lot of guidance, big sector of research.
The search over the parameter space γ and β is done heuristically (e.g., Gradient descent)

QAOA for Constrained Combinatorial Optimization

** Stay in the computational subspace!*

Associate one qubit to each q_i

Initialize the registers with a candidate solution found through, e.g., genetic algorithm or greedy search

$|\psi\rangle_{in}=|011010101100\rangle$

Mix the system by generating a superposition of the initial solution with all possible others (arbitrary parameter)

$|\psi\rangle_{mix}=\alpha|011010101100\rangle+\sum_k \beta_k|\phi_k\rangle$

Assign to each superposed solution a phase proportional (arbitrary parameter) to its objective function value

$|\psi\rangle_{mix}=\alpha e^{i\gamma E_{in}}|011010101100\rangle + \sum_k \beta_k e^{i\gamma E_k}|\phi_k\rangle$

Mix with new β^*

Add phases again with new γ

DIFFICULT

Only the logical subspace

VS

$|\psi\rangle_{mix}=\sum_s \beta_s|s\rangle$

All 2^N bitstrings

The problem of hard constraints

$$O_{\text{QUBO}}(q) = \sum_{i=1}^{N} a_i q_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} b_{ij} q_i q_j$$

$$\sum_i q_i = k \quad \xrightarrow{\text{Penalty Function}} \quad C\left(\sum_i q_i - k\right)^2$$

Difficult to scale, does not guarantee results, hardness is large softness

What you would want is to start from a classical bitstring, and then be able to "mix it" coherently in the subspace where the constraint is satisfied

Possible solution for these constraints: XY-Mixers.

Enforcing the same number of bits=1 is the same as doing two spin-flips

$$XY = s^+ s^- + s^- s^+$$

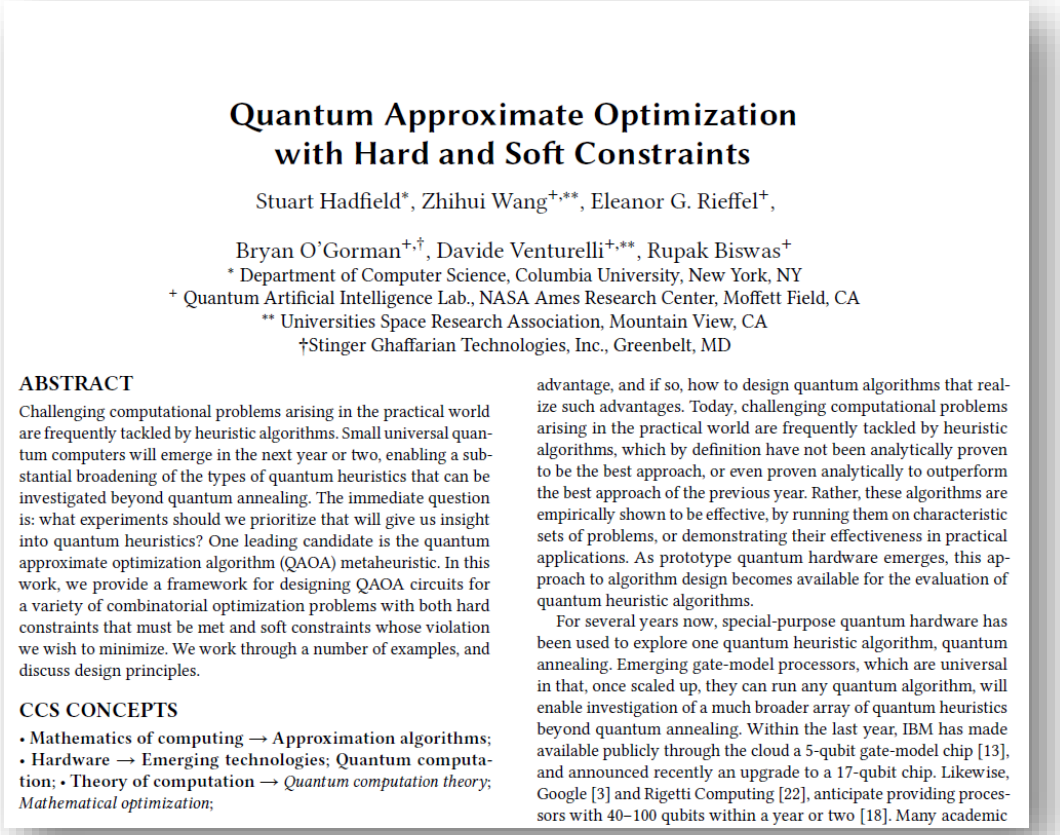
$$|001\rangle \xrightarrow{XY(2,3)} a|001\rangle + b|010\rangle \xrightarrow{XY(2,3)} a'|001\rangle + b'|010\rangle + c'|100\rangle$$

$$\text{XY}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\frac{\theta}{2} & i\sin\frac{\theta}{2} & 0 \\ 0 & i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

QAOA Applications

- Maximum Cut
- Max-SAT, Min-SAT, NAE-SAT
- Set Splitting
- MaxE3LIN2
- Max-ColorableSubgraph
- Graph Partitioning
- Maximum Bisection
- Max Vertex k-Cover
- MaxIndependentSet
- MaxClique
- MinVertexCover
- MaxSetPacking
- MinSetCover
- TSP
- SMS with various metrics and constraints

...



Quantum Approximate Optimization with Hard and Soft Constraints

Stuart Hadfield*, Zhihui Wang+,**, Eleanor G. Rieffel+,

Bryan O'Gorman+,†, Davide Venturelli+,**, Rupak Biswas+

* Department of Computer Science, Columbia University, New York, NY
+ Quantum Artificial Intelligence Lab., NASA Ames Research Center, Moffett Field, CA
** Universities Space Research Association, Mountain View, CA
†Stinger Ghaffarian Technologies, Inc., Greenbelt, MD

ABSTRACT

Challenging computational problems arising in the practical world are frequently tackled by heuristic algorithms. Small universal quantum computers will emerge in the next year or two, enabling a substantial broadening of the types of quantum heuristics that can be investigated beyond quantum annealing. The immediate question is: what experiments should we prioritize that will give us insight into quantum heuristics? One leading candidate is the quantum approximate optimization algorithm (QAOA) metaheuristic. In this work, we provide a framework for designing QAOA circuits for a variety of combinatorial optimization problems with both hard constraints that must be met and soft constraints whose violation we wish to minimize. We work through a number of examples, and discuss design principles.

CCS CONCEPTS

• Mathematics of computing → Approximation algorithms; • Hardware → Emerging technologies; Quantum computation; • Theory of computation → *Quantum computation theory*; *Mathematical optimization*;

advantage, and if so, how to design quantum algorithms that realize such advantages. Today, challenging computational problems arising in the practical world are frequently tackled by heuristic algorithms, which by definition have not been analytically proven to be the best approach, or even proven analytically to outperform the best approach of the previous year. Rather, these algorithms are empirically shown to be effective, by running them on characteristic sets of problems, or demonstrating their effectiveness in practical applications. As prototype quantum hardware emerges, this approach to algorithm design becomes available for the evaluation of quantum heuristic algorithms.

For several years now, special-purpose quantum hardware has been used to explore one quantum heuristic algorithm, quantum annealing. Emerging gate-model processors, which are universal in that, once scaled up, they can run any quantum algorithm, will enable investigation of a much broader array of quantum heuristics beyond quantum annealing. Within the last year, IBM has made available publicly through the cloud a 5-qubit gate-model chip [13], and announced recently an upgrade to a 17-qubit chip. Likewise, Google [3] and Rigetti Computing [22], anticipate providing processors with 40–100 qubits within a year or two [18]. Many academic

Objective Function: Soft Constraints
Feasible States: Hard Constraints

Alternating Operator Ansatz

$$Q_p(\boldsymbol{\beta},\boldsymbol{\gamma}) = U_{\mathrm{M}}(\beta_p)U_{\mathrm{P}}(\gamma_p)\cdots U_{\mathrm{M}}(\beta_1)U_{\mathrm{P}}(\gamma_1)$$

$$|\boldsymbol{\beta},\boldsymbol{\gamma}\rangle = Q_p(\boldsymbol{\beta},\boldsymbol{\gamma})\,|s\rangle$$

$$e^{-i\beta H_{\mathrm{M}}}$$

$$e^{-i\gamma H_{\mathrm{P}}}$$

Some initial state respecting:

- It is a superposition of several solutions in the feasible subspace
- It can be prepared efficiently

Some unitary respecting:

- Preserve the feasible subspace
- Provide all-to-all nonzero transitions between all feasible states
- Non-necessarily time evolution of a local Hamiltonian

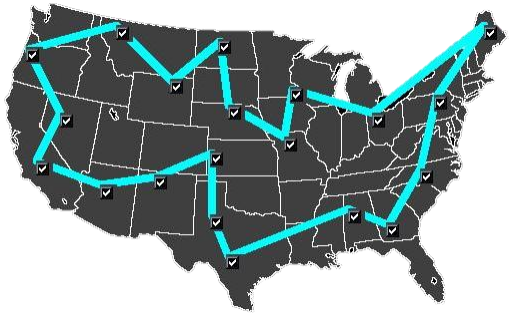
Some unitary respecting:

- Is diagonal in the computational basis
- The spectrum of H_P encodes the objective function

$$H_f\,|\mathbf{x}\rangle = f(\mathbf{x})\,|\mathbf{x}\rangle$$

Example for Mixers: TSP and Scheduling

In **traveling salesman** encoding



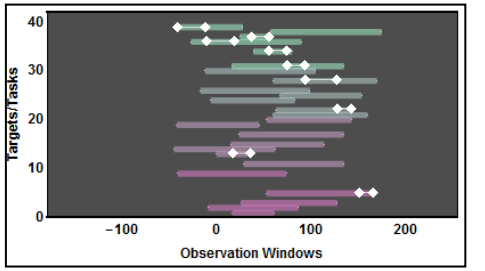
X_{vj}=1 if city v is visited as j^{th}

$$\sum_{\{u,v\}\in E} d_{u,v} \sum_{j=1}^{n} \left(x_{u,j}x_{v,j+1} + x_{v,j}x_{u,j+1}\right)$$

$$H^{(\mathrm{enc})}_{\mathrm{PS},\{i,j\},\{u,v\}} = S^{+}_{u,i}S^{+}_{v,j}S^{-}_{u,j}S^{-}_{v,i} + S^{-}_{u,i}S^{-}_{v,j}S^{+}_{u,j}S^{+}_{v,i},$$

(partitioned using edge coloring and parity
≈$(n-1)n^2/4$ mixers)
(needs to be repeated n(n-1)/2 times for all-to-all)

In **single machine scheduling**



X_{jt}=1 if job j starts at time t

$$C = \sum_{j} w_j \sum_{(d_j - p_j) < t < h} x_{j,t}(t + p_j - d_j)$$

$$H^{(\mathrm{enc})}_{\mathrm{TS},t,\{i,j\}} = S^{+}_{i,t+p_j}S^{+}_{j,t}S^{-}_{i,t}S^{-}_{j,t+p_i} + S^{+}_{i,t}S^{+}_{j,t+p_i}S^{-}_{i,t+p_j}S^{-}_{j,t}$$

(But if we add release dates then we need controls on the no-overlap constraint)

Zoology of Ansatze

(See Hadfield et al 2018 – «Quantum Alternating Operator Ansatz»)

Bitflip mixers
- Maximum Cut
- Max-SAT, Min-SAT, NAE-SAT
- Set Splitting
- MaxE3LIN2

...

Controlled Bitflip mixers
- MaxIndependentSet
- MaxClique
- MinVertexCover
- MaxSetPacking
- MinSetCover

...

XY mixers
- Max-ColorableSubgraph
- Graph Partitioning
- Maximum Bisection
- Max Vertex k-Cover

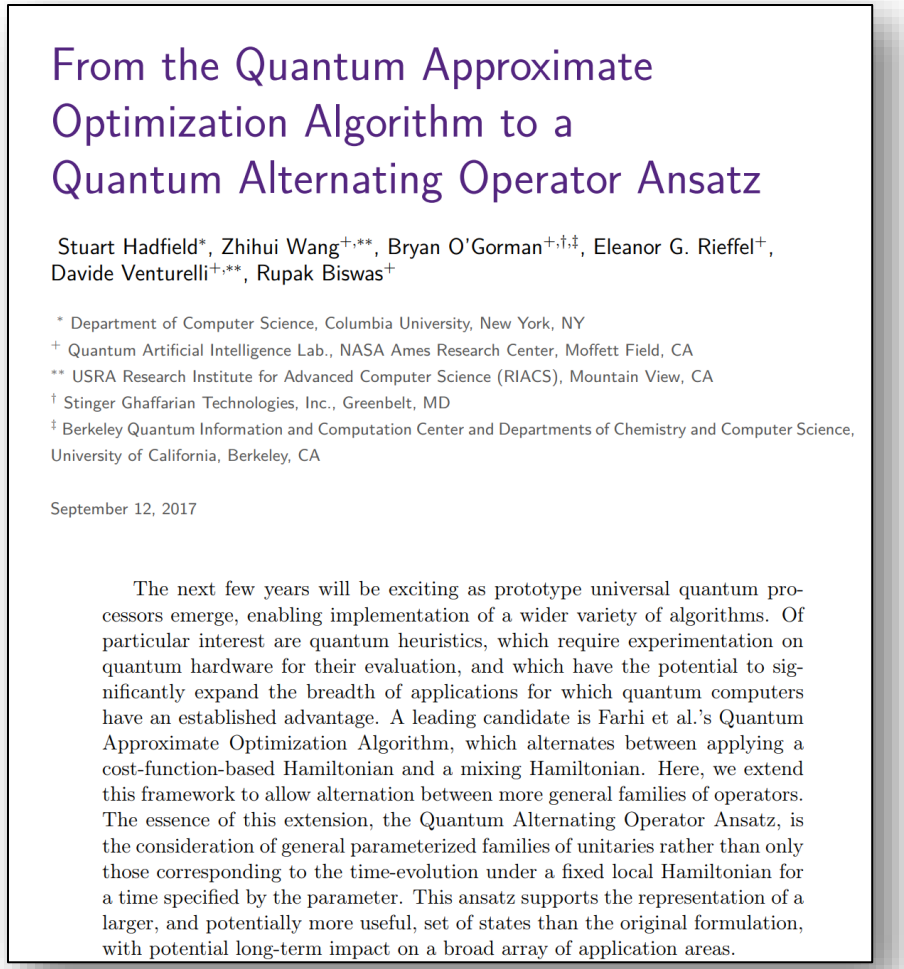
...

Controlled XY mixers
- Max-k-ColorableInducedSubgraph
- MinGraphColoring
- MinCliqueCover

...

Permutation mixers
- TSP
- SMS with various metrics and constraints

...



From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz

Stuart Hadfield*, Zhihui Wang+,**, Bryan O'Gorman+,†,‡, Eleanor G. Rieffel+, Davide Venturelli+,**, Rupak Biswas+

* Department of Computer Science, Columbia University, New York, NY
+ Quantum Artificial Intelligence Lab., NASA Ames Research Center, Moffett Field, CA
** USRA Research Institute for Advanced Computer Science (RIACS), Mountain View, CA
† Stinger Ghaffarian Technologies, Inc., Greenbelt, MD
‡ Berkeley Quantum Information and Computation Center and Departments of Chemistry and Computer Science, University of California, Berkeley, CA

September 12, 2017

The next few years will be exciting as prototype universal quantum processors emerge, enabling implementation of a wider variety of algorithms. Of particular interest are quantum heuristics, which require experimentation on quantum hardware for their evaluation, and which have the potential to significantly expand the breadth of applications for which quantum computers have an established advantage. A leading candidate is Farhi et al.'s Quantum Approximate Optimization Algorithm, which alternates between applying a cost-function-based Hamiltonian and a mixing Hamiltonian. Here, we extend this framework to allow alternation between more general families of operators. The essence of this extension, the Quantum Alternating Operator Ansatz, is the consideration of general parameterized families of unitaries rather than only those corresponding to the time-evolution under a fixed local Hamiltonian for a time specified by the parameter. This ansatz supports the representation of a larger, and potentially more useful, set of states than the original formulation, with potential long-term impact on a broad array of application areas.

The Flexible Design of NISQ Optimization Algorithms

Vanilla QAOA (Fahri 2014) and the QAOAnsatz (Hadfield 2017) were just the start of the field of modern Quantum Optimization Approaches

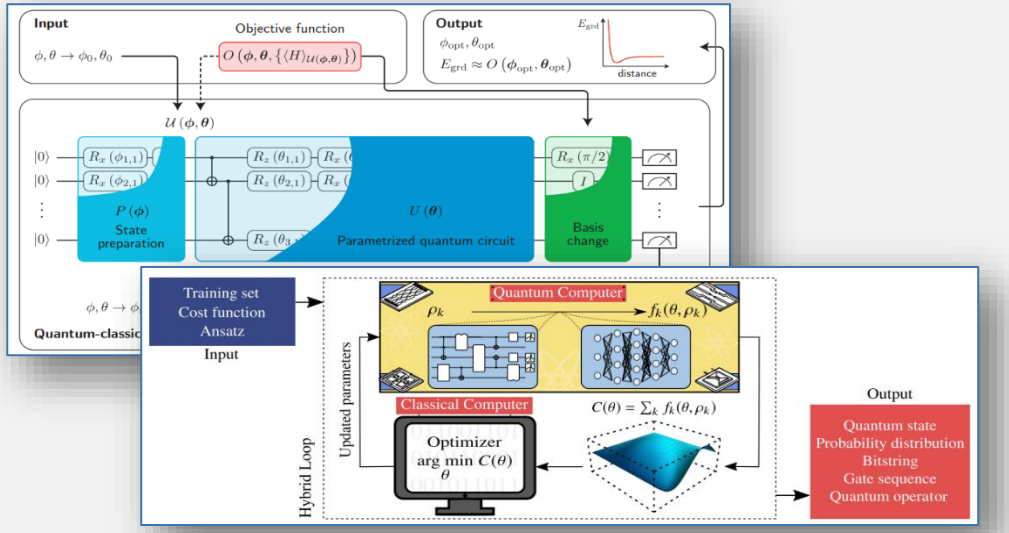
Variations:

- Incomplete/Approximate: e.g. mixing of a limited number of variables randomly selected.
- Adaptive: e.g. changing the circuit at runtime based on parameter exploration.
- Unstructured: e.g. the cost function could be evaluated only by classical hardware and is not in the ansatz, like learning in a neural network.
- Overparametrized: e.g. some gates might have offset angles
- Digital-Analog: i.e. global pulsing techniques that generate multi-qubit long range interactions.

Recent Review Articles:

Noisy intermediate-scale quantum (NISQ) algorithms
Bharti et al. (Jan 2021) – arXiv:2101.08448

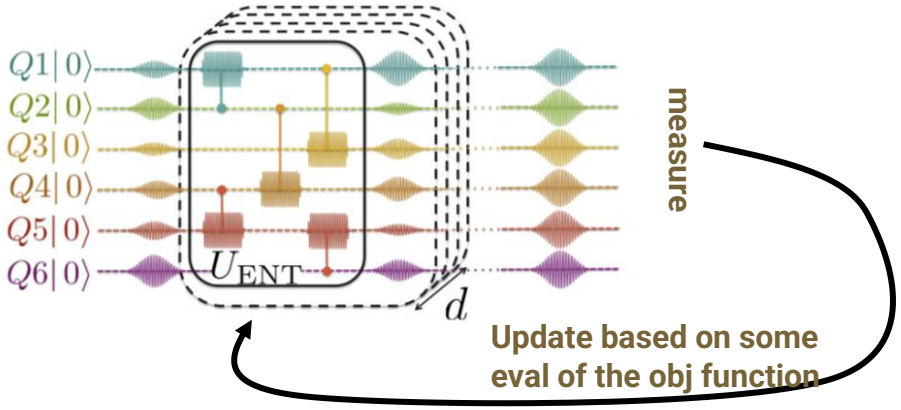
Variational Quantum Algorithms
Cerezo et al. (Dec 2020) – arXiv:2012.09265



The Flexible Design of NISQ Optimization Algorithms

"hardware-efficient ansatz" definition mostly from Kandala et al. in Nature 549(7661) 2017 for VQE: *we consider use the naturally available entangling interactions of the superconducting hardware.*
Applied to QAOA MaxCut (e.g. Moll et al. Quantum Sci. Technolog. 3(030503) 2018) but not benchmarked systematically as an option for classical optimization.

Benefit: reduce compilation overhead (synthesis and routing)
Malus: why would it work?



Objective function in the PS ansatz (AQC-inspired) and in the parameter setting discovery

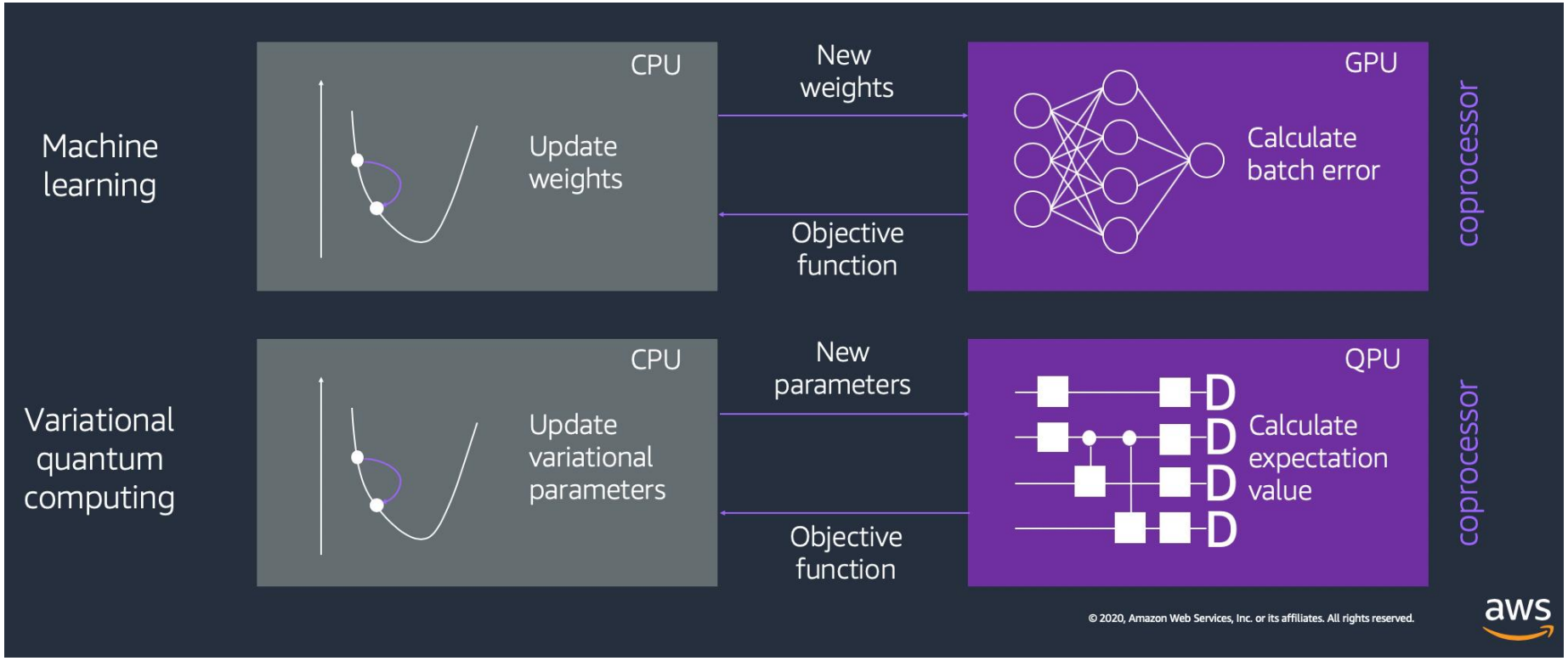
VS.

Objective function only in the parameter setting discovery.

Can we design a QAOA-Like algorithm with smaller compilation overhead via a prescriptive method for any combinatorial problem, still having guidance on why it should work?

Variational Quantum Computing

- Near term processors are used as co-processors

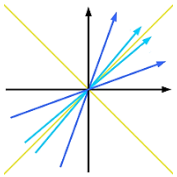


The Gate Synthesis Problem

Quantum Circuits can be composed by single and two-qubit gates of universal set*
CNOT, $R_y(q)$ and $R_z(a)$

Barenco et al. (1995)
Kraus, Cirac (2001)
Vatan, Williams (2003)

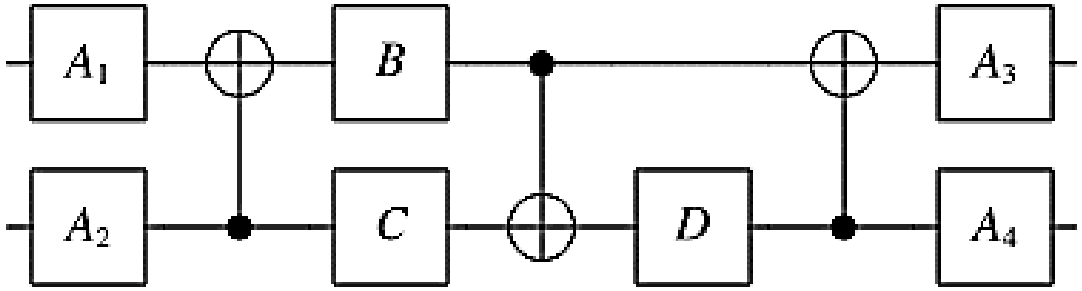
Each single qubit gate can be decomposed by single qubit rotations.
U1= $R_z(a)$ $R_y(b)$ $R_z(g)$ e^{if}



Each two qubit gate is reversible and it is representable by a Unitary Matrix.

R_Z gates can be «virtually» compiled.
(McKay 2017 and refs)

$U \equiv$



Maximum number of elementary 1-qubit gates: **15**
Maximum number of CNOTs: **3**
Maximum depth assuming R_Y, R_Z and simplifications: **11**

* active research to natively support multi-qubit gates

SWAP-Compilation (review)

Performance of algorithms in NISQ will depend on aspects such as gate fidelities, parallelization, idle time, crosstalks..

Different Metrics to optimize correlate to final performance:

- Total Quantum Factor
- Quantum Volume
- Number of Two-Qubit Gates
- **Makespan**

Guerreschi and Park (2018). Two-step approach to scheduling quantum circuits. *arXiv preprint arXiv:1708.00023*.

Khatri, Sumeet, et al. "Quantum assisted quantum compiling." *arXiv preprint arXiv:1807.00800* (2018).

Li, G., Ding, Y., & Xie, Y. (2018). Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices. *arXiv preprint arXiv:1809.02573 (2018)*.

Oddi, Angelo, and Riccardo Rasconi. "Greedy Randomized Search for Scalable Compilation of Quantum Circuits." *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer, Cham, (2018.

…

Example: MaxCut

$S_i=\pm 1$ Defines the cut

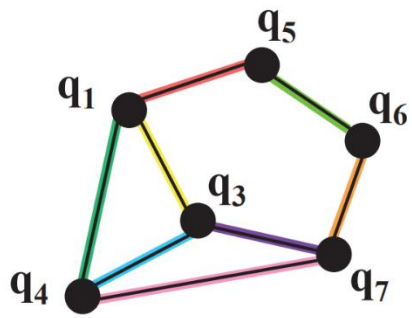
$$U = \frac{1}{2} \sum_{(i,j)\in E} (1 - s_i s_j)$$ Counts the edges in the cut

$\sum_i X_i$ Mixes the two partitions

$$U_{PS} = \prod_{jk} \exp(i\beta Z_j Z_k)$$

$$U_M = \prod_{j} \exp(i\gamma X_i)$$

Interaction graph obtained from quadratic objective function (MAXCUT)

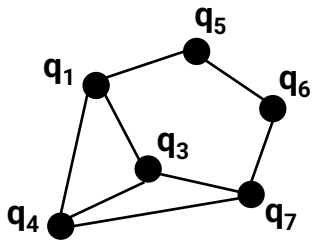


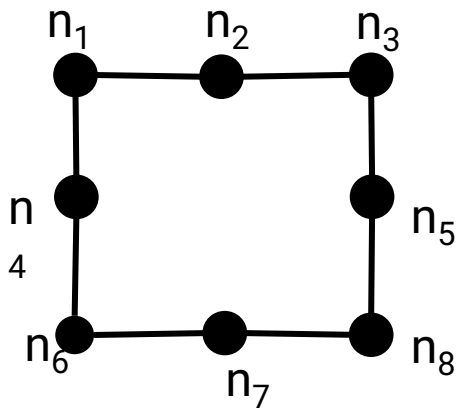
PS1	*MX*	*PS2*
P-S(q_1, q_4)	MIX(q_1)	P-S(q_1, q_4)
P-S(q_1, q_3)	MIX(q_3)	P-S(q_1, q_3)
P-S(q_3, q_4)	MIX(q_4)	P-S(q_3, q_4)
P-S(q_3, q_7)	MIX(q_5)	P-S(q_3, q_7)
P-S(q_4, q_7)	MIX(q_6)	P-S(q_4, q_7)
P-S(q_6, q_7)	MIX(q_7)	P-S(q_6, q_7)
P-S(q_5, q_6)		P-S(q_5, q_6)
P-S(q_1, q_5)		P-S(q_1, q_5)

- Every edge is a gate that needs to be executed (in arbitrary order)
- The same graph has to be executed multiple times (*p rounds*).
- Every qubit has to complete all the gates of round *p* before being involved in *p+1*

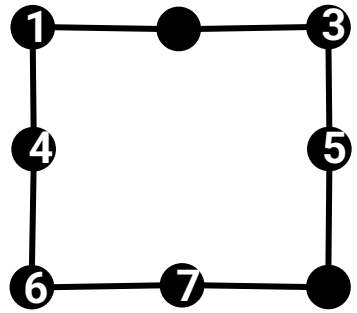
Circuit Execution Schedule

Interaction graph obtained from quadratic objective function (MAXCUT)



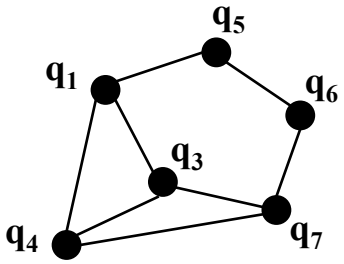


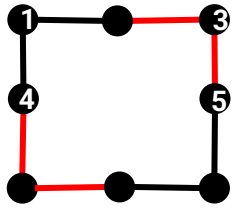
Initial assignment
$\mathbf{q_i \rightarrow n_i}$



- Every edge is a gate that needs to be executed (in arbitrary order)
- The same graph has to be executed multiple times (*p rounds*).
- Every qubit has to complete all the gates of round *p* before being involved in *p+1*

Circuit Execution Schedule

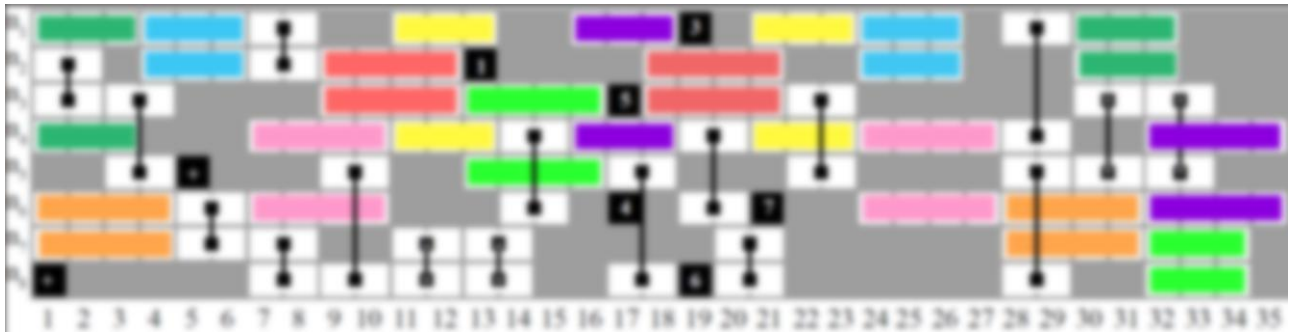




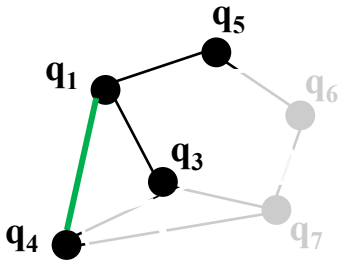
fast P-S (t=3)
slow P-S (t=4)
Swap (t=2)

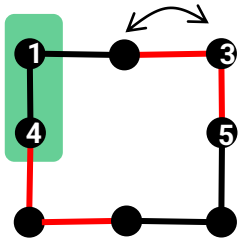
Benchmark presented at ICAPS17

Objective: finding the makespan-minimizing Gantt Schedule for p=1, p=2, N=8, N=21

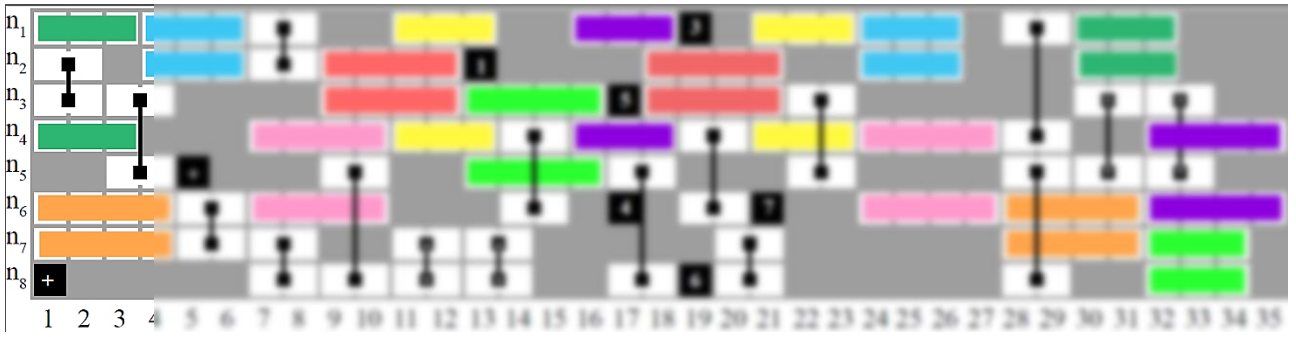


Circuit Execution Schedule

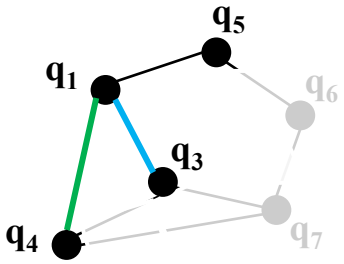


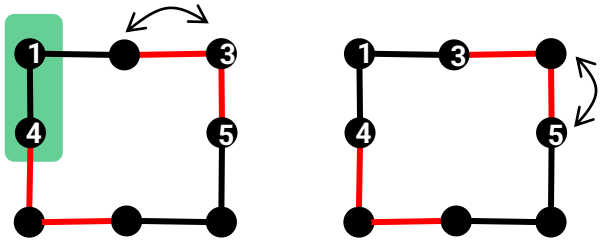


Q_{14}

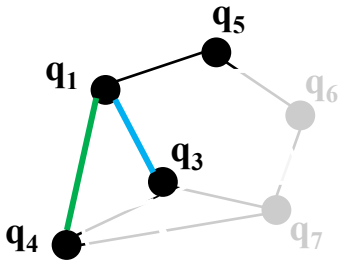


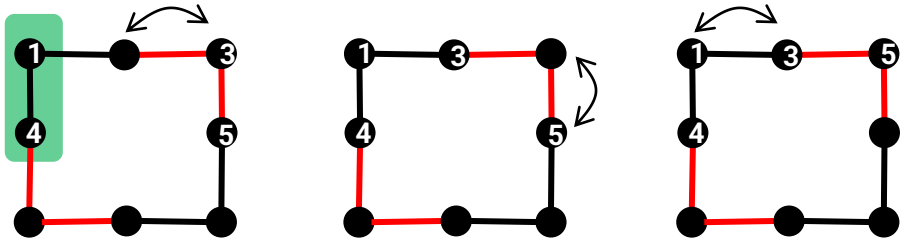
Circuit Execution Schedule



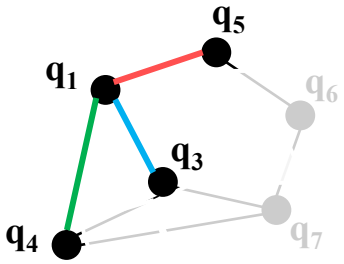


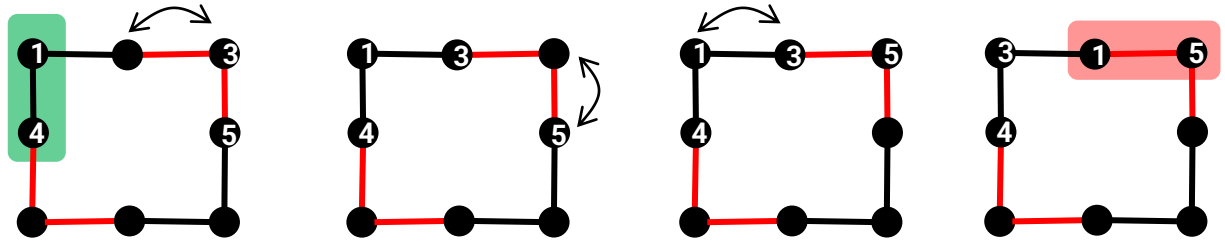
Circuit Execution Schedule

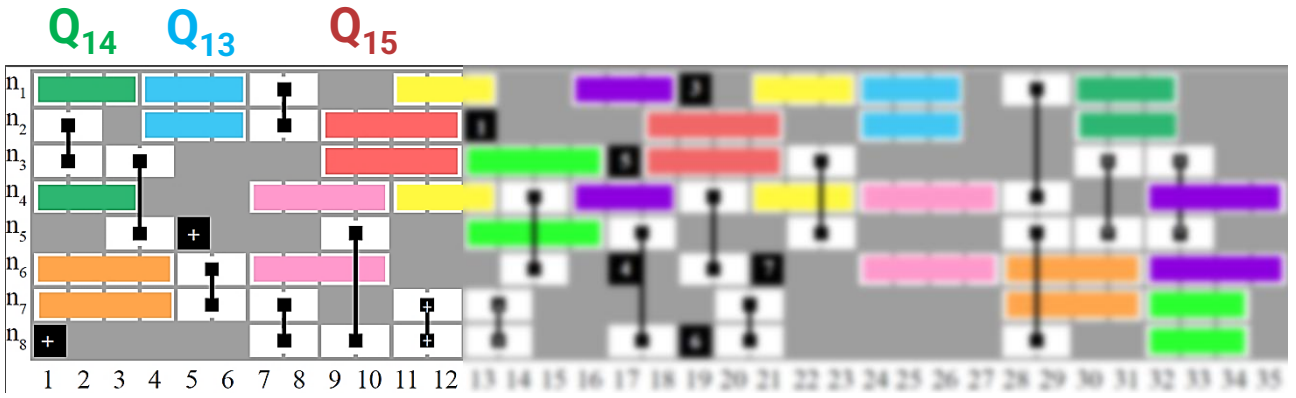




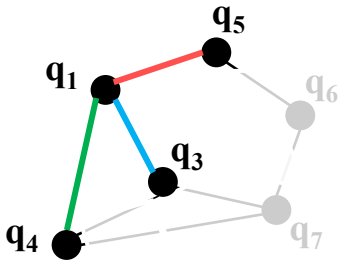
Circuit Execution Schedule

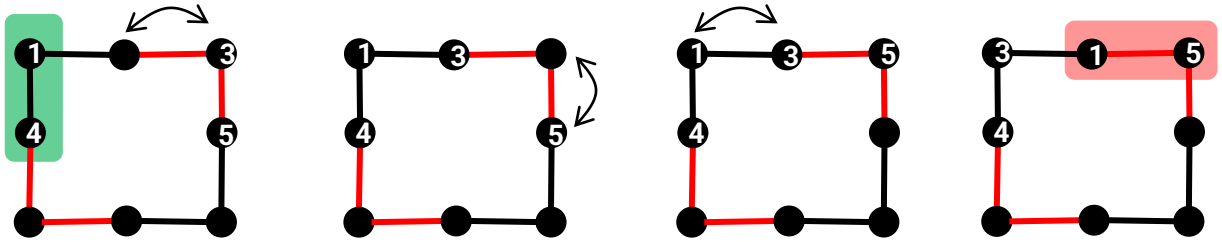




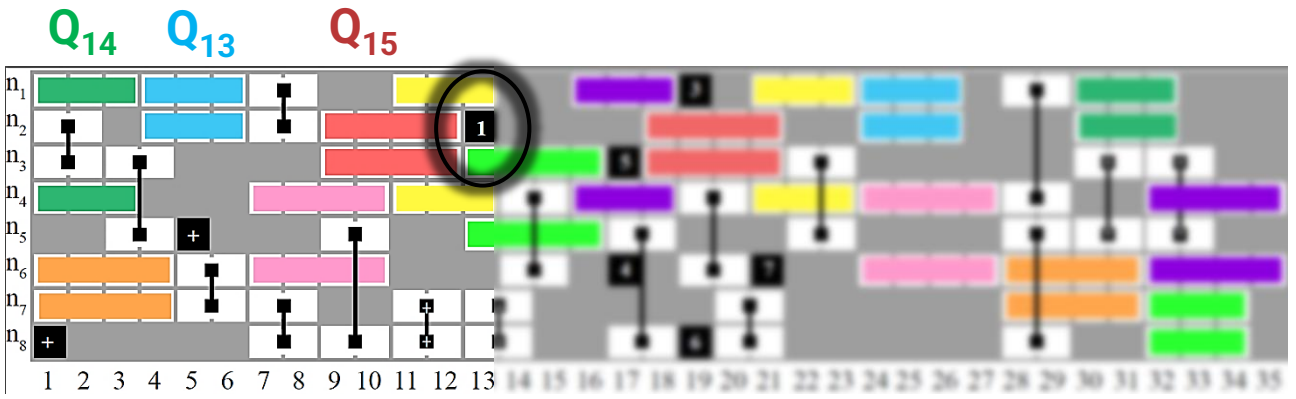


Circuit Execution Schedule

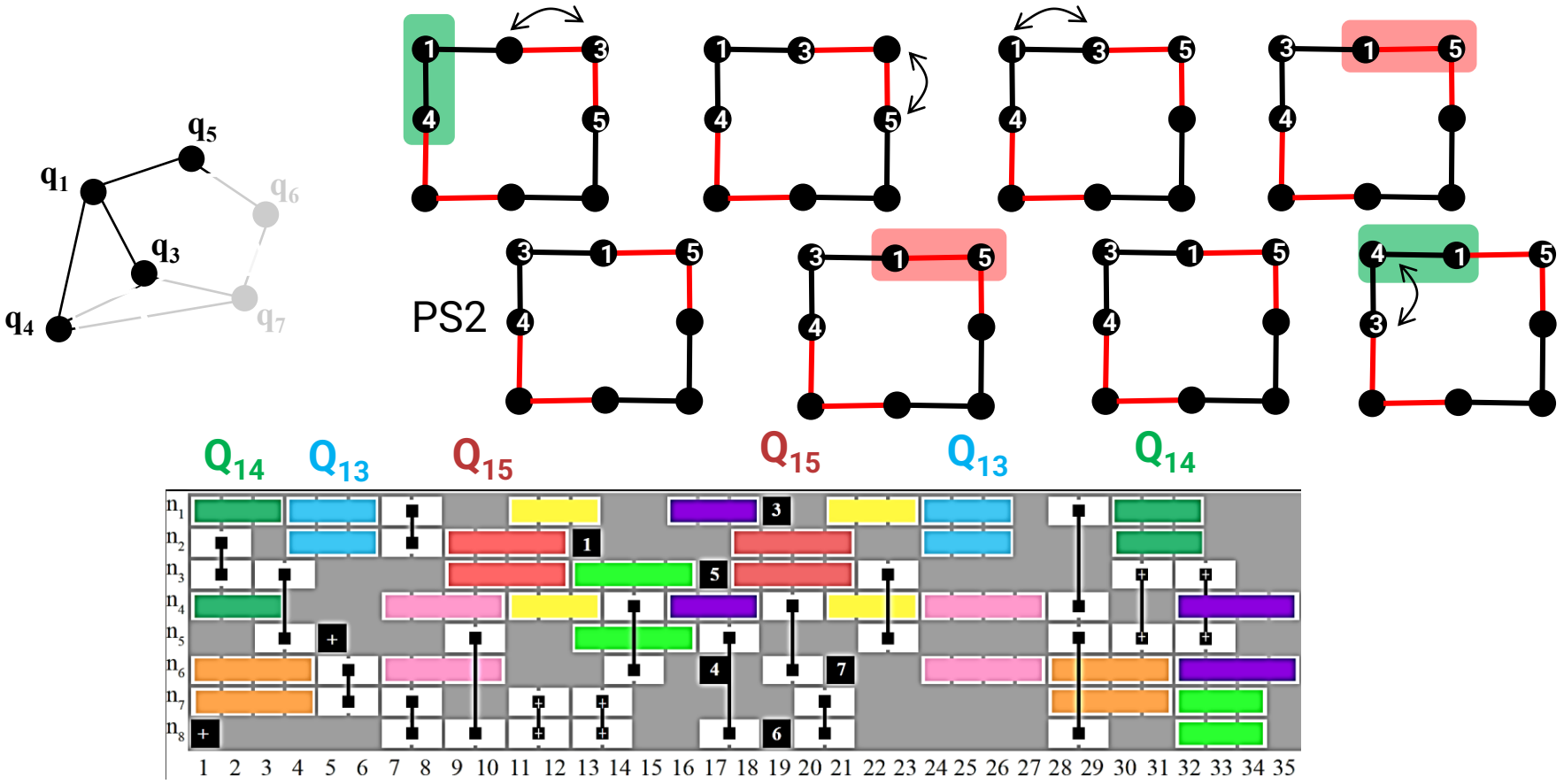




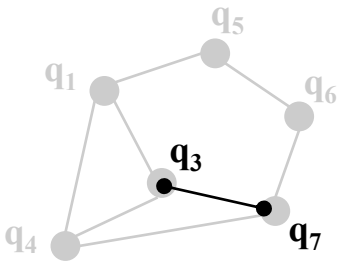
All actions of round 1 are completed – qubit can be mixed.
Qubit 1 can start participating to round 2.

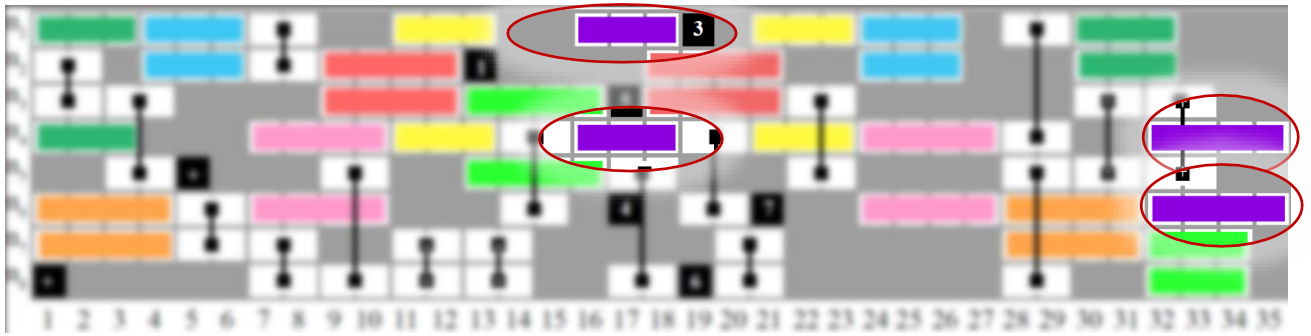


Circuit Execution Schedule

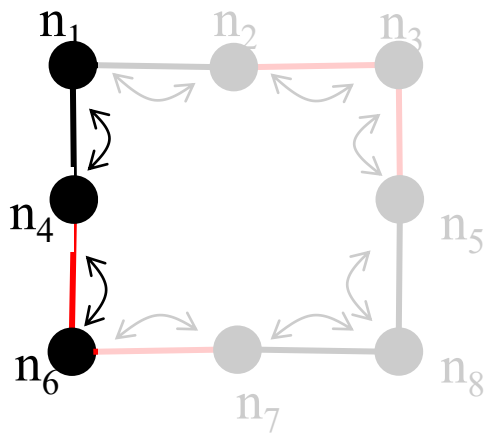


Circuit Execution Schedule





P-S(3,7) is fast on n_1, n_4
P-S(3,7) is slow on n_4, n_6



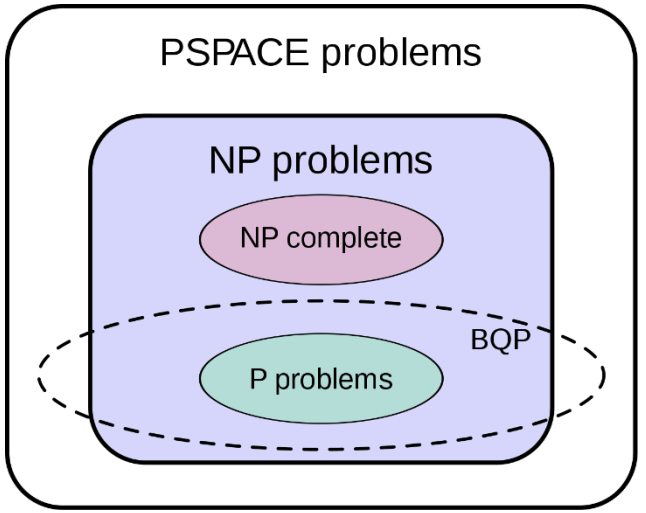
How to obtain these schedules efficiently?

Classical planning software is useful, and this is an active research field.

Complexity introduction

- A problem is said to belong to the complexity class BPP, bounded-error probabilistic polynomial time, if there is an algorithm that solves it such that
  - It is allowed to flip coins and make random decisions
  - It is guaranteed to run in polynomial time
  - On any given run of the algorithm, it has a probability of at most 1/3 of giving the wrong answer, whether the answer is YES or NO.

- There exists another complexity class called BQP, bounded-error quantum polynomial time, which is the quantum analogue of BPP
  - We hope that some problems belong to BQP and not to BPP to observe Quantum Advantage
    - E.g., Integer factorization



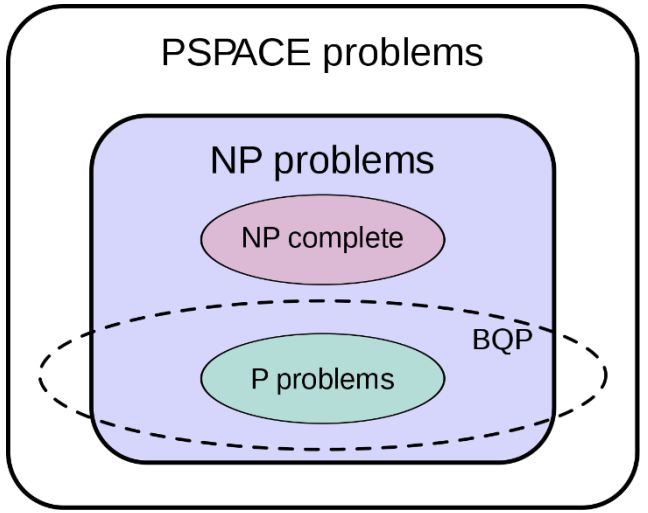
The suspected relationship of BQP to other problem spaces

[1] Michael Nielsen and Isaac Chuang (2000). Quantum Computation and Quantum Information. Cambridge: Cambridge University Press. ISBN 0-521-63503-9.

Optimization via Quantum Computing

- Many of these optimization problems cannot be solved “efficiently”
  - i.e., belong to NP-complete or NP-Hard complexity classes
  - Exponential speedups are believed as unreachable
    - Given application we still are encouraged to develop solution methods

- **QC can provide asymptotic speedup for certain optimization problems**
  - Usually quadratic speedup of exponentially expensive tasks
  - Hard to predict the scale at which this is observable
  - **No quantum advantage for practical optimization yet observed**
  - Practical and heuristic speedups are of interest!



The suspected relationship of BQP to other problem spaces

Retrieved from Michael Nielsen and Isaac Chuang (2000). Quantum Computation and Quantum Information

1. Bernal, D. E., Ajagekar, A., Harwood, S. M., Stober, S. T., Trenev, D., & You, F. (2022). Perspectives of quantum computing for chemical engineering. AIChE Journal, 68(6), e17651.

Optimization via Quantum Computing

Optimization problems presented from a mathematical programming perspective

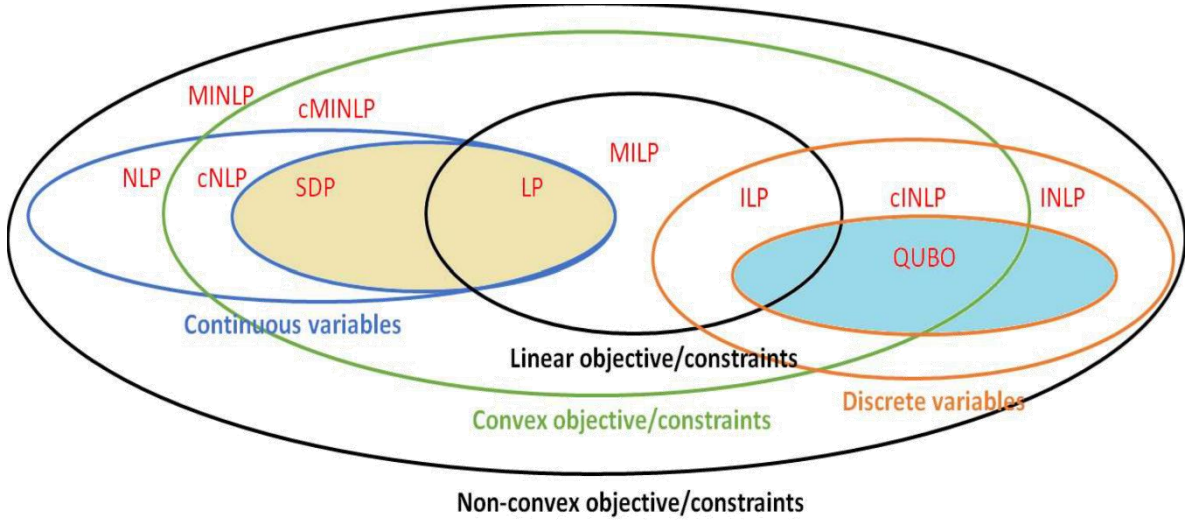
- Classified according to objective/constraints linearity/convexity and variables nature

Proposed algorithms with provable speedup for convex problems (LP/SDP)

- Most rely on Q Fourier transform, Q phase estimation, qRAM – Beyond NISQ capabilities

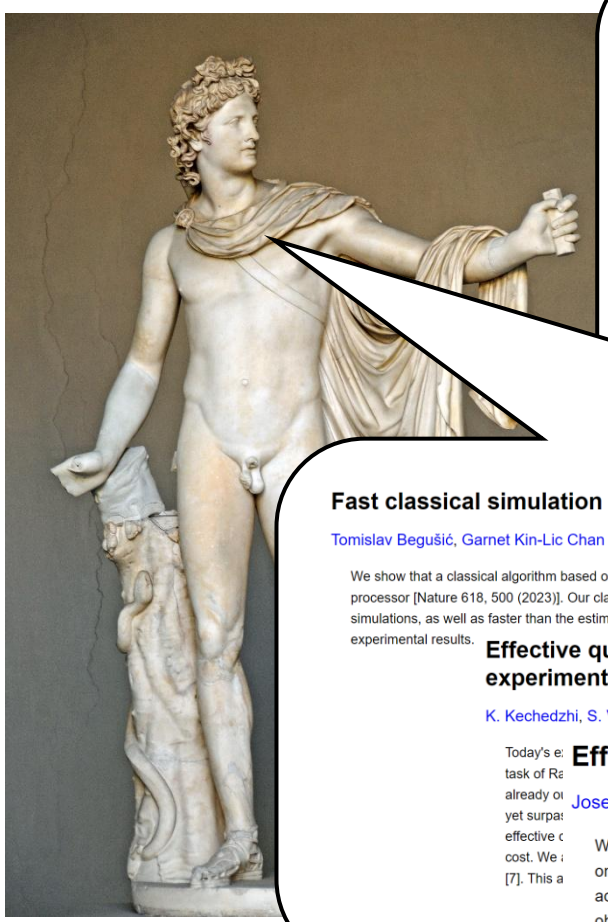
Heuristic/Approx methods for discrete optimization

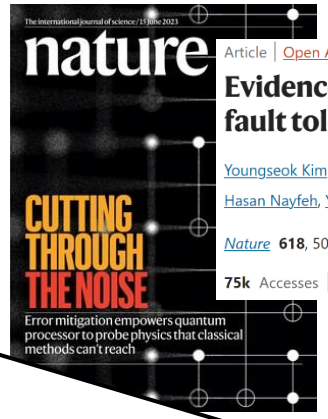
- Performance bounded by complexity results
  - **Unachievable exponential speedup (if P!=NP)**
  - **Complete enumeration quadratically faster via Grover search**
  - **Possible exploitation of structure in the problems**
- Several methods devised for Ising model ↔ Quadratic Unconstrained Binary Opt (QUBO)



1. Nannicini, Giacomo. "Fast quantum subroutines for the simplex method." Operations Research (2022).
2. Augustino, Brandon, et al. "Quantum interior point methods for semidefinite optimization." arXiv preprint arXiv:2112.06025 (2021).
3. Bernal, D. E., Ajagekar, A., Harwood, S. M., Stober, S. T., Trenev, D., & You, F. (2022). Perspectives of quantum computing for chemical engineering. AIChE Journal, 68(6), e17651.

Foreseeable future





Article | Open Access | Published: 14 June 2023

Evidence for the utility of quantum computing before fault tolerance

Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout van den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme & Abhinav Kandala

Nature **618**, 500–505 (2023) | Cite this article

75k Accesses | **1** Citations | **607** Altmetric | Metrics

Fast classical simulation of evidence for the utility of quantum computing before fault tolerance

Tomislav Begušić, Garnet Kin-Lic Chan

We show that a classical algorithm based on sparse Pauli dynamics can efficiently simulate quantum circuits studied in a recent experiment on 127 qubits of IBM's Eagle processor [Nature 618, 500 (2023)]. Our classical simulations on a single core of a laptop are orders of magnitude faster than the reported walltime of the quantum simulations, as well as faster than the estimated quantum hardware runtime without classical processing, and are in good agreement with the zero-noise extrapolated experimental results.

Effective quantum volume, fidelity and computational cost of noisy quantum processing experiments

K. Kechedzhi, S. V. Isakov, S. Mandrà, B. Villalonga, X. Mi, S. Boixo, V. Smelyanskiy

Today's e
task of Ra
already ou
yet surpas
effective c
cost. We a
[7]. This a

Efficient tensor network simulation of IBM's kicked Ising experiment

Joseph Tindall, Matt Fishman, Miles Stoudenmire, Dries Sels

We report an accurate, memory and time efficient classical simulation of a 127-qubit kicked Ising quantum system on the heavy-hexagon lattice. A simulation of this system on a quantum processor was recently performed using noise mitigation techniques to enhance accuracy (Nature volume 618, p. 500-505 (2023)). Here we show that, by adopting a tensor network approach that reflects the qubit connectivity of the device, we can perform a classical simulation that is significantly more accurate than the results obtained from the quantum device in the verifiable regime and comparable to the quantum simulation results for larger depths. The tensor network approach used will likely have broader applications for simulating the dynamics of quantum systems with tree-like correlations.

1. https://en.wikipedia.org/wi
2. https://en.wikipedia.org/wiki/Quantum_Mah
3. Youngseok, et al. "Evidence for the utility of quantum computing before fault tolerance." Nature 618.7965 (2023)
4. https://arxiv.org/abs/2306.15970

PURDUE UNIVERSITY

Quantum computing for operations research: current state and perspectives

David E. Bernal Neira

July 14, 2026

Principal Investigator of SECQUOIA

Assistant Professor - Davidson School of Chemical Engineering, Purdue University

International Federation of Operations Research Societies (IFORS) Meeting 2026

Vienna, Austria