Quantum combinatorial optimization

Daniel Egger
Senior Research Scientist
IBM Quantum

Course outline

Part 1: Considered problem classes

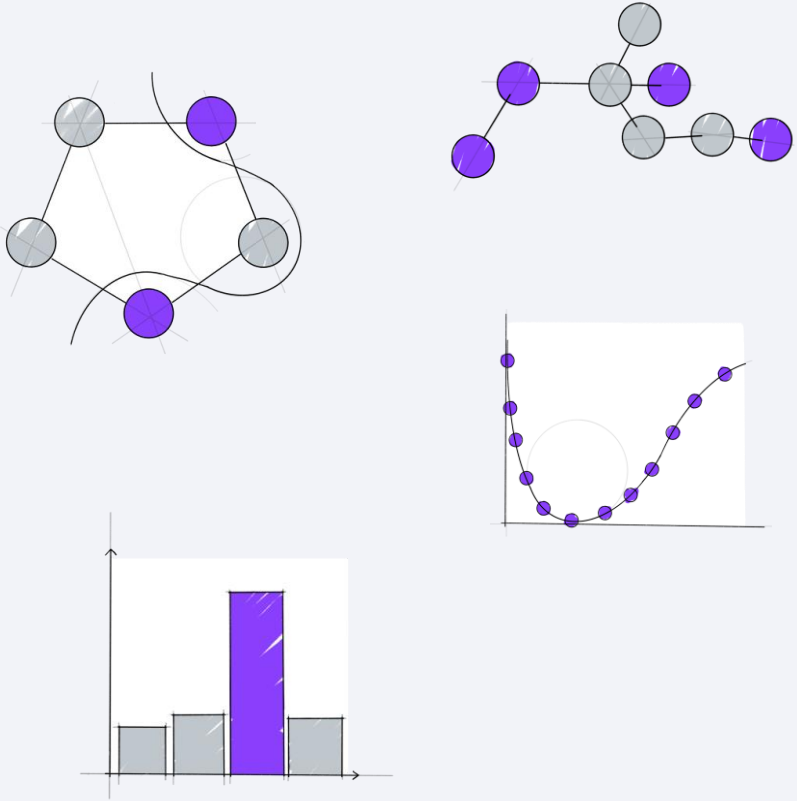
Part 2: From QUBO to Ising Hamiltonian

Part 3: The Adiabatic Theorem and Adiabatic Annealing

Part 4: From Adiabatic Annealing to QAOA

Part 5: Extensions of QAOA

Part 6: Executing QAOA on Superconducting Qubits



Course outline

Part 1: Considered problem classes

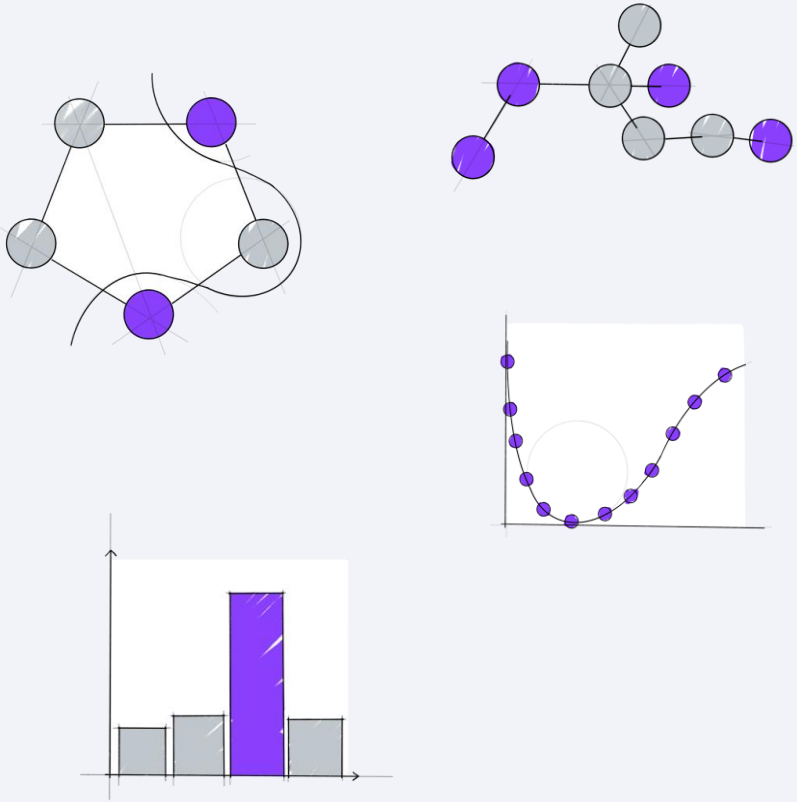
Part 2: From QUBO to Ising Hamiltonian

Part 3: The Adiabatic Theorem and Adiabatic Annealing

Part 4: From Adiabatic Annealing to QAOA

Part 5: Extensions of QAOA

Part 6: Executing QAOA on Superconducting Qubits

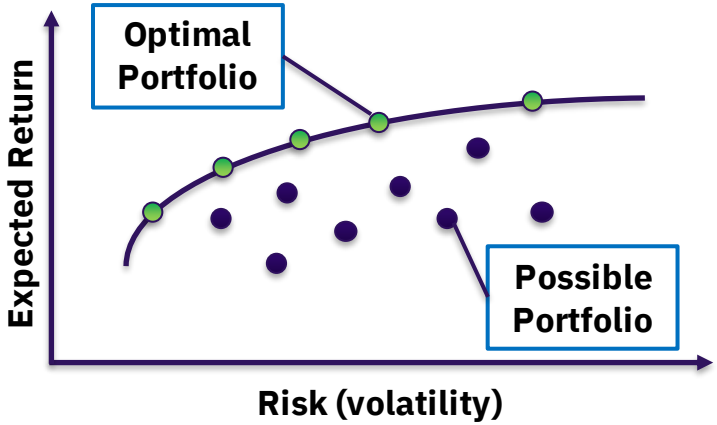


Example 1

Markowitz portfolio optimization
[Markowitz, J. Finance (1952)]

Goal

Select the best combination of n assets to **maximize return** and **minimize risk.**



Portfolio optimization has **many formulations** that depend on the context.

- Continuous weights: $\min_{w \in \mathbb{R}^n} w^T \Sigma w - q\mu^T w$ such that $\sum w_i = 1$
- Binary weights: $\min_{x \in \{0,1\}^n} x^T \Sigma x - q\mu^T x$ such that $\sum x_i = B$

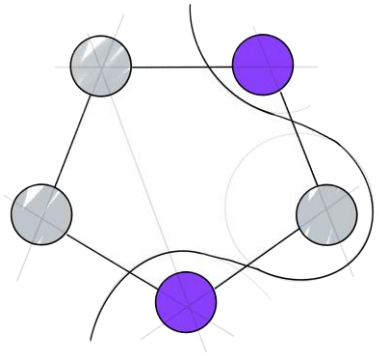
Mixed formulation: e.g., continuous weights but constraints introduce binary variables.

Example 2

Maximum Cut
(MaxCut)

Goal

Partition a graph in two sets of nodes such that the number of edges traversed by the cut is maximum.



- For a graph $G = (E, V)$ with $|V| = n$ nodes

$$\max_{x \in \{0,1\}^n} \sum_{(i,j) \in E} x_i(1 - x_j)$$

- Best known classical algorithm is the Goemans-Williamson algorithm which reaches an approximation ratio $0.87856 \leq \alpha \leq 0.87857$ in polynomial time by solving a semidefinite program relaxation.

An overview of optimization

Optimization

Continuous variables

Easy classically

Gradients & very efficient classical solvers

Integer variables and mixed continuous/integer

Classical

Methods

- Brute force (DFS, BFS)
- Warm-start
- Relaxations
  - $x_i \in \{0, 1\} \rightarrow x_i \in [0, 1]$
  - Semi-definite programming

Packages

- CPLEX
- Gurobi
- Etc.

Quantum methods

Noisy hardware — Fault tolerant

Methods (mostly heuristic)

- Approximate optimization (QAOA)
  [Farhi (2014), Bravyi (2020), Egger (2021), etc.]
- Adiabatic algorithms
  [Farhi (2021), Boixo (2009)]
- Quantum methods for simulated annealing
  [Szegedy (2004), Somma (2008), Lemieux (2020)]
- Etc.

Performance estimation & resource estimation

[Franca (2021), Weidenfeller (2022), Sanders (2020)]

Quadratic programs

A **quadratic program** is defined as

$$\begin{aligned} \text{Minimize} \quad & x^T Q_0 x + c^T x \\ \text{Subject to} \quad & Ax \leq b \\ & x^T Q_i x + a_i^T x \leq r_i \text{ for } 1, \ldots, i, \ldots, q \\ & l_i \leq x_i \leq u_i \text{ for } 1, \ldots, i, \ldots, n \end{aligned}$$

x are the decision variables which can be continuous, integer, or binary.

A QP also supports the "≥" and "=" constraints.

Quadratic Unconstrained Binary Optimization

- Binary decision variables
- Defined by

$$\min_{x\in\{0,1\}^n} x^T\Sigma x + b^T x$$

- Describes many types of problems (MaxCut, Travelling salesman, portfolio optimization, …)
- Covers NP-Complete problems
- Easy to transform a quadratic program into a QUBO

Equality constraints

The constraint

$$Ax = b$$

is recast to

$$\min_{x\in\{0,1\}^n} M(Ax-b)^2$$

Inequality constraints

The constraint

$$Ax \leq b$$

Is recast to

$$Ax - b + s = 0$$

& slack variable $s \geq 0$

Integer to binary conversion

Integer variables are converted to binary ones

$$0 \leq x \leq 7$$

Replaced by three binary variables

$$x_0, x_1, x_2 \in \{0,1\}$$

Summary

- Combinatorial optimization deals with problems like

$$\min_{x \in \{0,1\}^n} f(x),$$

  but more complex problem classes exists.
- We will focus on **quadratic problems** where $f(x) = x^T \Sigma x$.
- Equality and inequality constraints can be included in a quadratic problem formulation.

Course outline

Part 1: Considered problem classes

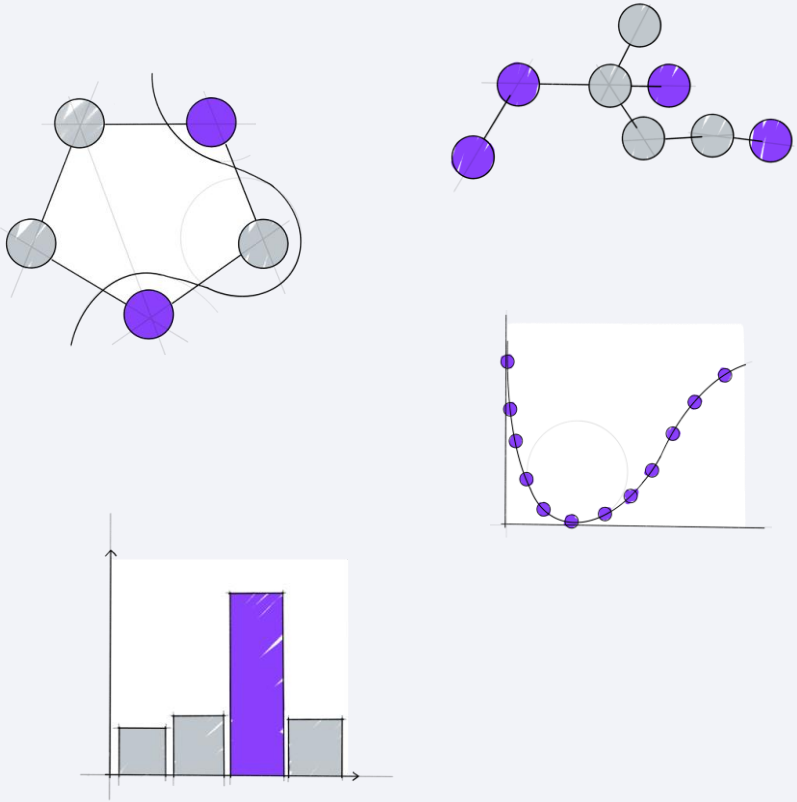
Part 2: From QUBO to Ising Hamiltonian

Part 3: The Adiabatic Theorem and Adiabatic Annealing

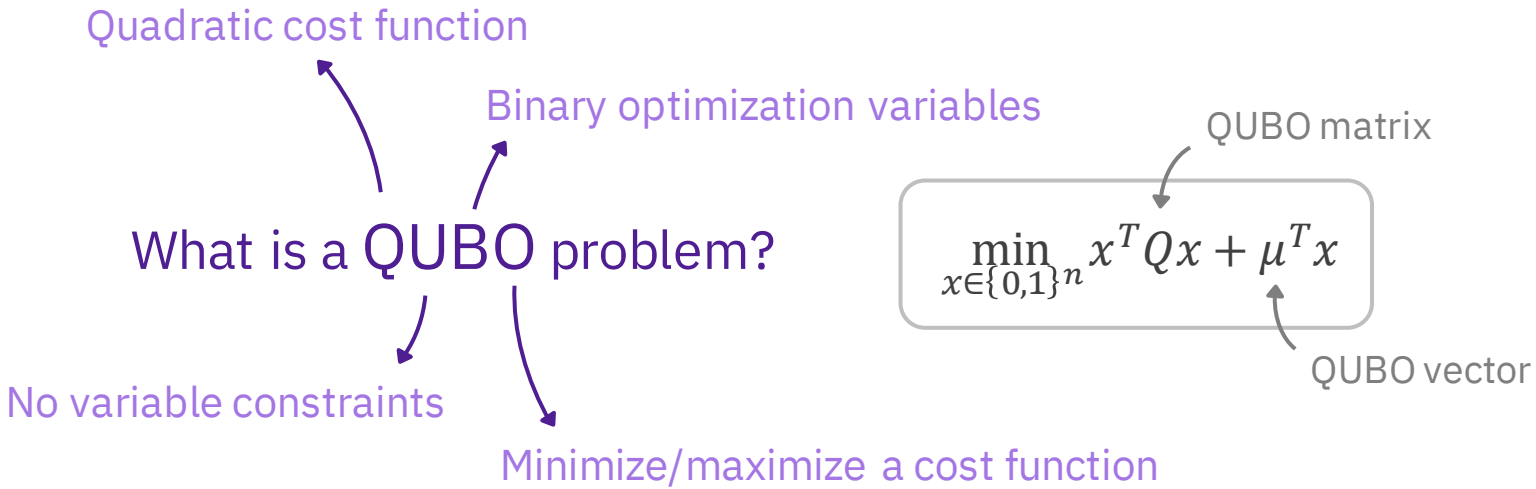
Part 4: From Adiabatic Annealing to QAOA

Part 5: Extensions of QAOA

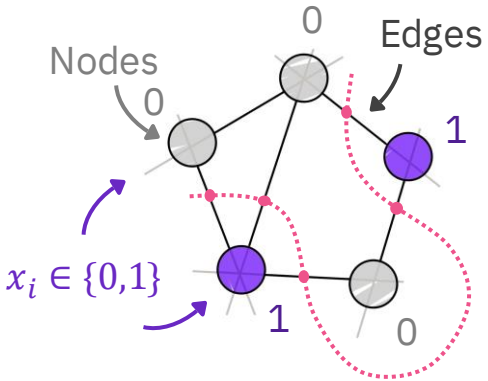
Part 6: Executing QAOA on Superconducting Qubits



Quadratic Unconstrained Binary Optimization (QUBO)

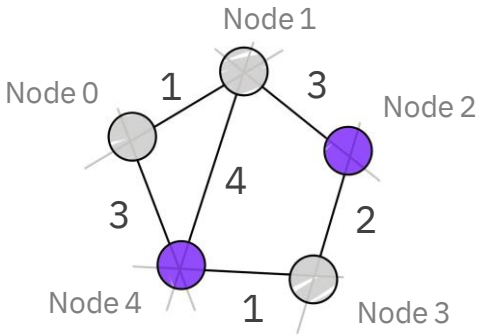


MaxCut: an example of a QUBO



Goal: to partition nodes into two groups such that the number of edges connecting nodes from the different groups is maximized

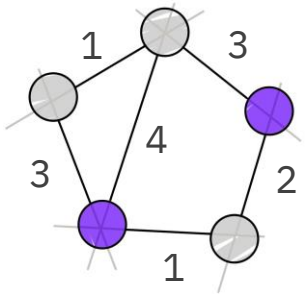
⇒ Leads to *binary* optimization variables, since we can assign each node a value of 0 or 1 depending on the group which they are on



We can also assign weights to the edges, which will give us our QUBO matrix, e.g.

$$w_{ij} = \begin{pmatrix} 0 & 1 & 0 & 0 & 3 \\ 1 & 0 & 3 & 0 & 4 \\ 0 & 3 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 3 & 4 & 0 & 1 & 0 \end{pmatrix}$$

MaxCut: an example of a QUBO



$$w_{ij} = \begin{pmatrix} 0 & 1 & 0 & 0 & 3 \\ 1 & 0 & 3 & 0 & 4 \\ 0 & 3 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 3 & 4 & 0 & 1 & 0 \end{pmatrix}$$

$$\min_{x \in \{0,1\}^n} x^T Q x + \mu^T x$$

Only edges between nodes of *different* groups contribute

$$C(x) = \sum_{i,j=1}^{n} w_{ij} x_i (1 - x_j)$$

$$= \sum_{i,j=1}^{n} w_{ij} x_i - \sum_{i,j=1}^{n} x_i w_{ij} x_j$$

$$\mu_i = \sum_{j=1}^{n} w_{ij} x_i$$

$$Q_{ij} = -w_{ij}$$

$$C(x) = \sum_{i,j=1}^{n} x_i Q_{ij} x_j + \sum_{i=1}^{n} \mu_i x_i$$

$$= x^T Q x + \mu^T x$$

The ground state problem

How can we solve QUBOs on quantum computers?

- Quantum computers require for us to formulate our problems as **Hamiltonian problems**.
- Hamiltonian: mathematical description of a physical system's **energy**, in terms of operators or matrices. For an eigenstate $|x\rangle$ of the system,

$$H|x\rangle = E_{|x\rangle}|x\rangle \qquad E_{|x\rangle} = \langle x|H|x\rangle$$

- *Goal*: to find a Hamiltonian, H_C, which **encodes our cost function**, $C(x)$.

$$H_C|x\rangle = C(x)|x\rangle$$

- If we can encode $C(x)$ in some H_C, we can then minimize it by finding the system's lowest energy state, i.e. the system's **ground state**

$$|x_{\text{optimal}}\rangle = \underset{|x\rangle \in \mathcal{H}}{\operatorname{argmin}} E_{|x\rangle}$$

Mapping to a Hamiltonian

So, how do we find H_C?

1. Map each of the optimization variables to a qubit using the substitution

$$x_i = \frac{1 - z_i}{2}, \qquad z_i \in \{-1, 1\}$$

2. Promote z_i to a Pauli spin operator Z_i, where $Z_i|x_i\rangle = (-1)^{x_i}|x_i\rangle$. This is equivalent to the above since $(-1)^{x_i} = 1 - 2x_i$ for $x_i \in \{0, 1\}$.

3. Make the substitution,

$$H_C = \sum_{ij} Q_{ij} \frac{\mathbb{1} - Z_i}{2} \frac{\mathbb{1} - Z_j}{2} + \sum_i \mu_i \frac{\mathbb{1} - Z_i}{2} = \sum_{ij} \frac{Q_{ij}}{4} - \sum_i \left(\sum_j \frac{Q_{ij}}{4} \right) Z_i - \overbrace{\sum_j \left(\sum_i \frac{Q_{ij}}{4} \right) Z_j}^{\substack{Q_{ij} \text{ symmetric} \\ i \leftrightarrow j}} + \sum_{ij} \frac{Q_{ij}}{4} Z_i Z_j + \sum_i \frac{\mu_i}{2} - \sum_i \frac{\mu_i}{2} Z_i =$$

$$= \underbrace{\sum_{ij} \frac{Q_{ij}}{4} Z_i Z_j - \sum_i \frac{1}{2} \left(\mu_i + \sum_j Q_{ij} \right) Z_i + \sum_{ij} \frac{Q_{ij}}{4} + \sum_i \frac{\mu_i}{2}}_{\text{This is an Ising Hamiltonian}}$$

Summary

- In combinatorial optimization, we often encounter **QUBO** problems.
- We have seen how we can map these types of problems to **Ising Hamiltonians**, such that we can solve them on quantum computers.
- Once we have encoded our cost function in a Hamiltonian, we can minimize it by solving the **ground state problem**, i.e. finding the ground state of our system.

Course outline

Part 1: Considered problem classes

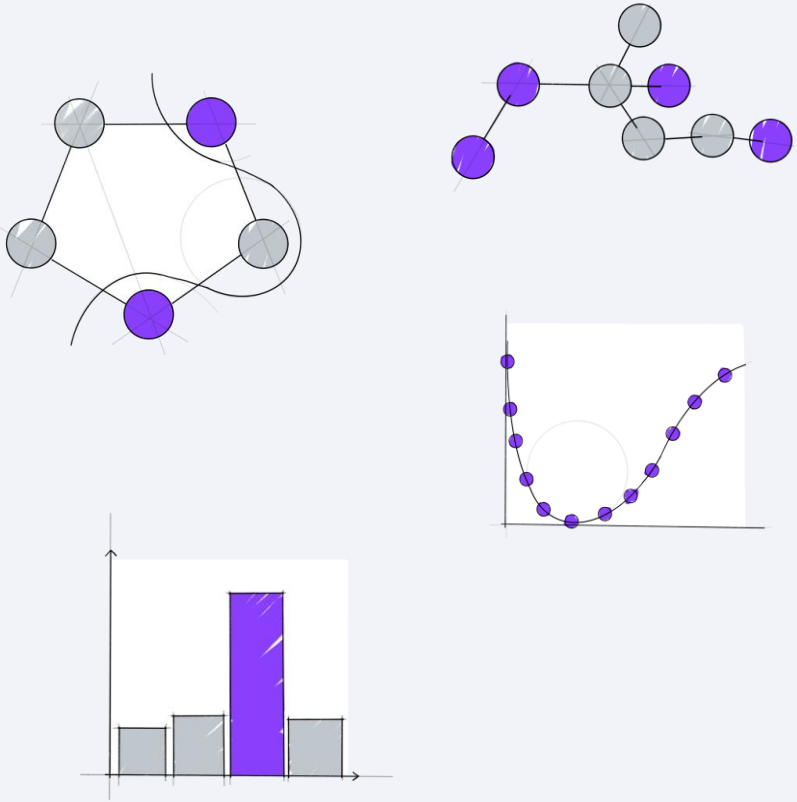
Part 2: From QUBO to Ising Hamiltonian

Part 3: The Adiabatic Theorem and Adiabatic Annealing

Part 4: From Adiabatic Annealing to QAOA

Part 5: Extensions of QAOA

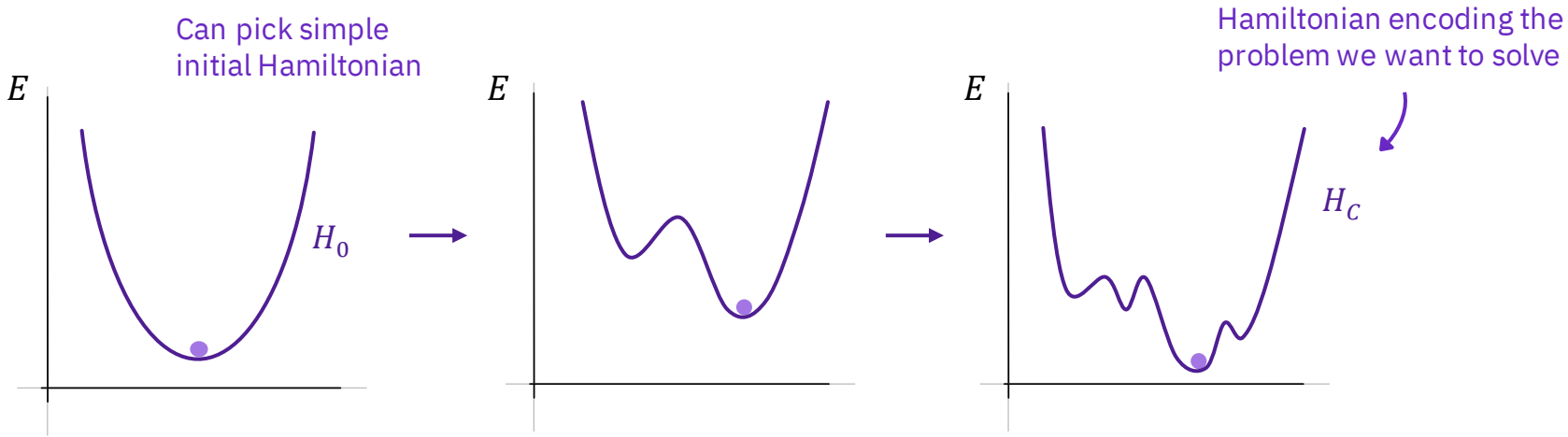
Part 6: Executing QAOA on Superconducting Qubits



Adiabatic Quantum Computing

The Adiabatic Theorem

If the Hamiltonian of a system which is in its ground state is perturbed slowly enough, the system will remain in its ground state.



Adiabatic Quantum Computing

Adiabatic Annealing

Tunable parameters

$$H(t) = \boxed{-\boldsymbol{A(t)} \sum_i X_i} + \boxed{\boldsymbol{B(t)} \sum_{i,j} Z_i Z_j}$$

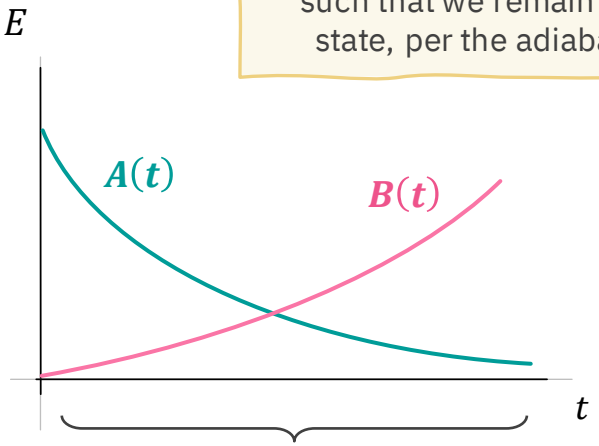
Initial Hamiltonian

Ground state $|+\rangle^{\otimes n}$ easily preparable

Problem Hamiltonian

(may have additional terms)

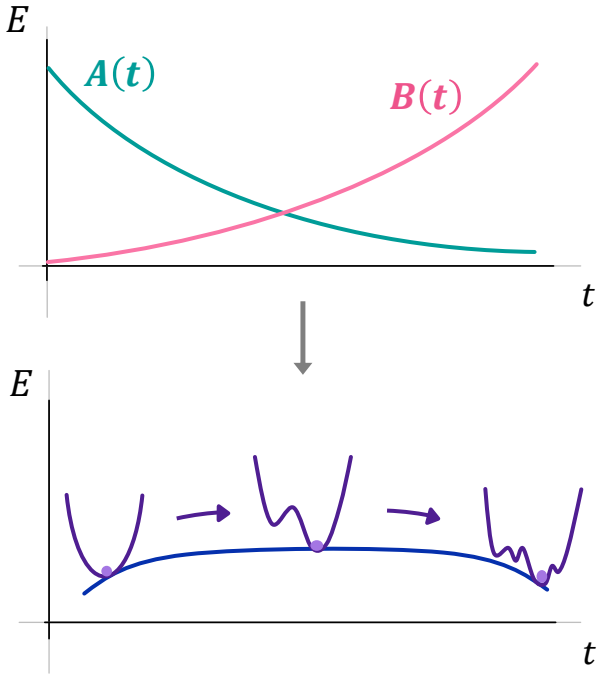
Goal: perform this tuning slow enough such that we remain in the ground state, per the adiabatic theorem



- *Why* does this transition need to be slow?
- *How slow* do we need to make this transition?

Adiabatic Quantum Computing

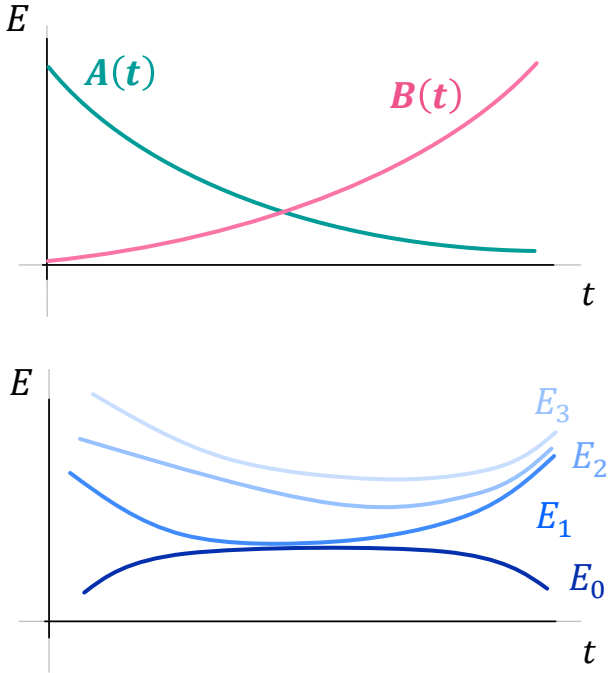
The critical region and the minimum gap problem



- As we perform the tuning of A and B in time, and $H(t)$ changes, we can trace the value of the **ground state energy**

Adiabatic Quantum Computing

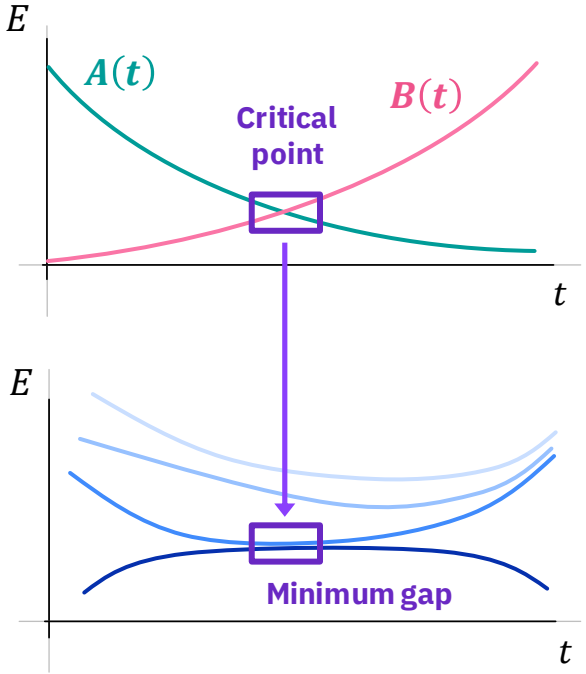
The critical region and the minimum gap problem



- As we perform the tuning of A and B in time, and $H(t)$ changes, we can trace the value of the **ground state energy**
- *However*, in our system, there isn't just the ground state, there also exist **excited states!**

Adiabatic Quantum Computing

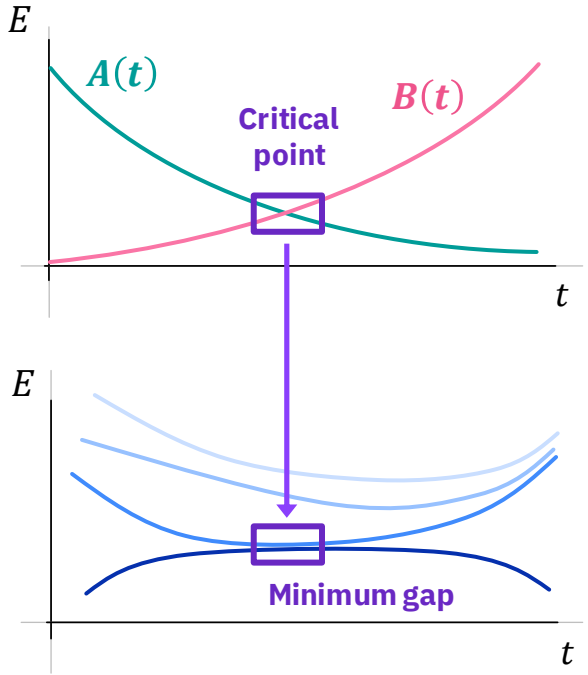
The critical region and the minimum gap problem



- As we perform the tuning of A and B in time, and $H(t)$ changes, we can trace the value of the **ground state energy**
- *However*, in our system, there isn't just the ground state, there also exist **excited states!**
- At the critical point, the gap between the ground state and the first excited state becomes small $\Longrightarrow$ **minimum gap**.

Adiabatic Quantum Computing

The critical region and the minimum gap problem



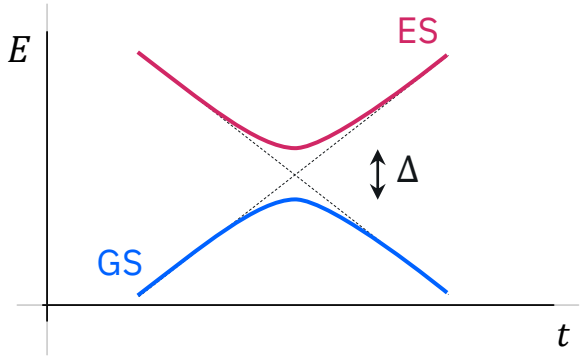
- As we perform the tuning of A and B in time, and $H(t)$ changes, we can trace the value of the **ground state energy**
- *However*, in our system, there isn't just the ground state, there also exist **excited states!**
- At the critical point, the gap between the ground state and the first excited state becomes small $\Longrightarrow$ **minimum gap**.

Problem!

- Due to quantum fluctuations, if we don't make the annealing slow enough, we will jump onto the excited state.
- Further, the size of the minimum gap decreases as we increase the system size, and the annealing time required may scale badly.

Adiabatic Quantum Computing

Landau Zener example



- Consider the Hamiltonian

$$H(t) = \frac{1}{2}\varepsilon(t)Z + \frac{1}{2}\Delta X$$

- We start in the ground state at $t = -\infty$.
- Then with the schedule $\varepsilon(t) = vt$ we are in the excited state at $t = \infty$ with probability

$$P = \exp\left(-\frac{\pi\Delta^2}{2v}\right)$$

- **To avoid transitions we need slower schedules for smaller gaps**. I.e., we remain adiabatic if $v \ll \Delta$.

Summary

- Adiabatic quantum computing allows us to **find the ground state of a Hamiltonian**. Start in an easy-to-prepare ground state and evolve towards the Hamiltonian of interest.

- The gap between the ground state and the first excited state in the instantaneous system may be (exponentially) small. This requires very slow annealing schedules.
  Small gaps $\Longrightarrow$ slow evolution.

Course outline

Part 1: Considered problem classes

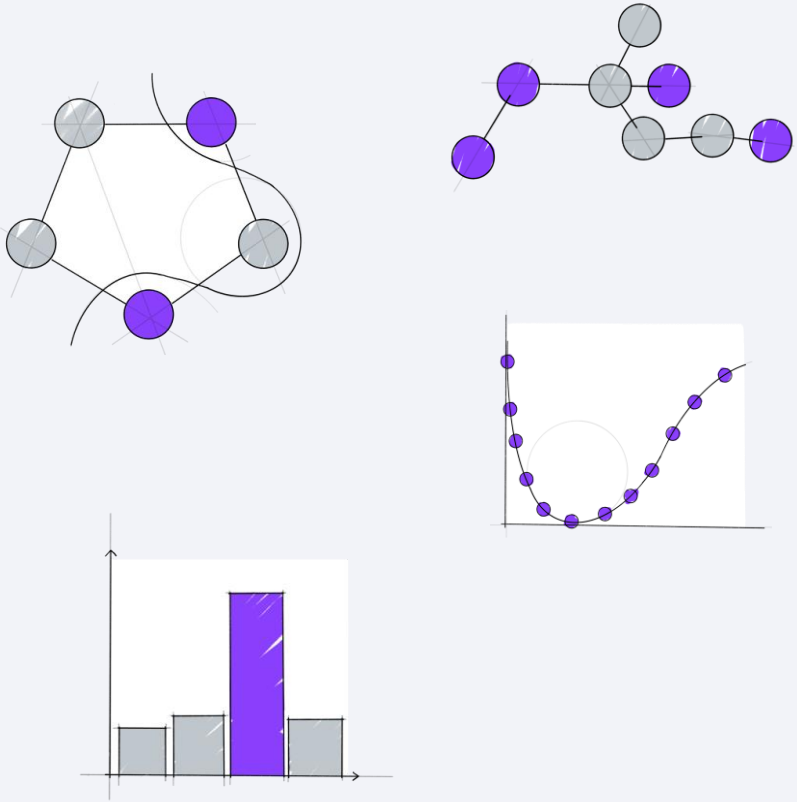
Part 2: From QUBO to Ising Hamiltonian

Part 3: The Adiabatic Theorem and Adiabatic Annealing

Part 4: From Adiabatic Annealing to QAOA

Part 5: Extensions of QAOA

Part 6: Executing QAOA on Superconducting Qubits



The Quantum Approximate Optimization Algorithm (QAOA): an overview

- First introduced in 2014 by Edward Farhi, Jeffrey Goldstone and Sam Gutmann
- Solves **QUBO** problems
- Is a **variational** algorithm and can be considered a special case of VQE
- Has its variational form based on the **trotterization** of quantum **adiabatic annealing**

Trotterization

Time evolution

As well as describing the energy of quantum systems, Hamiltonians also represent the **time evolution** of these systems. We know this because of the **Schrödinger equation**,

$$H\,|\psi(t)\rangle \;=\; i\frac{\partial}{\partial t}\,|\psi(t)\rangle$$

Time evolution of quantum states is then described as

$$|\psi(t)\rangle = \; U_t|\psi(0)\rangle$$

Trotterization

Time evolution

As well as describing the energy of quantum systems, Hamiltonians also represent the **time evolution** of these systems. We know this because of the **Schrödinger equation**,

$$H\,|\psi(t)\rangle \;=\; i\frac{\partial}{\partial t}\,|\psi(t)\rangle$$

Time evolution of quantum states is then described as

$$|\psi(t)\rangle = \boldsymbol{U_t}|\psi(0)\rangle$$

Time-independent $\boldsymbol{H}$: $U_t = e^{-iHt}$

Time-dependent $\boldsymbol{H(t)}$: $U_t = \mathcal{T}\exp\left(-i\int_0^t dt'\,H(t')\right)$

Trotterization

$$H(t) = \underbrace{-A(t)\sum_i X_i}_{\boldsymbol{H(0) = H_M}} + \underbrace{B(t)\sum_{i,j} Z_i Z_j}_{\boldsymbol{H_C}}$$

"Trotterizing" the Hamiltonian

Recall our adiabatic annealing Hamiltonian, which was **time-dependent** and composed of two **non-commuting** terms. We will try to simplify this Hamiltonian by dividing it into sequential pieces, for which we need two approximations.

$$U_t = \mathcal{T}\exp\left(i\int_0^t dt' H(t')\right)$$

1. Discretization

$$\int_0^t dt' H(t') \;\rightarrow\; \sum_{n=0}^{N} \Delta t\, H(n\Delta t) \quad\Rightarrow\quad |\psi(t)\rangle \approx e^{-i\Delta t\cdot H(N\Delta t)} \cdots e^{-i\Delta t\cdot H(2\Delta t)}\, e^{-i\Delta t\cdot H(\Delta t)}\, |\psi(0)\rangle$$

$$H = H_M + H_C$$

2. Suzuki-Trotter expansion

Recall $e^{(A+B)} \neq e^A e^B$ for non-commuting A, B $\quad\Rightarrow\quad$ $e^{\Delta t(A+B)} = e^{\Delta t\cdot A} e^{\Delta t\cdot B} + O(\Delta t^2)$

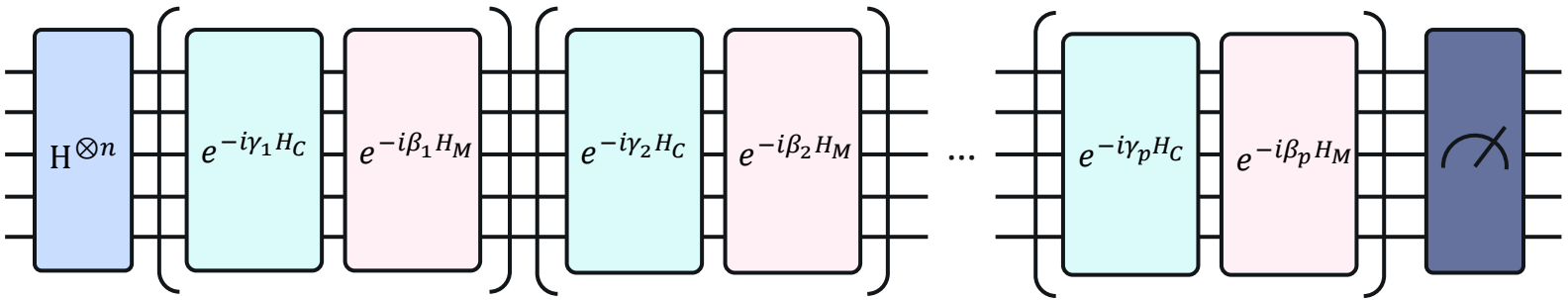
From the trotterized Hamiltonian to QAOA

Steps 1 and 2 of the Trotterization result in an approximate state:

$$|\psi(t)\rangle \approx \left[e^{-i\Delta t\cdot H_M(N\Delta t)}e^{-i\Delta t\cdot H_C(N\Delta t)}\right]\ldots\left[e^{-i\Delta t\cdot H_M(2\Delta t)}e^{-i\Delta t\cdot H_C(2\Delta t)}\right]\cdot\left[e^{-i\Delta t\cdot H_M(\Delta t)}e^{-i\Delta t\cdot H_C(\Delta t)}\right]|\psi(0)\rangle$$

We have effectively divided the Hamiltonian into “layers”, each with two components.

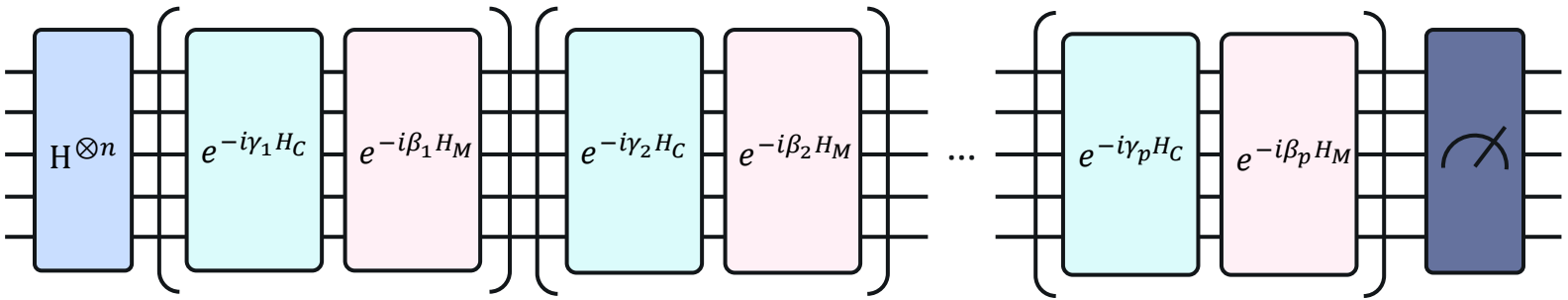
QAOA goes one step further and makes the time-steps in each layer optimization parameters $\Longrightarrow \beta_i,\gamma_i$.



The QAOA protocol

The quantum circuit

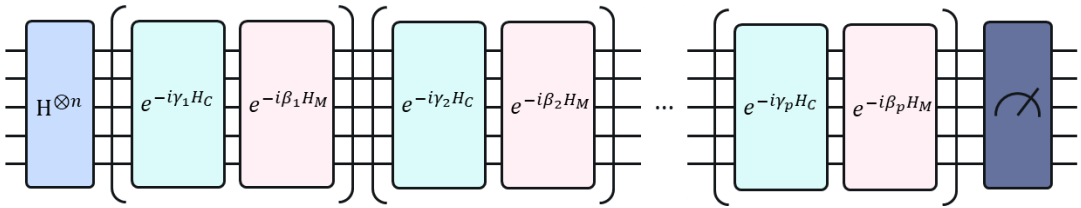
1. Define $H_M = -\sum_i X_i$ and H_C such that it encodes the cost function $f(x)$ of the optimization problem
2. Apply Hadamard gates to all qubits to begin in the ground state of H_M, i.e. $|+\rangle^{\otimes n}$
3. Apply p layers, each consisting of two terms, $e^{-i\gamma_k H_M}$ and $e^{-i\beta_k H_C}$ for $\mathrm{k} = 1, \ldots, p$
4. Measure



The QAOA protocol

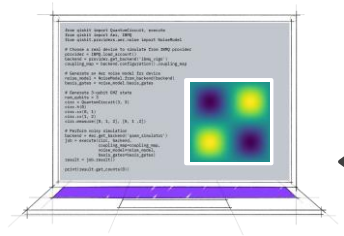
The variational approach

1-4. Construct the circuit



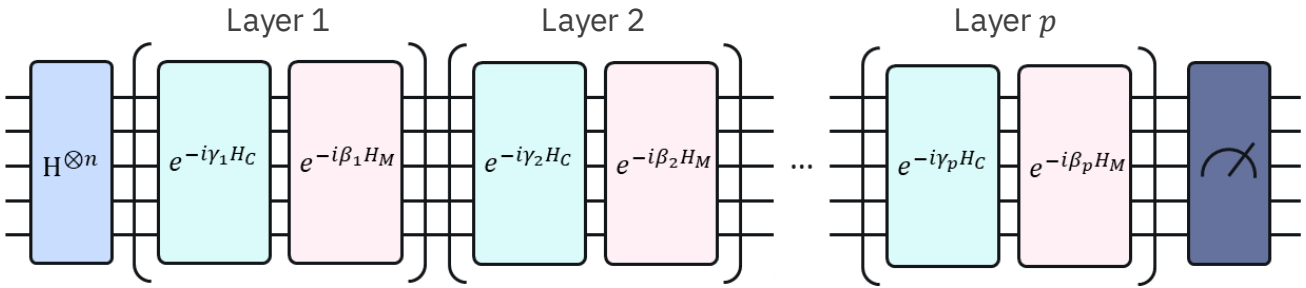
7. Update parameters γ_k, β_k

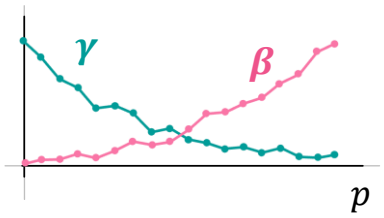
5. From measurement, obtain bitstrings and aggregate cut values



6. Send to classical optimizer

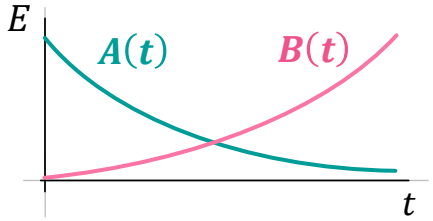
QAOA as an adiabatic schedule





Parameters γ, β represent A, B in the annealing, and the depth p represents the number of "time steps"

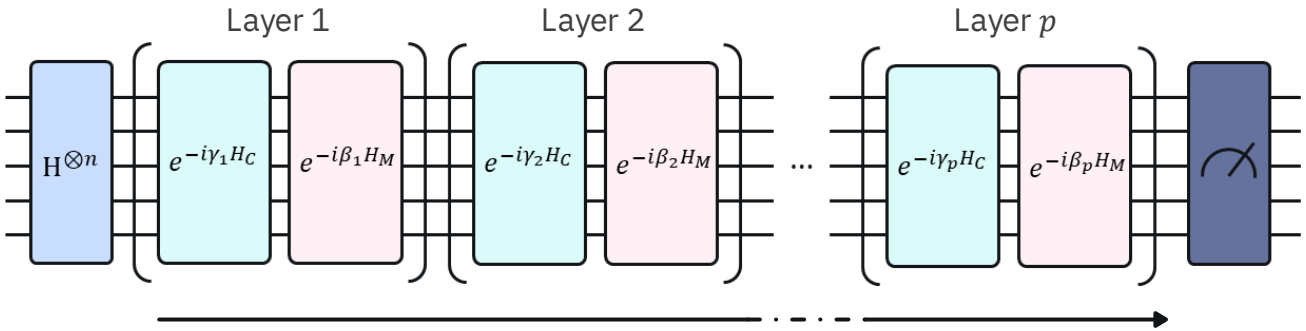
$\approx$



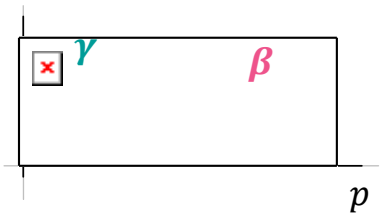
The image part with relationship ID rId6 was not found in the file.

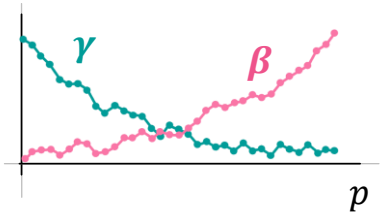
QAOA as an adiabatic schedule

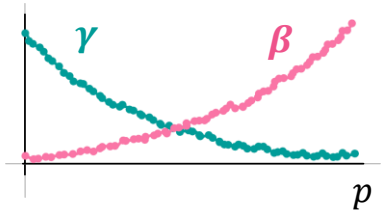
IBM **Quantum**



As we increase p, we slow down the "annealing" and make the approximation more accurate







Summary

Advantages of QAOA

- **Discretized** version of adiabatic annealing, which makes it implementable on superconducting hardware
- **Variational** version of adiabatic annealing, which makes it optimized and reduces runtimes
- Can **outperform** adiabatic annealing for subexponential runtimes
- Ability to include diabatic transitions may help overcome the **minimum gap problem**

Downsides of QAOA

- No performance **guarantees**
- **High depth** needed
- **Limited connectivity** of superconducting hardware
- Increasing number of **shots and iterations required** as problem size increases

Course outline

Part 1: Considered problem classes

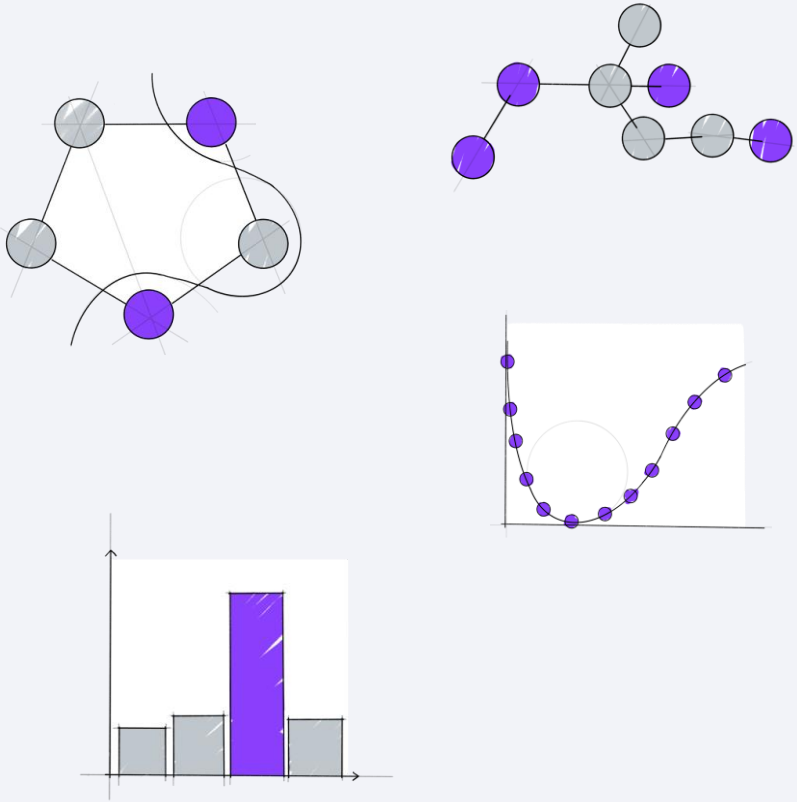
Part 2: From QUBO to Ising Hamiltonian

Part 3: The Adiabatic Theorem and Adiabatic Annealing

Part 4: From Adiabatic Annealing to QAOA

Part 5: Extensions of QAOA

Part 6: Executing QAOA on Superconducting Qubits



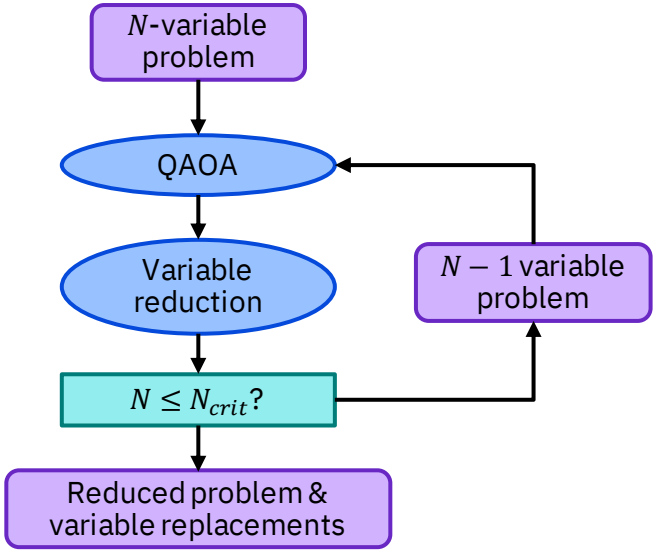
Recursive QAOA

PHYSICAL REVIEW LETTERS **125,** 260505 (2020)

Obstacles to Variational Quantum Optimization from Symmetry Protection

Sergey Bravyi,[1] Alexander Kliesch,[2] Robert Koenig,[3] and Eugene Tang[4]
[1]*IBM Quantum, IBM T. J. Watson Research Center, Yorktown Heights, New York 10598, USA*
[2]*Zentrum Mathematik, Technical University of Munich, 85748 Garching, Germany*
[3]*Institute for Advanced Study and Zentrum Mathematik, Technical University of Munich, 85748 Garching, Germany*
[4]*Institute for Quantum Information and Matter, Caltech, Pasadena, California 91125, USA*

(Received 22 October 2019; revised 16 September 2020; accepted 3 December 2020; published 24 December 2020)



Warm-start QAOA

IBM **Quantum**

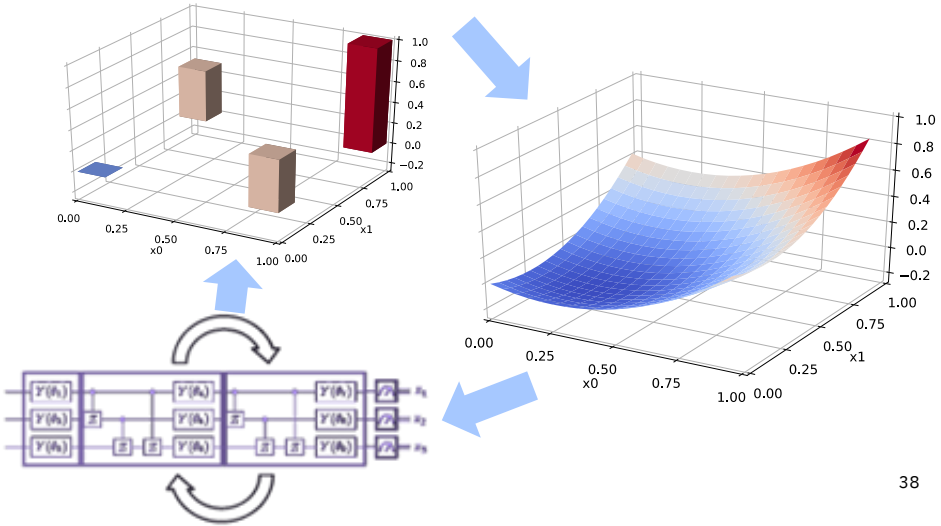
Quantum
the open journal for quantum science

Warm-starting quantum optimization

Daniel J. Egger[1], Jakub Mareček[2], and Stefan Woerner[1]

[1]IBM Quantum, IBM Research – Zurich, Säumerstrasse 4, 8803 Rüschlikon, Switzerland
[2]Czech Technical University, Karlovo nam. 13, Prague 2, the Czech Republic

Published: 2021-06-17, volume 5, page 479



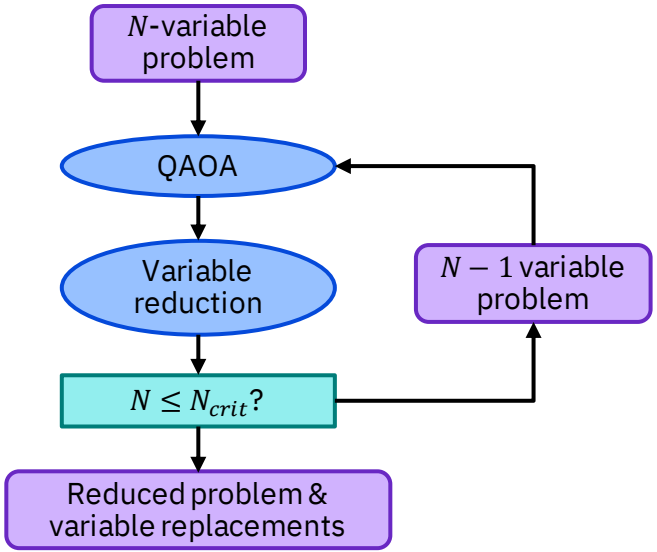
Recursive QAOA

PHYSICAL REVIEW LETTERS **125,** 260505 (2020)

Obstacles to Variational Quantum Optimization from Symmetry Protection

Sergey Bravyi,[1] Alexander Kliesch,[2] Robert Koenig,[3] and Eugene Tang[4]
[1]*IBM Quantum, IBM T. J. Watson Research Center, Yorktown Heights, New York 10598, USA*
[2]*Zentrum Mathematik, Technical University of Munich, 85748 Garching, Germany*
[3]*Institute for Advanced Study and Zentrum Mathematik, Technical University of Munich, 85748 Garching, Germany*
[4]*Institute for Quantum Information and Matter, Caltech, Pasadena, California 91125, USA*

(Received 22 October 2019; revised 16 September 2020; accepted 3 December 2020; published 24 December 2020)



Warm-start QAOA

IBM **Quantum**

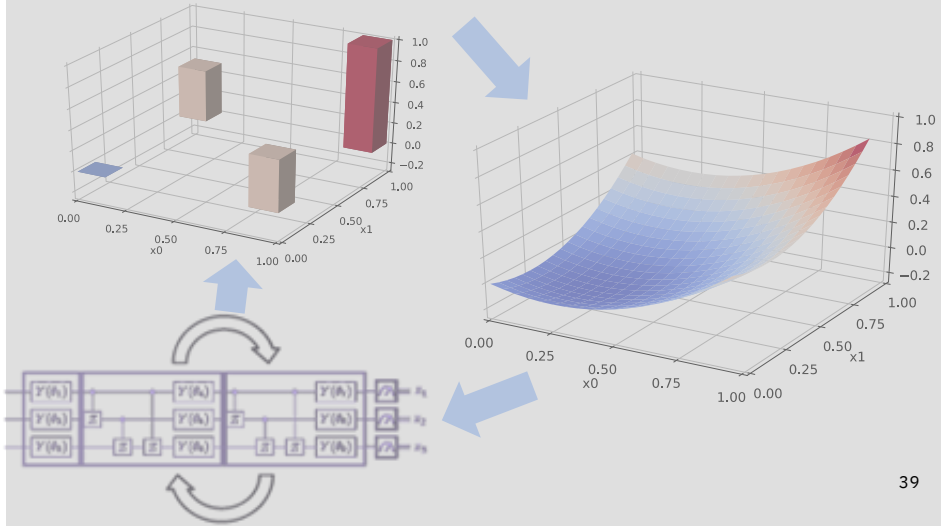
Quantum
the open journal for quantum science

Warm-starting quantum optimization

Daniel J. Egger[1], Jakub Mareček[2], and Stefan Woerner[1]

[1]IBM Quantum, IBM Research – Zurich, Säumerstrasse 4, 8803 Rüschlikon, Switzerland
[2]Czech Technical University, Karlovo nam. 13, Prague 2, the Czech Republic

Published: 2021-06-17, volume 5, page 479



Recursive QAOA

Idea

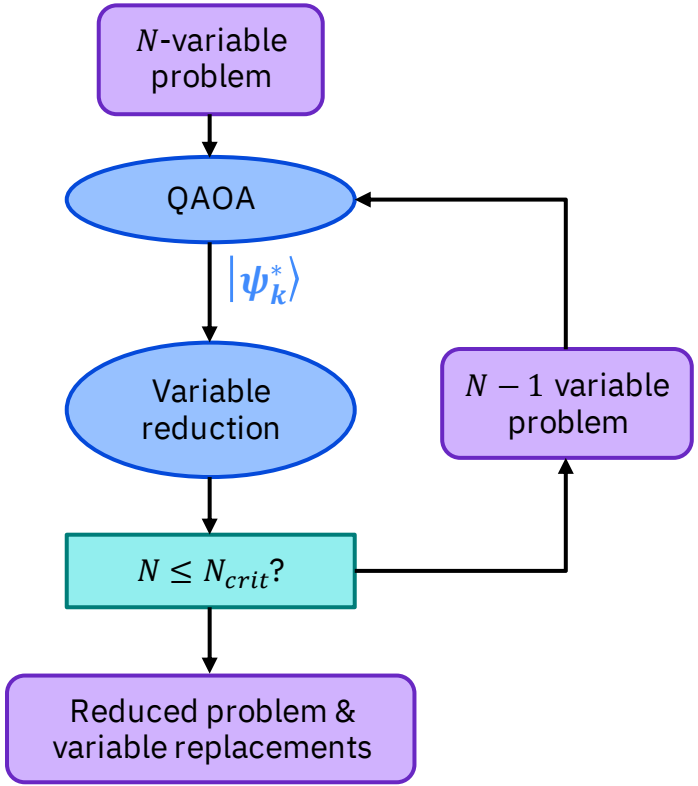
Run QAOA multiple times. At each recursion reduce the number of variables by one.

Variable reduction

- Optimal **QAOA state** $|\psi_k^*\rangle$ at recursion k.
- Compute the correlation matrix $\mathcal{M}_{ij,k} = \langle\psi_k^*|Z_iZ_j|\psi_k^*\rangle$
- Find the maximum correlation $\{i^*, j^* : |\mathcal{M}_{i^*j^*,k}| \geq |\mathcal{M}_{ij,k}|\}$
- Substitute variables $Z_{i^*} = \mathrm{sgn}(\mathcal{M}_{i^*j^*,k})\, Z_{j^*}$

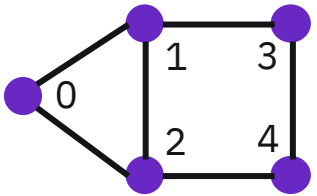
Output

- Substitutions $\{Z_1 = -Z_5, Z_2 = Z_3, Z_4 = Z_5, \ldots\}$



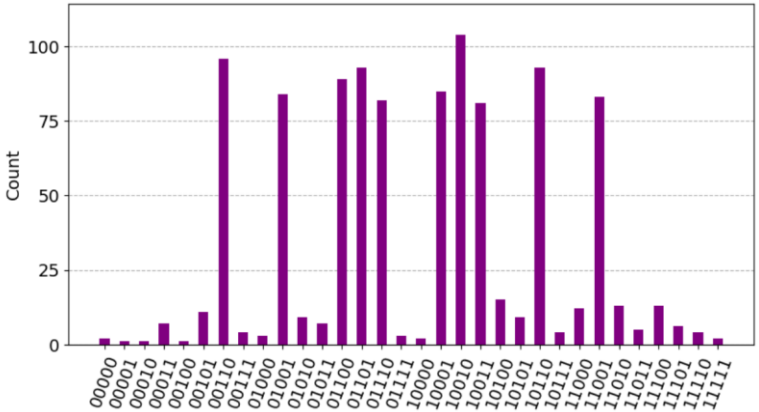
Recursive QAOA example

Optimization problem

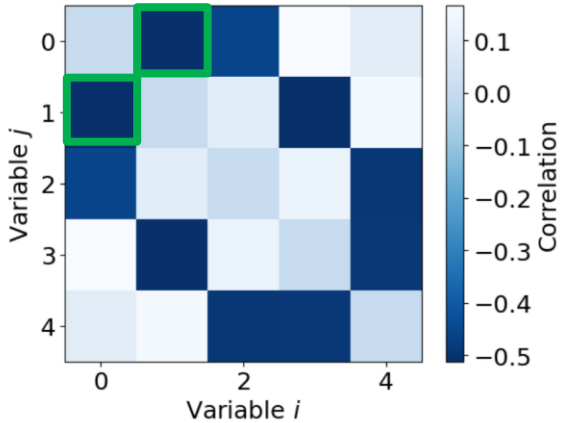


QAOA

Optimized state of $p = 1$



Compute correlation matrix $\mathcal{M}$



Largest correlation

Variable substitution

$$Z_0 = -Z_1$$

New problem

Recursive QAOA justification

Theorem

For each integer n divisible by 6 there is a family of $2^{n/3}$ Ising Hamiltonians of the form $H_n = \sum_{k \in \mathbb{Z}_n} J_k Z_k Z_{k+1}$ such that the following holds for all Hamiltonians in the family

1. There is a **local classical algorithm** which achieves the approximation **ratio of 1**.
2. **Level-p QAOA** achieves an approximation **ratio of at most $p/(p+1)$**.
3. **Level-1 RQAOA** achieve an approximation **ratio of 1**.

Proof: See Bravyi et al. PRL 2020

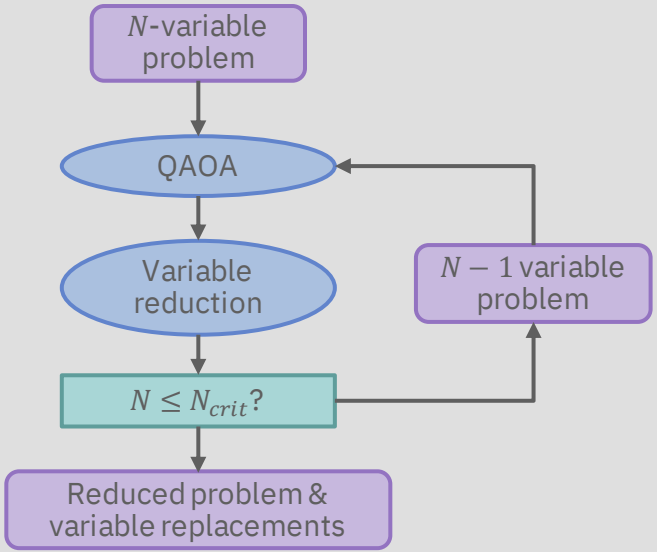
Recursive QAOA

PHYSICAL REVIEW LETTERS **125,** 260505 (2020)

Obstacles to Variational Quantum Optimization from Symmetry Protection

Sergey Bravyi,[1] Alexander Kliesch,[2] Robert Koenig,[3] and Eugene Tang[4]
[1]*IBM Quantum, IBM T. J. Watson Research Center, Yorktown Heights, New York 10598, USA*
[2]*Zentrum Mathematik, Technical University of Munich, 85748 Garching, Germany*
[3]*Institute for Advanced Study and Zentrum Mathematik, Technical University of Munich, 85748 Garching, Germany*
[4]*Institute for Quantum Information and Matter, Caltech, Pasadena, California 91125, USA*

(Received 22 October 2019; revised 16 September 2020; accepted 3 December 2020; published 24 December 2020)



Warm-start QAOA

IBM **Quantum**

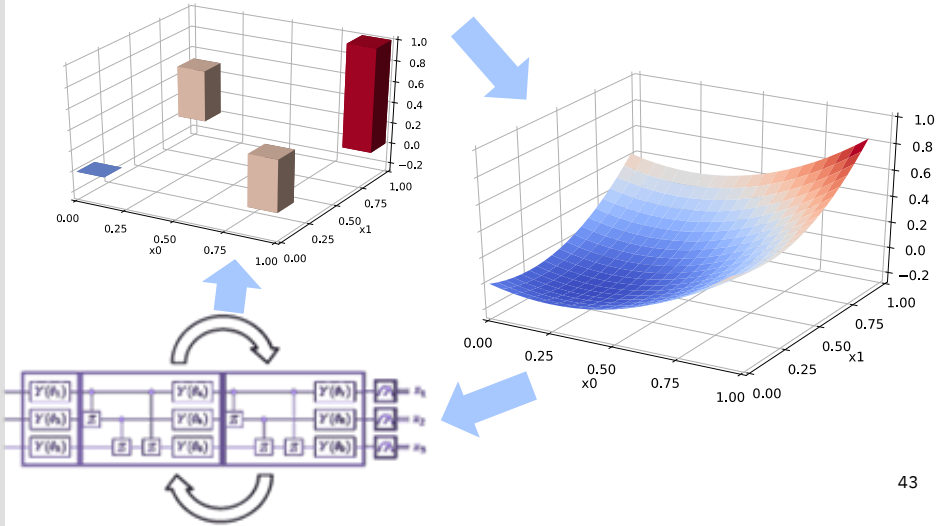
Quantum
the open journal for quantum science

Warm-starting quantum optimization

Daniel J. Egger[1], Jakub Mareček[2], and Stefan Woerner[1]

[1]IBM Quantum, IBM Research – Zurich, Säumerstrasse 4, 8803 Rüschlikon, Switzerland
[2]Czech Technical University, Karlovo nam. 13, Prague 2, the Czech Republic

Published: 2021-06-17, volume 5, page 479



Continuous relaxation of a QUBO

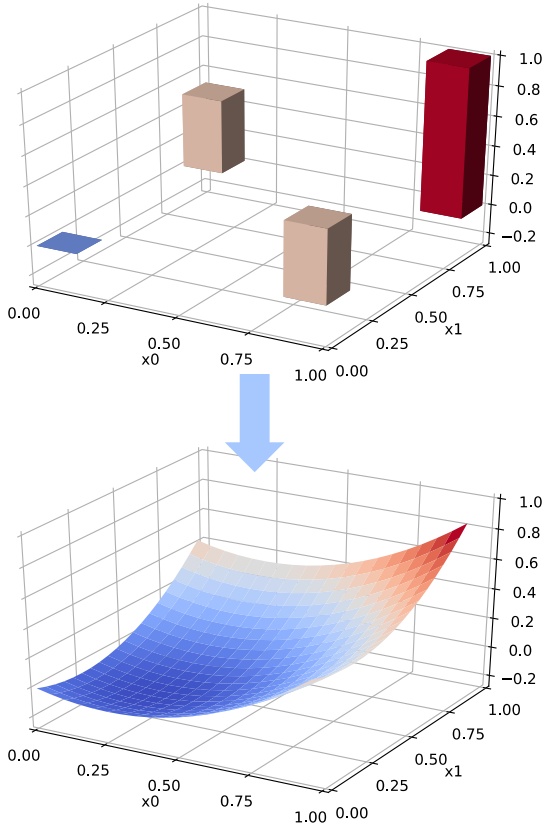
Drop binary constraints to get Quadratic Program (QP)

$$\min_{x \in [0,1]^n} x^T Q x + \mathrm{b}^{\mathrm{T}}\mathrm{x}$$

Convex QPs can be efficiently solved classically

If Q is positive semidefinite the QP is convex and can be solved efficiently classically.

Since $x_i^2 = x_i$ for $x_i \in \{0, 1\}$, we can move weights between the diagonal of Q and the linear part b. Thus, we can always construct a convex QP that corresponds to a continuous convex relaxation of the QUBO.



Warm-starting QAOA: Initial state

Suppose a QUBO with QP solution

$$c^* \in [0,1]^n$$

Prepare initial state as

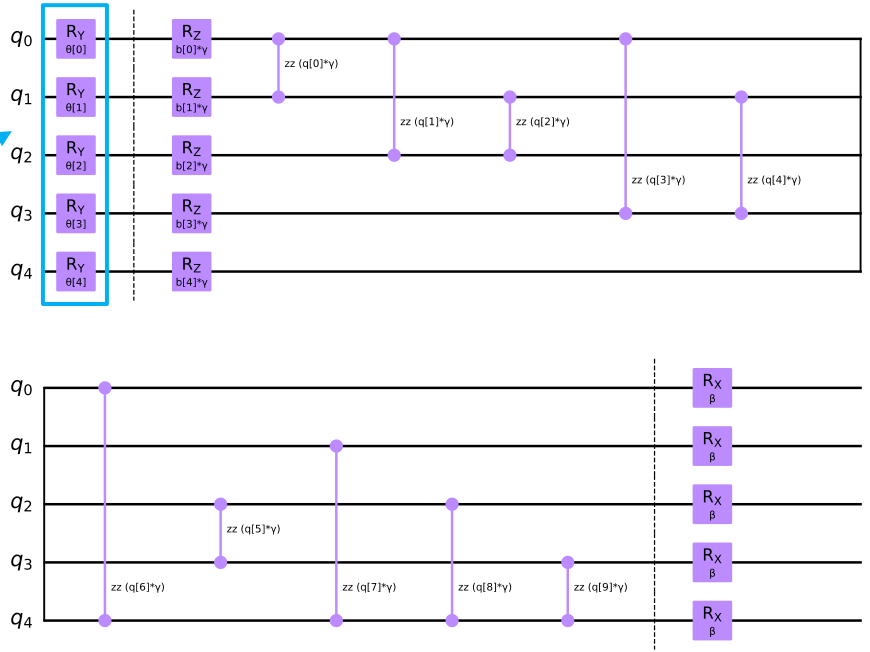
$$|\psi_{c^*}\rangle = \otimes_{i=0}^{n-1} R_Y(\theta_i)|0\rangle$$

with

$$\theta_i = 2\sin^{-1}\left(\sqrt{c_i^*}\right)$$

i.e., we have

$$|\psi_{c^*}\rangle = \otimes_{i=0}^{n-1} \left(\sqrt{1-c_i^*}|0\rangle + \sqrt{c_i^*}|1\rangle\right)$$



Warm-starting QAOA: Mixing Hamiltonian

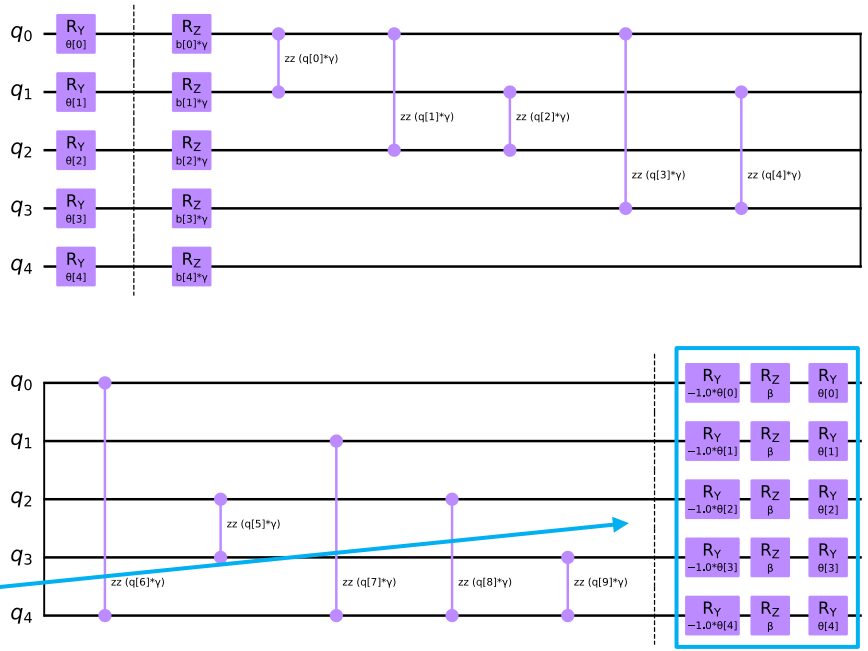
Define the mixing Hamiltonian as

$$H^i_{X,c^*} = \begin{pmatrix} 2c_i^* - 1 & -2\sqrt{c_i^*(1-c_i^*)} \\ -2\sqrt{c_i^*(1-c_i^*)} & 1 - 2c_i^* \end{pmatrix}$$

and $H_{X,c^*} = \sum_i H^i_{X,c^*}$, which has ground state $|\psi_{c^*}\rangle$

Then, the evolution can be implemented as

$$e^{-iH_{X,c^*}\beta} = \bigotimes_{i=0}^{n-1} R_Y(\theta_i)\, R_Z(-2\beta) R_Y(-\theta_i)$$



→ Warm-start QAOA

WS-QAOA: Convergence & regularization

Suppose $c^* \in (0,1)^n$, then $|\psi_{c^*}\rangle$

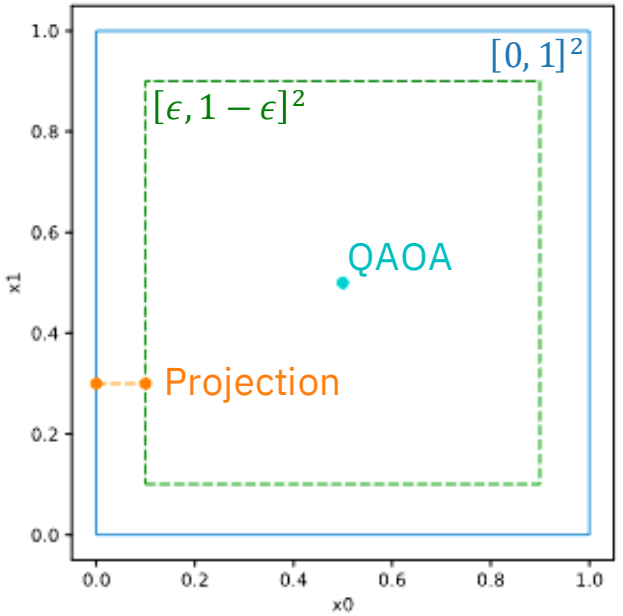
- overlaps with optimal solution
- is the ground state of H_{X,c^*}

→ The **Adiabatic Theorem** implies that WS-QAOA converges to optimal solution as $p \to \infty$.

If there is a $c_i^* \in \{0,1\}$ then WS-QAOA will only change the phase of the corresponding qubit.

To overcome this limitation, we project c^* in to the $[\epsilon, 1-\epsilon]^n$ for an $\epsilon > 0$.

For $\epsilon = 0.5$ this recoveres the original QAOA.



WS-QAOA: Portfolio optimization

Markowitz portfolio optimization

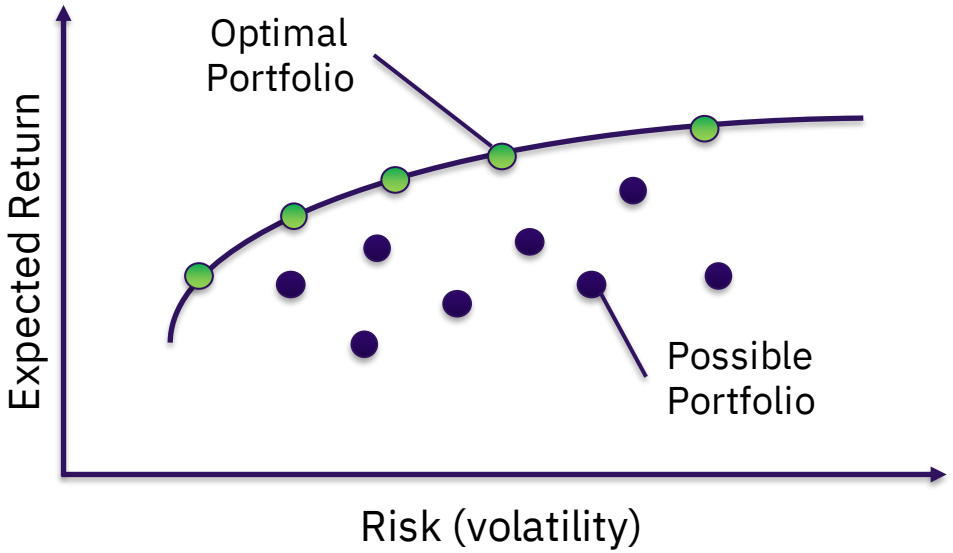
$$\max \quad \boldsymbol{\mu}^T \boldsymbol{x} - q\boldsymbol{x}^T \Sigma \boldsymbol{x}$$

$$\text{such that} \quad \mathbf{1}^T \boldsymbol{x} = B$$

- Weights $\boldsymbol{x} \in \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{n}}$
- Returns $\boldsymbol{\mu} \in \mathbb{R}^n$
- Covariance $\boldsymbol{\Sigma}$ $n \times n$ real matrix
- Risk-return trade-off q
- B budget

Goal: minimize risk & maximize returns

Portfolio Optimization Efficient Frontier



WS-QAOA: Portfolio optimization

Markowitz portfolio optimization

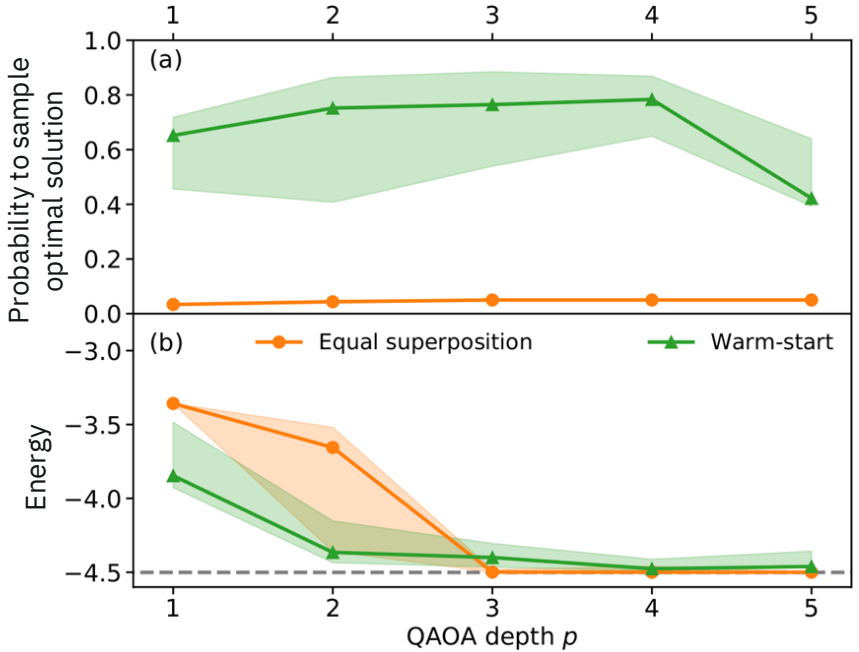
$$\max \quad \boldsymbol{\mu}^T \boldsymbol{x} - q\boldsymbol{x}^T \Sigma \boldsymbol{x}$$

$$\text{such that} \quad \mathbf{1}^T \boldsymbol{x} = B$$

- Weights $\boldsymbol{x} \in \{\mathbf{0}, \mathbf{1}\}^n$
- Returns $\boldsymbol{\mu} \in \mathbb{R}^n$
- Covariance $\boldsymbol{\Sigma}$ $n \times n$ real matrix
- Risk-return trade-off q
- B budget

Goal: minimize risk & maximize returns

WS-QAOA: better performance especially at short depth



- 1 random instance with $n = 6$
- 10 repetitions for param. opt.

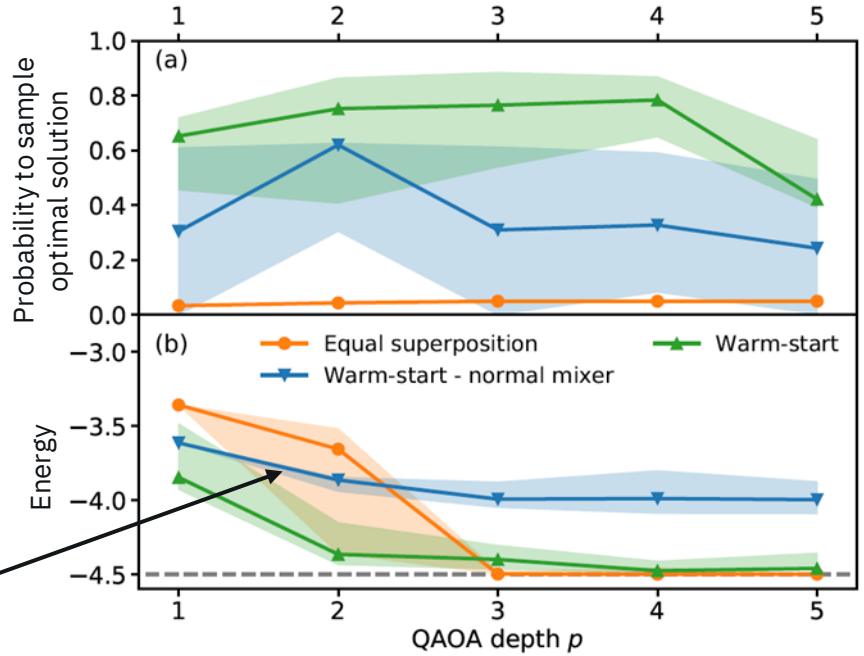
WS-QAOA: Portfolio optimization

Markowitz portfolio optimization

$$\max \quad \boldsymbol{\mu}^T \boldsymbol{x} - q\boldsymbol{x}^T \Sigma \boldsymbol{x}$$

$$\text{such that} \quad \mathbf{1}^T \boldsymbol{x} = B$$

Starts with the continuous solution but uses the equal superposition mixer.



- 1 random instance with $n = 6$
- 10 repetitions for param. opt.

Summary

- **Recursive QAOA** applies QAOA multiple times on reduced problems.
  $\Longrightarrow$ Provably better performance than QAOA on a certain class of Hamiltonians.

- **Warm-start QAOA** classically solves a relaxed problem to warm-start QAOA by changing the initial state and mixer operators.
  $\Longrightarrow$ Can inherit performance guarantees from classical approaches.

Course outline

Part 1: Considered problem classes

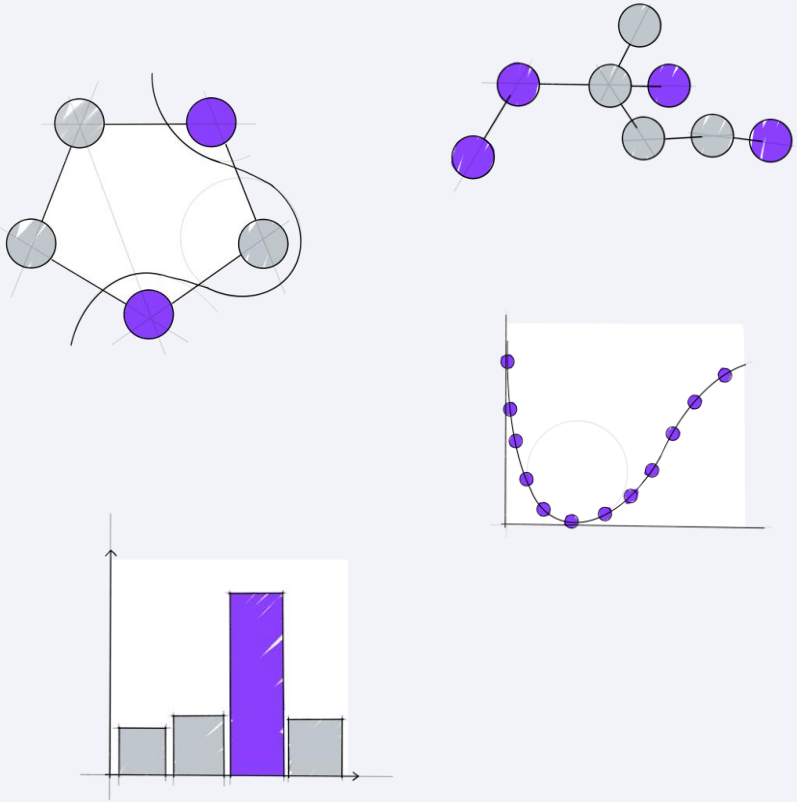
Part 2: From QUBO to Ising Hamiltonian

Part 3: The Adiabatic Theorem and Adiabatic Annealing

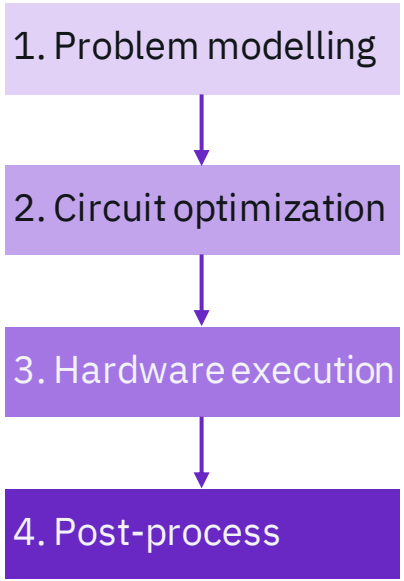
Part 4: From Adiabatic Annealing to QAOA

Part 5: Extensions of QAOA

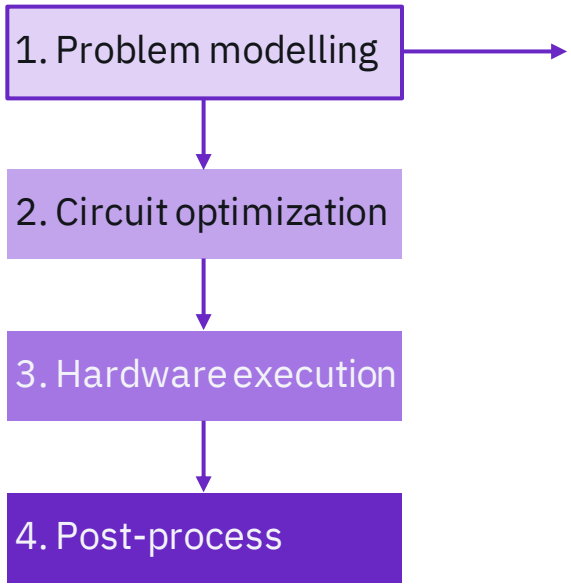
Part 6: Executing QAOA on Superconducting Qubits



Execution on superconducting qubits



Execution on superconducting qubits



Optimization problems can be modelled in different ways. This has an impact on how the problem is solved

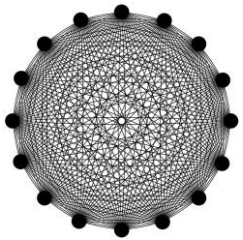
Example: model higher-order terms or introduce extra variables to reduce to a QUBO?

Execution on superconducting qubits

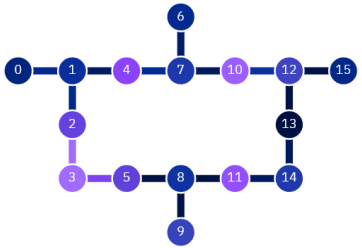
1. Problem modelling
2. Circuit optimization
3. Hardware execution
4. Post-process

Circuit optimization

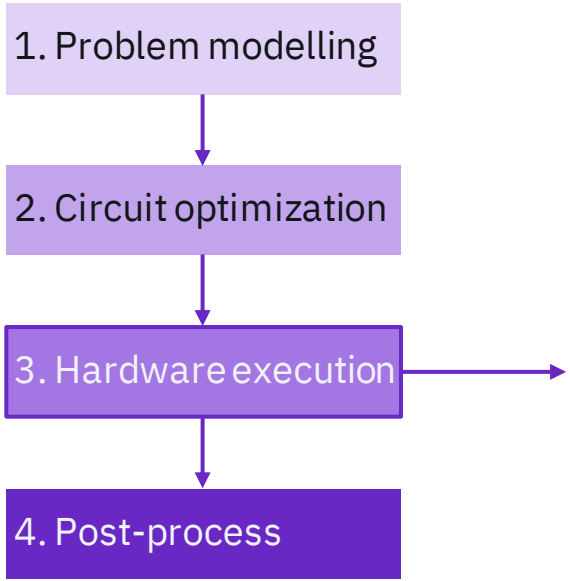
- Initial mapping (assigning program qubits to physical ones)
  SABRE [Liu (2019)], SAT Mapping [Matsuo (2023)]
- SWAP routing (overcoming limited device connectivity)
  SABRE [Liu (2019)], SWAP strategies [Weidenfeller (2022)]
- Noise suppression
  Dynamical decoupling [Ezzell (2022)], Pulse-efficient [Earnest (2021)]



Requires
SWAP gates



Execution on superconducting qubits



Sampler or Estimator primitive?

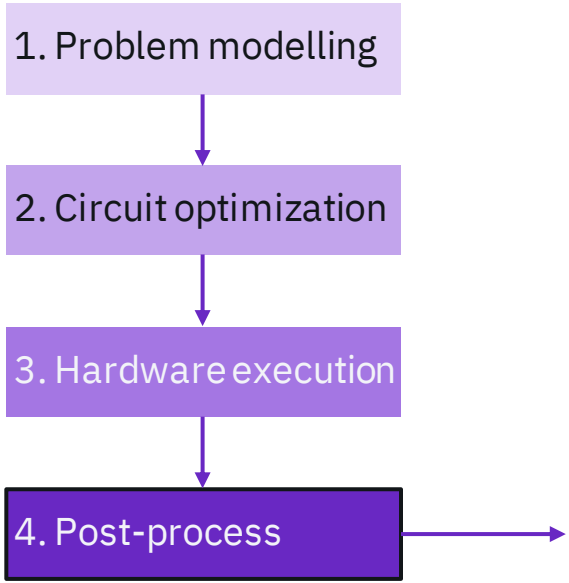
Typically, obtaining *samples* $x_i \in \{0,1\}^n$ from the quantum device, i.e., candidate solutions to

$$\min_{x \in \{0,1\}^n} f(x)$$

Some algorithms, e.g., R-QAOA can operate with expectation values such as $\langle Z_i Z_j \rangle$.

May also require aggregating the samples according to different aggregation functions such as CVaR_α: take the best α fraction of samples.

Execution on superconducting qubits



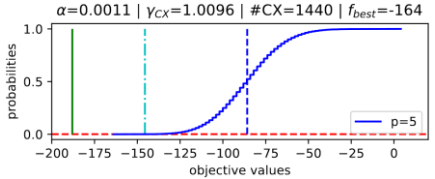
Turn the measurements into a solution to the initial problem.

Examples

- Standard QAOA: the samples $x_i \in \{0,1\}^n$ directly translate into candidate solutions.
- Recursive QAOA: the largest magnitude $\langle Z_i Z_j \rangle$ correlator gives a variable replacement: $Z_i = \mathrm{sgn}(\langle Z_i Z_j \rangle) Z_j$ to reduce the problem.

Some recent QAOA experiments on superconducting qubits

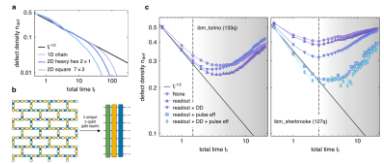
Provable bounds for noise-free expectation values computed from noisy samples.



What: up to 127 qubits on HW native 3^{rd} order with $p = 5$.

Barron et al. arXiv:2312.00733

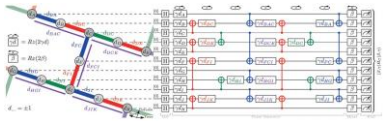
Benchmarking digital quantum simulations and optimization using quantum critical dynamics.



What: up to 133 qubits on HW native Ising problems.

Miessen et al. arXiv:2404.08053

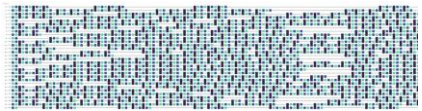
Quantum Annealing vs. QAOA: 127 Qubit Higher-Order Ising Problems on NISQ Computers



What: 127 qubits on HW native 3^{rd} order problems with $p = 2$.

Pelofske et al. (2023)
ISC High Performance 2023

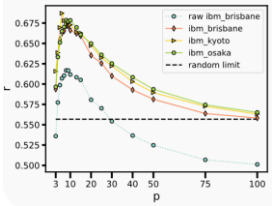
Large-scale QAOA on non-planar graphs with machine learning noise mitigation.



What: up to 40 qubits on RR3 graphs and depth $p = 2$.

Sack et al. (2024) Phys. Rev. Research

Towards a universal QAOA protocol: Evidence of quantum advantage in solving combinatorial optimization problems.



What: up to 105 qubits on HW native problems with p up to 100.

Montanez-Barrera et al. arxiv:2405.09169

Utility-scale hardware enables research and development at scale

Course summary

We learnt about

- Combinatorial optimization and how to formulate the problem as a ground state problem
- Adiabatic computing and how to derive the QAOA from it
- Extensions of QAOA and considerations when running on hardware.

Concluding thoughts

Many optimization problems are NP-hard. However, many classical approaches are heuristic and work well in practice. Quantum approaches to combinatorial optimization, typically also heuristics, must be explored on practically valuable instances where classical heuristics struggle.

Course outline

Part 1: Considered problem classes

Part 2: From QUBO to Ising Hamiltonian

Part 3: The Adiabatic Theorem and Adiabatic Annealing

Part 4: From Adiabatic Annealing to QAOA

Part 5: Extensions of QAOA

Part 6: Executing QAOA on Superconducting Qubits