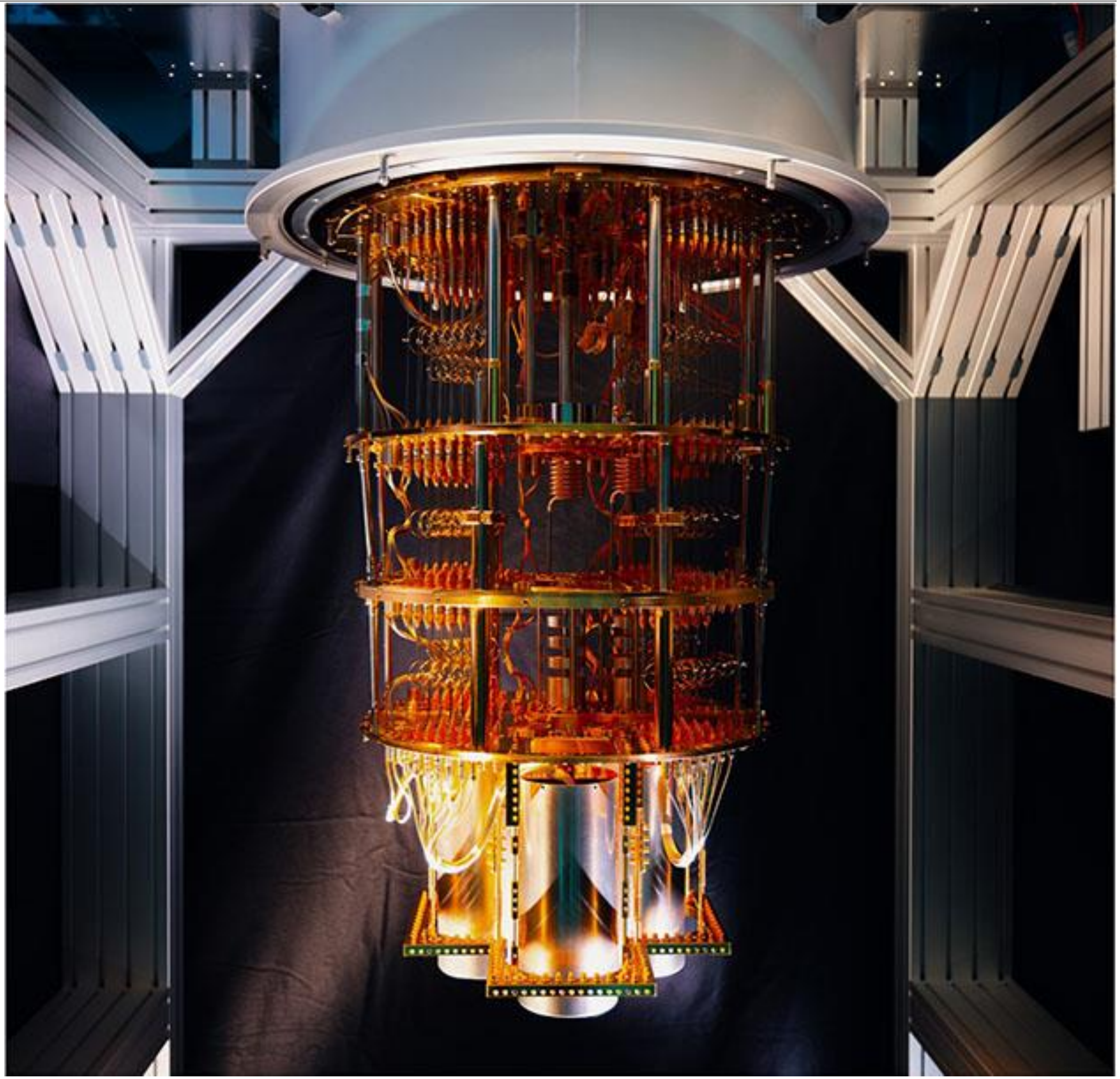
IQM

WE BUILD QUANTUM COMPUTERS

Introduction to Quantum Approximate Optimization Algorithm



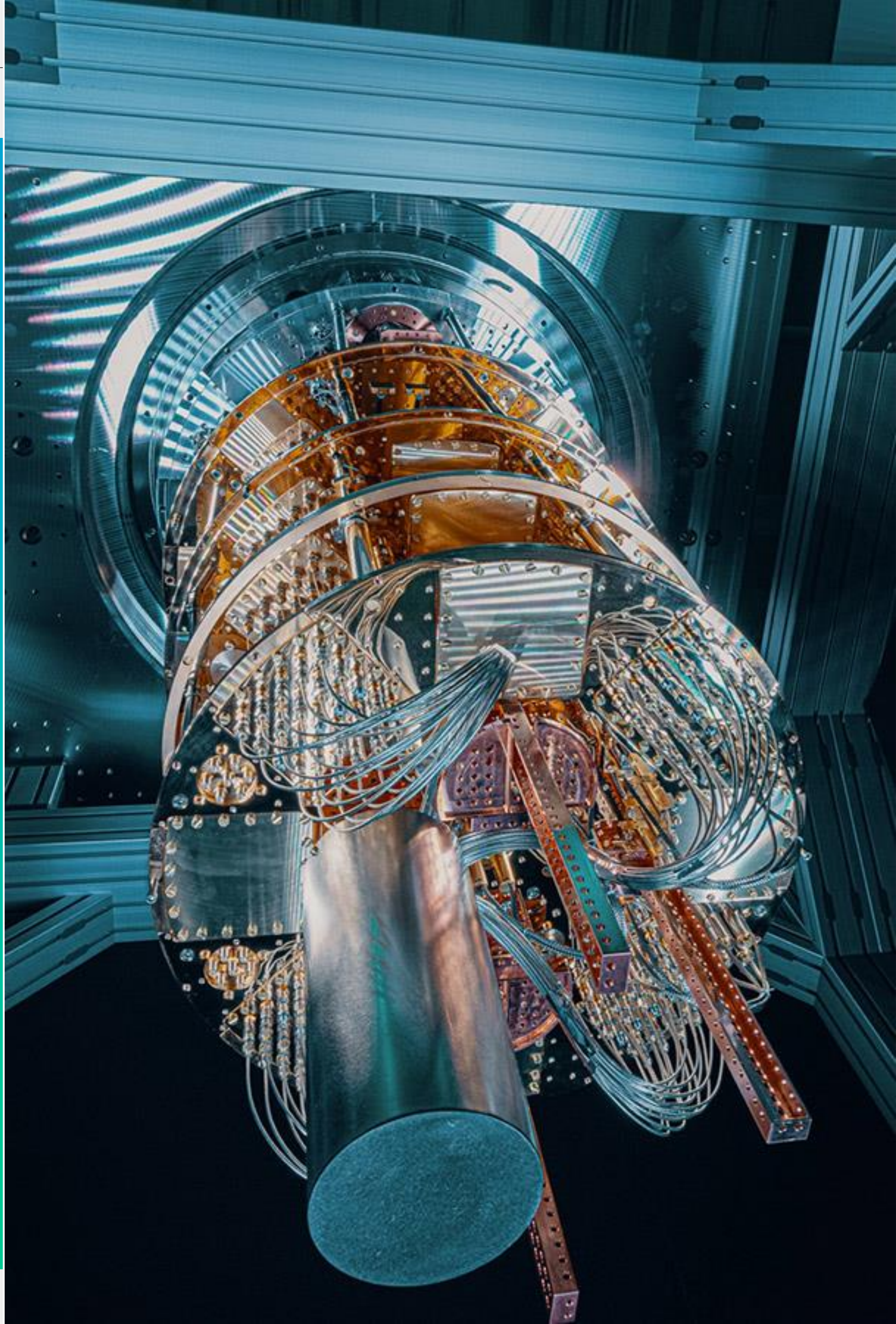
Jiří (Jirka) Guth Jarkovský

Applications & Algorithms team

IQM Germany

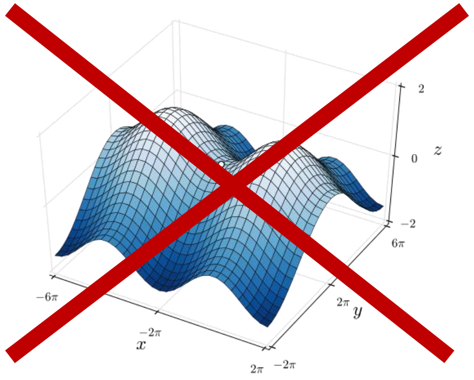
Outline

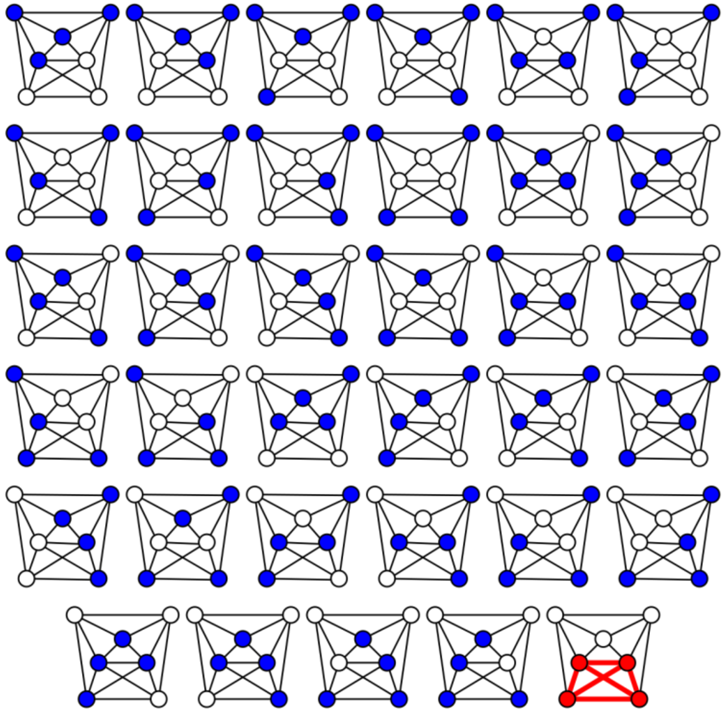
- Motivation
    - Combinatorial Optimization
    - Max-Cut Classically
- QAOA basics
    - Max-Cut with Quantum Annealing
    - QAOA
- QAOA advanced
    - Applications
    - Research at IQM



Motivation: Combinatorial Optimization

- Discrete variables – no calculus methods
- Typically exponential solution space: NP-hard
- Typically on a graph

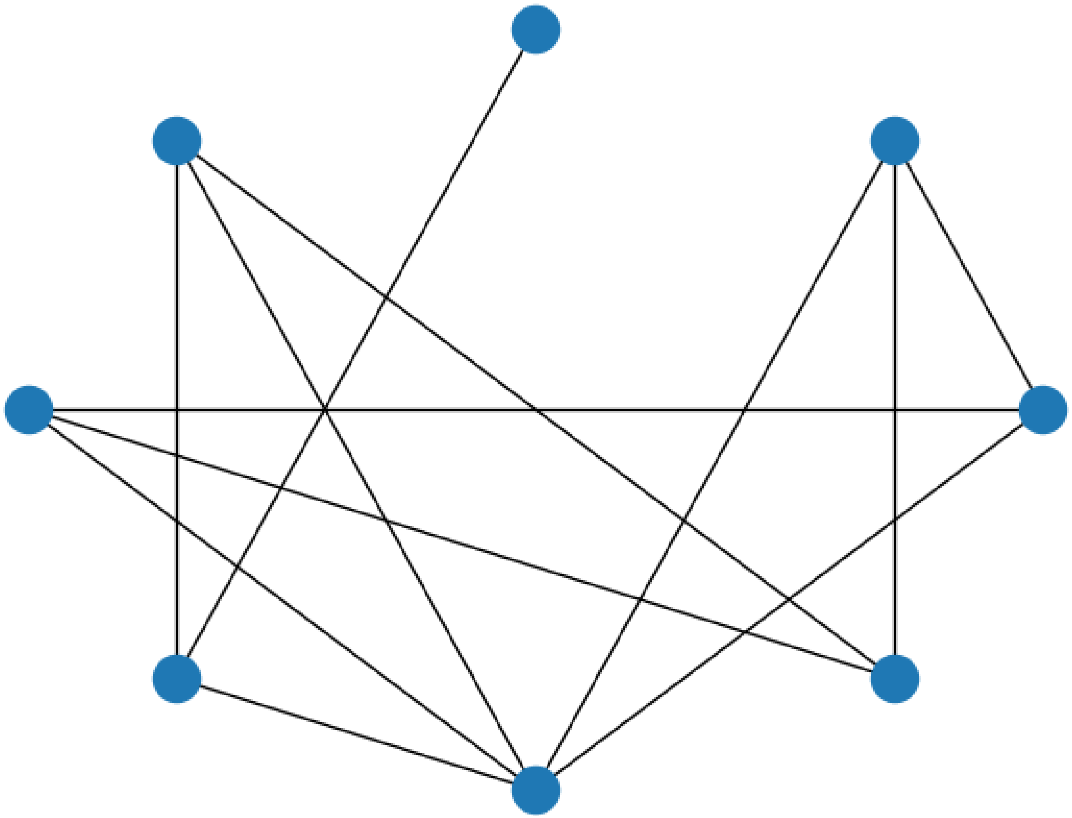




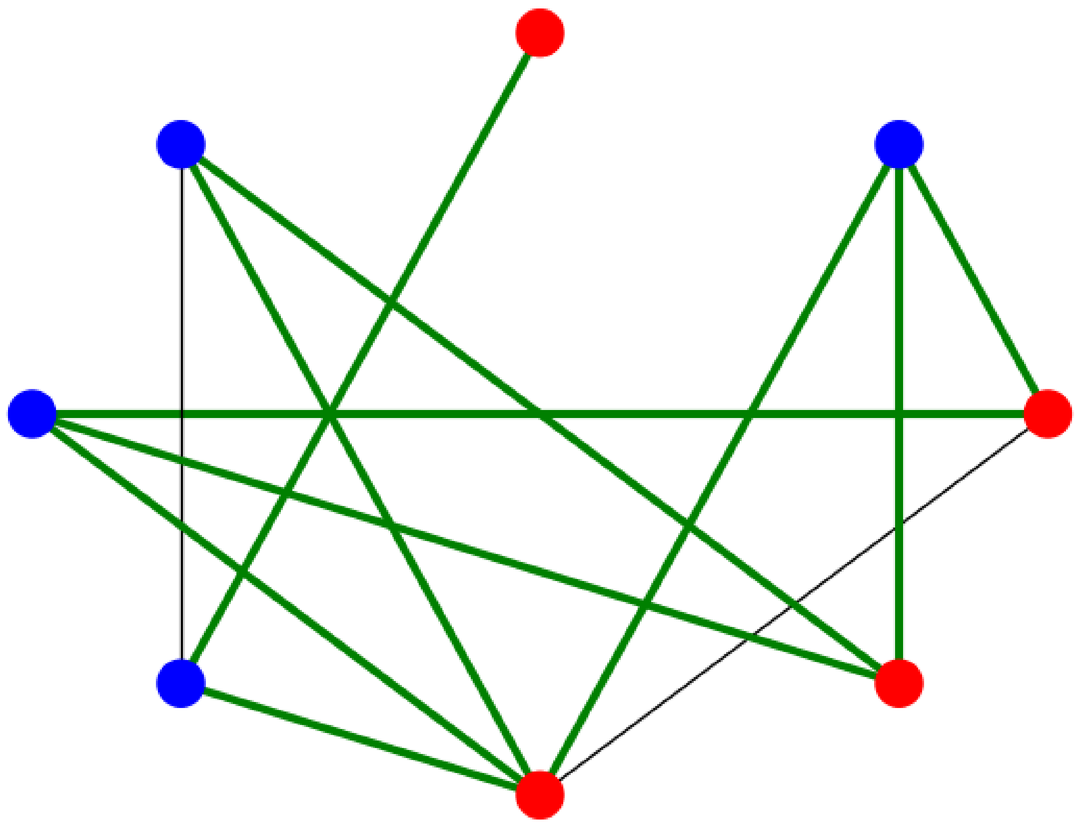
- Problems:
  - Traveling Salesman
  - Knapsack
  - Constraint Satisfaction
  - Maximum Cut
  - Maximum Independent Set
  - …
- Applications:
  - Logistics
  - Medicine
  - Sociology
  - Operations Research
  - Computer Science
  - …

Images: Nicoguaro, Thore Husfeldt

Maximum Cut

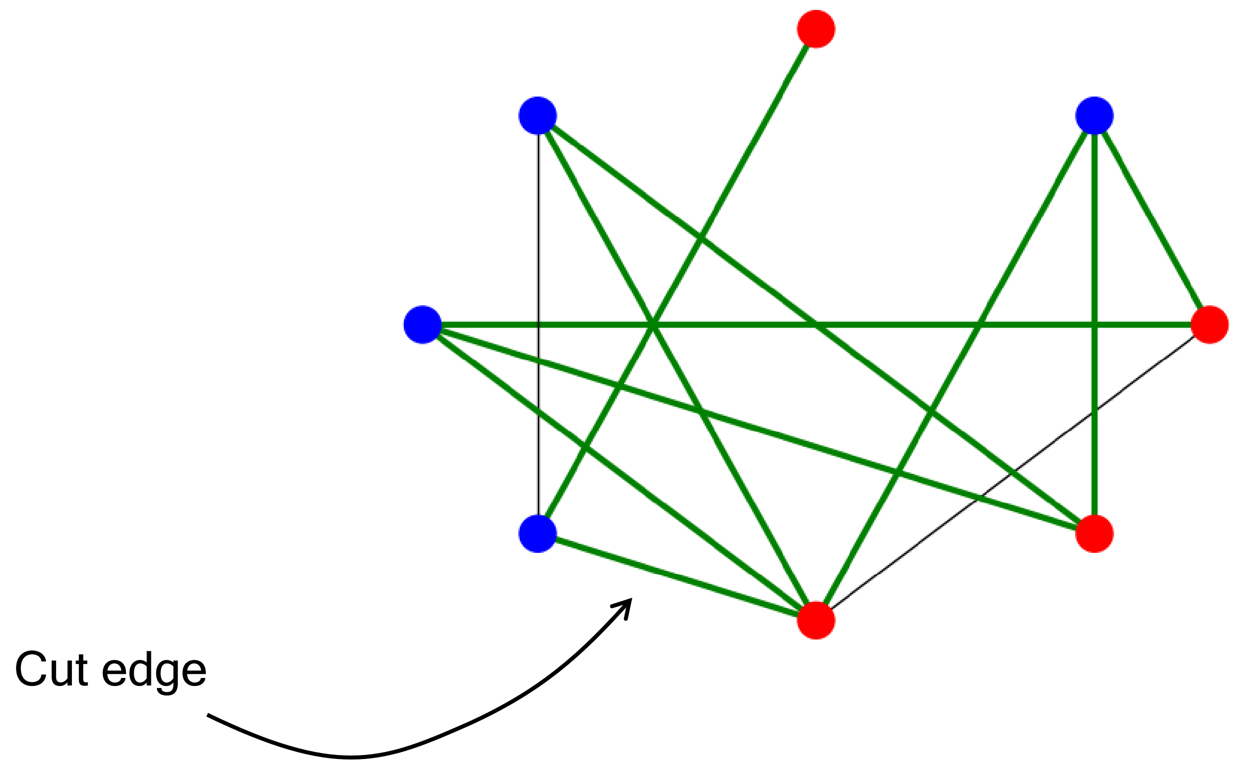


Maximum Cut



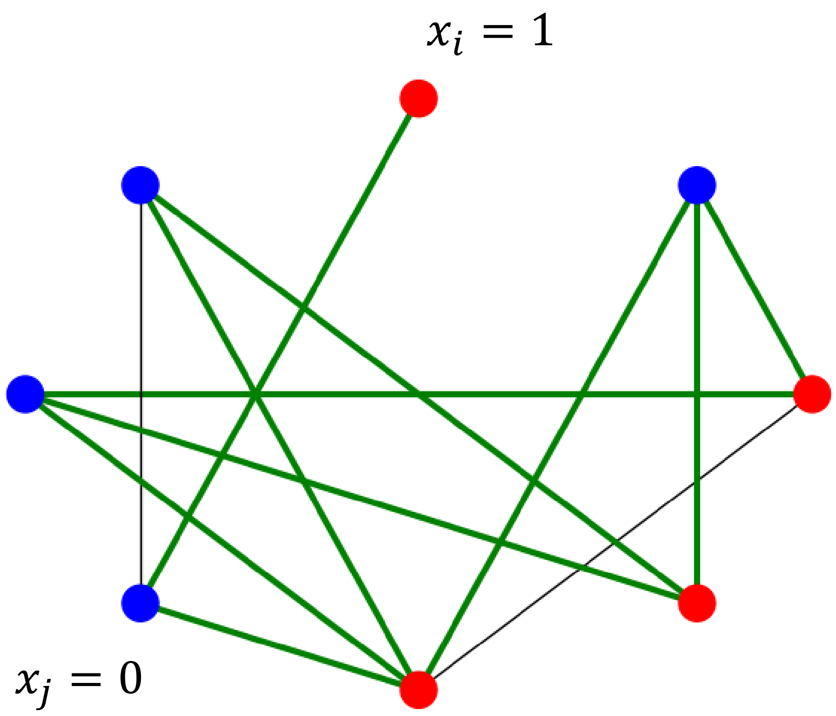
Maximum Cut

$$F_C(\text{cut}) = -\sum_{\text{cut edges}} 1$$



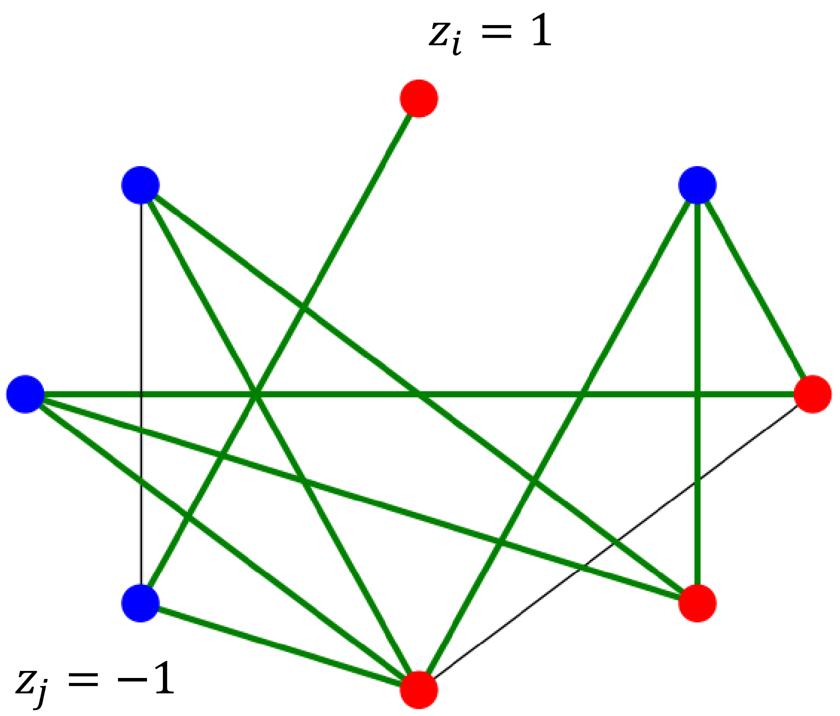
Maximum Cut

$$F_C(\text{cut}) = -\sum_{\text{edges } i,j} (x_i + x_j - 2x_i x_j)$$



Maximum Cut

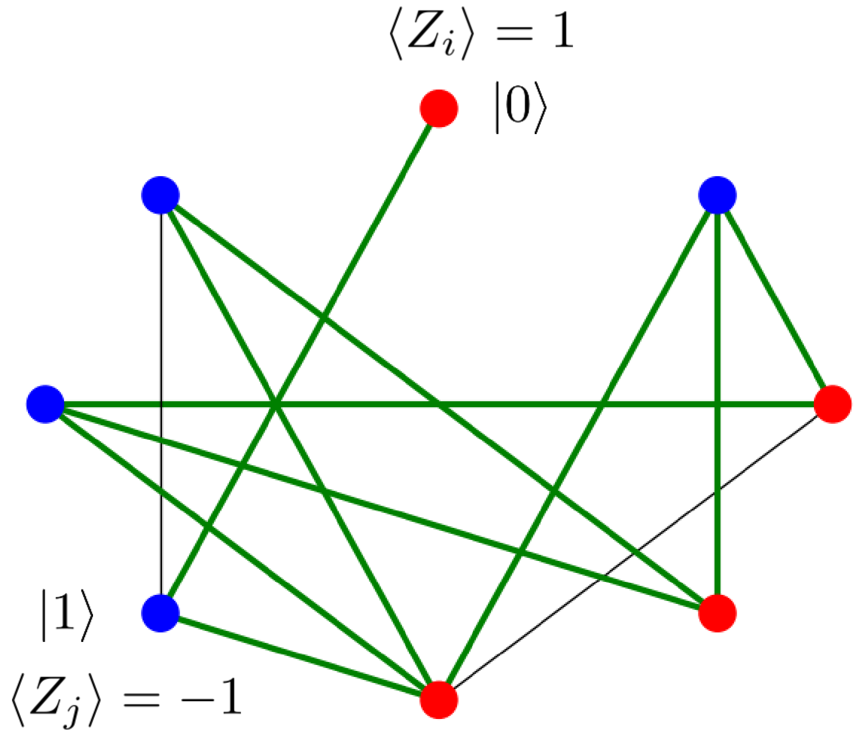
$$F_C(\text{cut}) = \sum_{\text{edges } i,j} z_i z_j$$



Maximum Cut Quantized

$$F_C(\text{cut}) = \sum_{\text{edges } i,j} z_i z_j$$

$$H_C = \sum_{\text{edges } i,j} Z_i \otimes Z_j \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$



Solution to optimization problem = ground state of quantum system

Quantum Annealing

$$H_C = \sum_{\text{edges } i,j} Z_i \otimes Z_j$$

1. Start with a simple “mixer” Hamiltonian and prepare its ground state

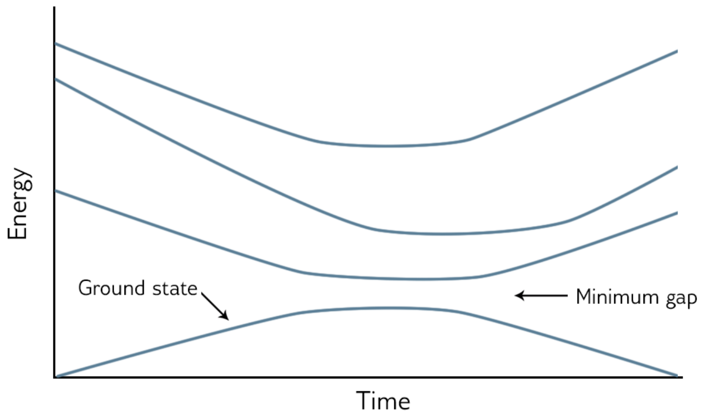
$$H_0 = -\sum_i X_i \qquad |\phi\rangle = |+\rangle^N$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

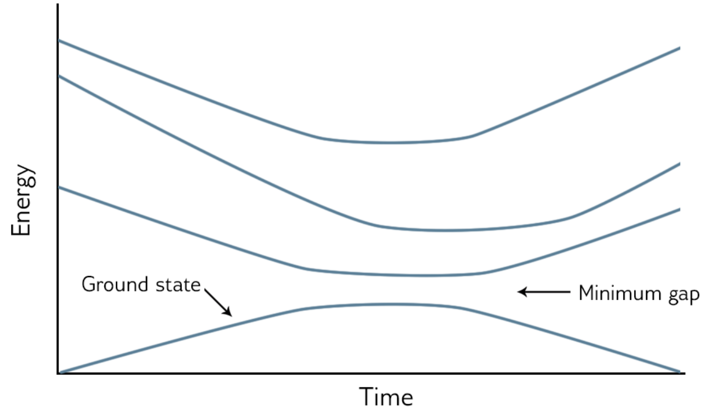
2. Slowly continuously change the Hamiltonian to the target Hamiltonian

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

3. At the end of the process, the state will be the ground state of the target Hamiltonian



Quantum Annealing



1. Start with a simple “mixer” Hamiltonian and prepare its ground state
2. Slowly continuously change the Hamiltonian to the target Hamiltonian
3. At the end of the process, the state will be the ground state of the target Hamiltonian

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

Time evolution from Schrödinger equation:

$$|\phi(T)\rangle = \mathcal{T}e^{-i\int_0^T H(t)\mathrm{d}t}|+\rangle^N \approx \prod_{l=1}^{p} e^{-i\beta_l H_0}e^{-i\gamma_l H_C}|+\rangle^N$$

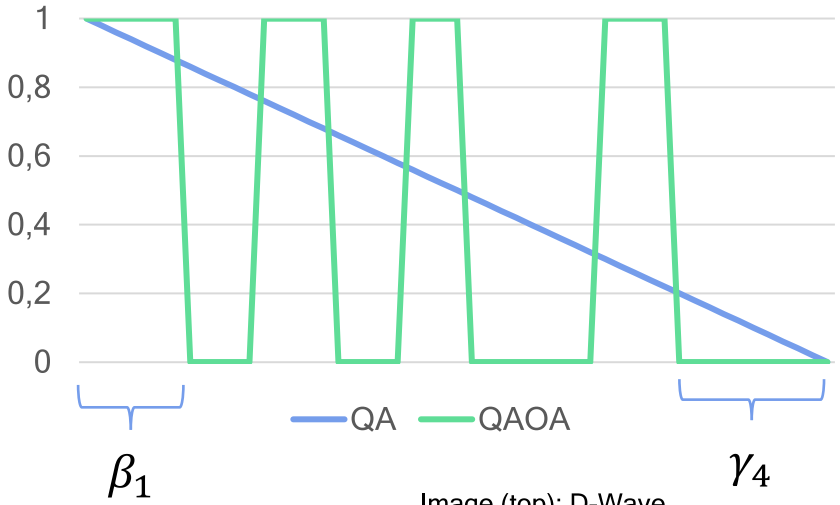
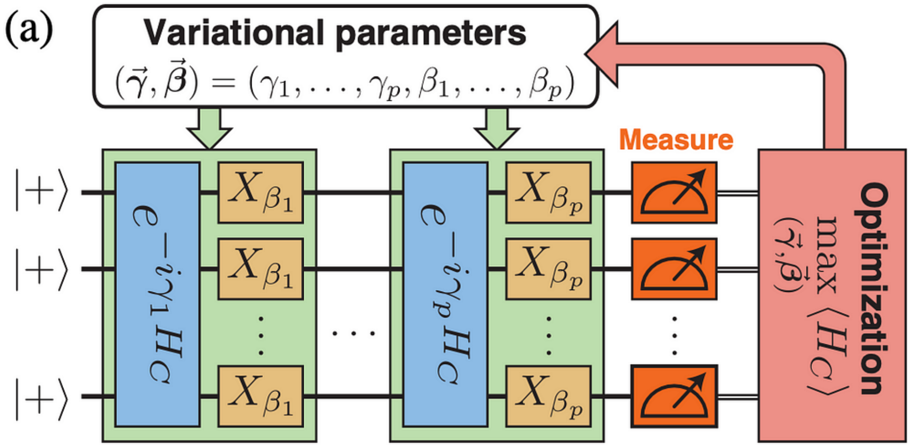


Image (top): D-Wave

QAOA

$$|\phi_{QAOA}\rangle = \prod_{l=1}^{p} e^{-i\beta_l H_0} e^{-i\gamma_l H_C} |+\rangle^N$$



$$H_C = \sum_{\text{edges } i,j} Z_i \otimes Z_j$$

$e^{-i\gamma Z_i Z_j} =$

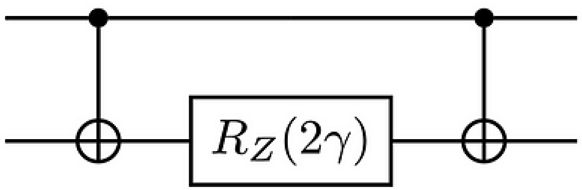
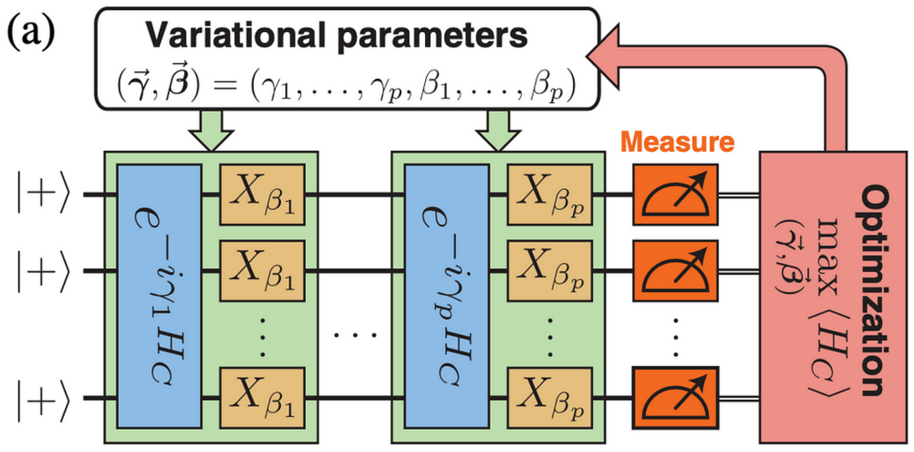


Image (left): Zhou et al.

QAOA



1. Initialize parameters $\theta = \{\beta_l, \gamma_l\}$, for example randomly
2. Run QAOA circuit on QPU and measure multiple times
3. Approximate mean value of problem Hamiltonian:

$$\langle \phi_{QAOA} | H_C | \phi_{QAOA} \rangle \approx \frac{1}{N} \sum_{v_i \in (Measurements)} F_C(v_i)$$

4. Update parameters, with classical optimisation routine $\theta = \{\beta_l, \gamma_l\}$
5. Repeat 2-4 until satisfying approximate mean value of problem Hamiltonian has been reached
6. Output best measurement result: $\underset{v_i}{argmin}\, C(v_i)$

Quadratic Unconstrained Binary Optimization

- QAOA-compatible cost functions
- AKA Ising model

$$F_C = \mathbf{x}^\top Q\mathbf{x} = \sum_{i,j} Q_{ij} x_i x_j$$

$$H_C = \sum_{i \neq j} J_{ij} Z_i \otimes Z_j + \sum_i h_i Z_i$$

- https://arxiv.org/pdf/1302.5843.pdf

Ising formulations of many NP problems

Andrew Lucas
Department of Physics, Harvard University, Cambridge, MA, USA 02138

We provide Ising formulations for many NP-complete and NP-hard problems, including all of Karp's 21 NP-complete problems. This collects and extends mappings to the Ising model from partitioning, covering and satisfiability. In each case, the required number of spins is at most cubic in the size of the problem. This work may be useful in designing adiabatic quantum optimization algorithms.

Example: Traveling Salesman Problem

Constraints:
- Every city must be visited exactly once
- Every step, the salesman visits exactly one city

→ Add constraints as penalty terms in cost function (soft constraints) with penalty term A high enough

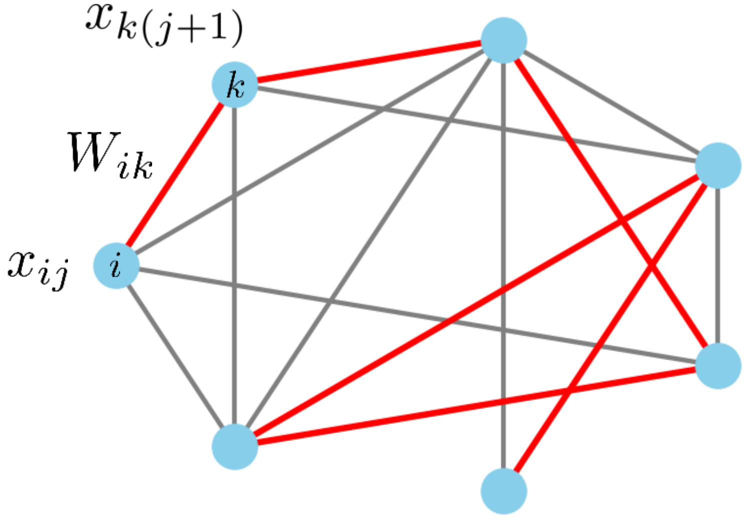
$$x_{ij} = \text{Salesman visits city } i \text{ in step } j.$$

$$\forall i : \sum_j x_{ij} = 1 \qquad \forall j : \sum_i x_{ij} = 1$$

$$H = H_A + H_B$$

$$H_B = B \sum_{ik} W_{ik} \sum_j x_{ij} x_{k(j+1)}$$

$$H_A = A \sum_i \left(1 - \sum_j x_{ij}\right)^2 + A \sum_j \left(1 - \sum_i x_{ij}\right)^2$$



Example: Rail Traffic Management

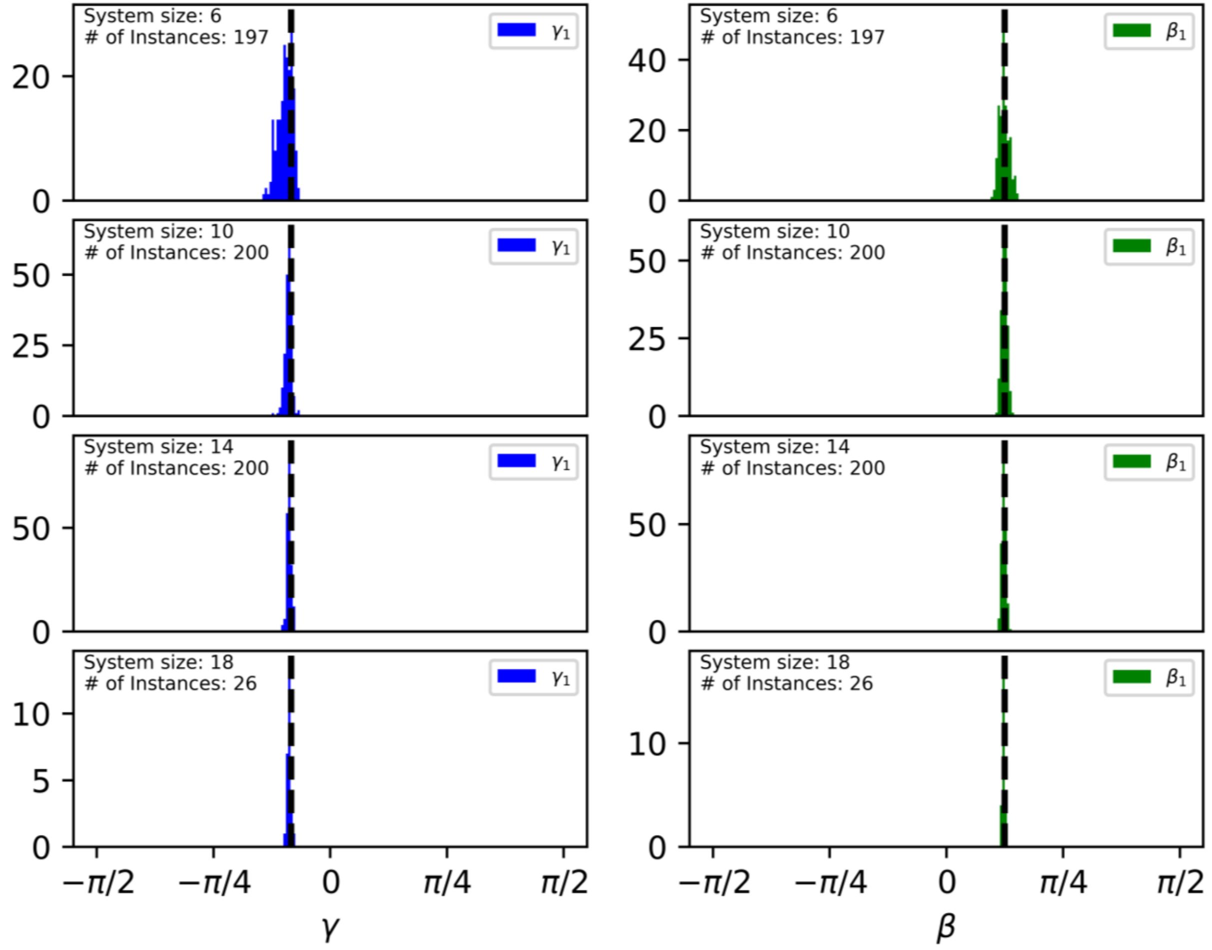
- Trains have starts and destinations
- Various paths collide
- Solution:
  - For each train, generate several routes (e.g., top 5)
  - Each route = node in graph
  - Connect **incompatible** routes
  - Maximum independent set = solution

$$H = -B \sum_i x_i + A \sum_{\text{edges } ij} x_i x_j$$

QAOA Research at IQM

Parameter Concentration

1. Create random 3-regular graph G of size n
2. Define Max-Cut problem on Graph G and set up corresponding QAOA circuit (1 layer)
3. Optimize parameters $\{\beta_1, \gamma_1\}$ with classical optimizer
4. Repeat from 1. and make histogram



QAOA Research at IQM

Parity-encoded QAOA

- Partnership with ParityQC
- Each qubit represents the parity of a pair of variables
- Only local interactions
- Quadratically more qubits → worse performance

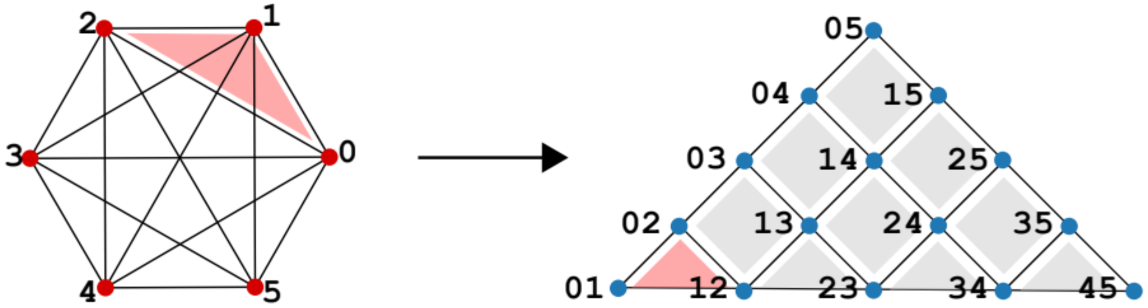


Image: Wybo et al.


Thank you
for tuning in!