A Quantum Algorithm for the Sub-Graph Isomorphism Problem

Nicola Mariella, **Andrea Simonetto**

IBM Research

ENSTA
IP PARIS

February 10, 2022

Setting up the scene

What can we expect from quantum optimization?

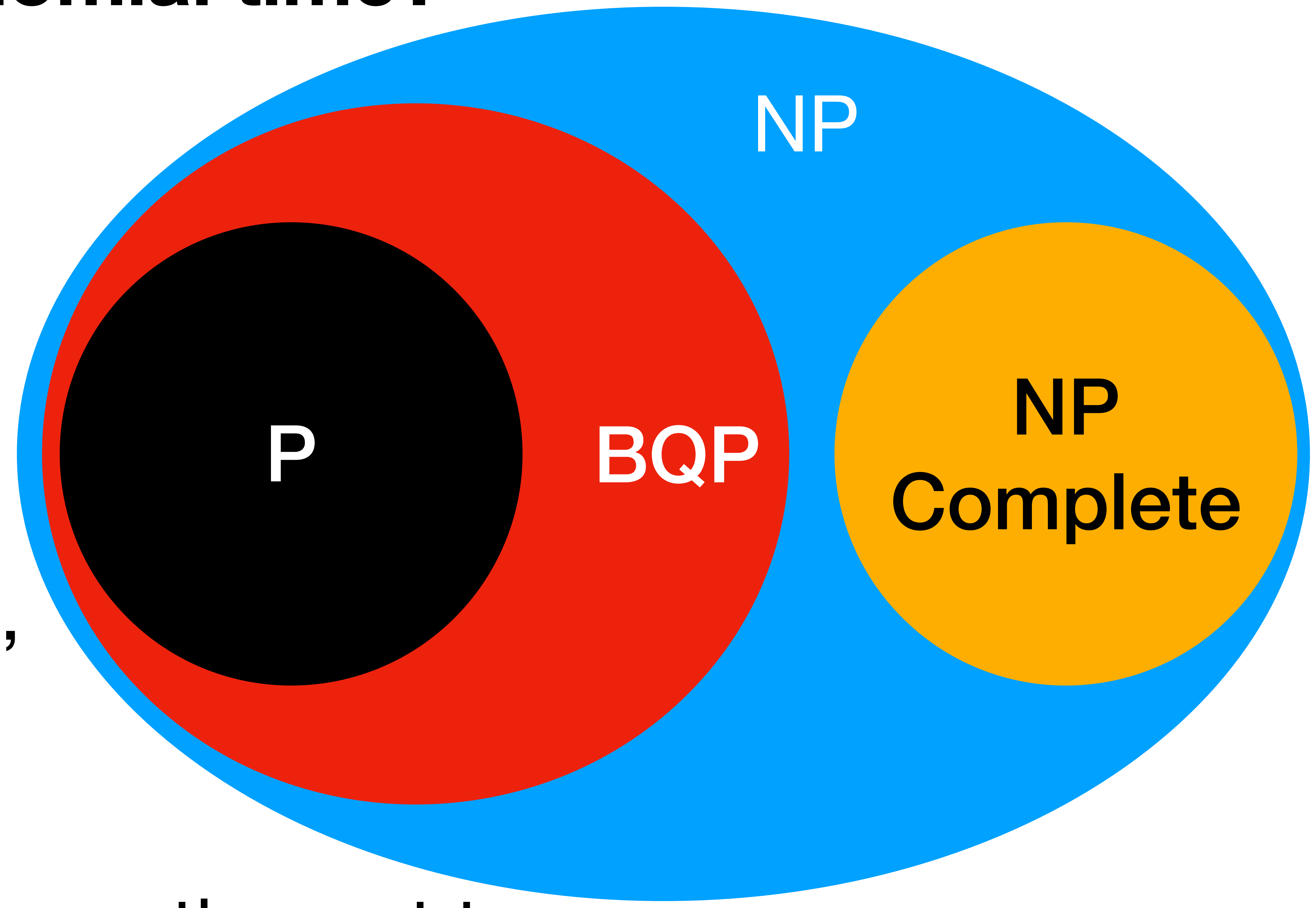
- The basic unit is the qubit and the concept of state
- The state is a unitary vector as $|\psi\rangle \in \mathbb{C}^{2^n}$, e.g., $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \alpha, \beta \in \mathbb{C}, \alpha^2 + \beta^2 = 1$and we can put it in superposition
- Quantum algorithms are quantum circuits (and in particular unitary matrices): $|\psi\rangle = U|\psi_0\rangle$
- The problem with quantum algorithms is to extract what we need (e.g., Grover search)
- Quantum optimization algorithms will design *U* to drive the state towards the desired solution

Demystifying

What can we expect from quantum optimization?

- **Can it solve NP Hard problems in polynomial time?**

Not likely



- BQP: Bounded error quantum polynomial,
- Suspected relationship
- Since we don’t know NP vs. P, we don’t know the rest too…

Demystifying

What can we expect from quantum optimization?

- **Can it solve NP Hard problems “better” (faster, …) than classical algorithms?**

Not likely soon: it’s very hard to do!

Classically we are VERY good: *In the last 25 years, algorithmic advances in integer optimization coupled with hardware improvements have resulted in a 800 billion factor speedup in mixed-integer optimization* (25 years ago a problem that would have required 25 years to run, runs now in 1ms).

We can solve 1000+ variable problems within minutes

Contrast with chemistry (only limited atoms !)

Demystifying

What can we expect from quantum optimization?

- Quantum algorithms for optimization will offer **novel heuristics** for the solution of NP problems, that may (or not) give some advantage, with respect to classical algorithms.
- In terms of: speed/size
- In terms of: search space

QUBOs

Quadratic, unconstrained, binary optimization problems

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

- E.g., max-cut, soft-constrained travelling salesman, portfolio selection, etc..

QUBOs

Quadratic, unconstrained, binary optimization problems

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

- Quantum: encode the problem into a circuit. In the VQE case, an Hamiltonian that encodes all the search space:

- $\{0,1\}^n \ni x \longrightarrow |\psi\rangle \in \mathbb{C}^{2^n}; \qquad H \in \mathbb{C}^{2^n \times 2^n}; \qquad \min_x x^T Q x \longrightarrow \min \text{eig}(H)$

- (+) The encoding is straightforward

- (+) The basis vectors of $|\psi\rangle$ are the bitstrings of all the possible combinations

QUBOs

Quadratic, unconstrained, binary optimization problems

- Quantum: encode the problem into a circuit. In the VQE case, an Hamiltonian that encodes all the search space:
- $\{0,1\}^n \ni x \longrightarrow |\psi\rangle \in \mathbb{C}^{2^n}; \qquad H \in \mathbb{C}^{2^n \times 2^n}; \qquad \min_x x^T Q x \longrightarrow \min \text{eig}(H)$
- **Then design a trial function that probes the Hilbert space:**
- $|\psi(\theta)\rangle = U(\theta)|\psi_0\rangle \longrightarrow \min_x x^T Q x \approx \min_{\theta \in \mathbb{R}^D} \langle \psi(\theta)|H|\psi(\theta)\rangle = \min_{\theta \in \mathbb{R}^D} \langle \psi_0|U(\theta) H U(\theta)|\psi_0\rangle$

VQE

Variational Quantum Eigensolver

- Example:

On quantum hardware

- $$\theta^+ = \theta - \alpha \tilde{\mathbb{E}}[\nabla_\theta \langle \psi_0 | U(\theta) H U(\theta) | \psi_0 \rangle]$$

On classical hardware

- Dimension *D* dictates approximation level and circuit depth
- This is a (classical) stochastic black-box, non-convex, continuous optimization problem (and NP-Hard)

VQE

Variational Quantum Eigensolver

- Strengths:
  - one-fits-all scheme (see Qiskit optimization module)
  - Can be extended (heuristically and with some acrobatics) to constrained problems, e.g. via operator splitting
  - Can be extended to polynomial optimization (allowing high-order interconnection: careful here)
  - It’s insane (= non simulatable) classically (= complete enumeration of all feasible space)

VQE

Variational Quantum Eigensolver

- Strengths:
  - It’s insane (= non simulatable) classically (= complete enumeration of all feasible space)
  - You are enumerating all the possibilities, put them in superposition and ``trying them all at once”. Then you scan the space by rotations…
    $|\psi\rangle = \alpha_0|0000\rangle + \alpha_1|0001\rangle + \alpha_2|0010\rangle + \alpha_3|0011\rangle + \ldots + \alpha_{15}|1111\rangle$

VQE

Variational Quantum Eigensolver

- Weaknesses:
  - Any advantage? Not clear at this point
  - It scales terribly: e.g., in network problems if *N* is the number of vertices, then we need $n = O(N^2)$ qubits
  - Ex: for vehicle routing: Classically we are exact up to 250 vehicles, and approximate with guarantees till 1000+; quantum-ly, VQE offers tops 10 vehicles with an heuristic

The bigger picture: it's all about encoding

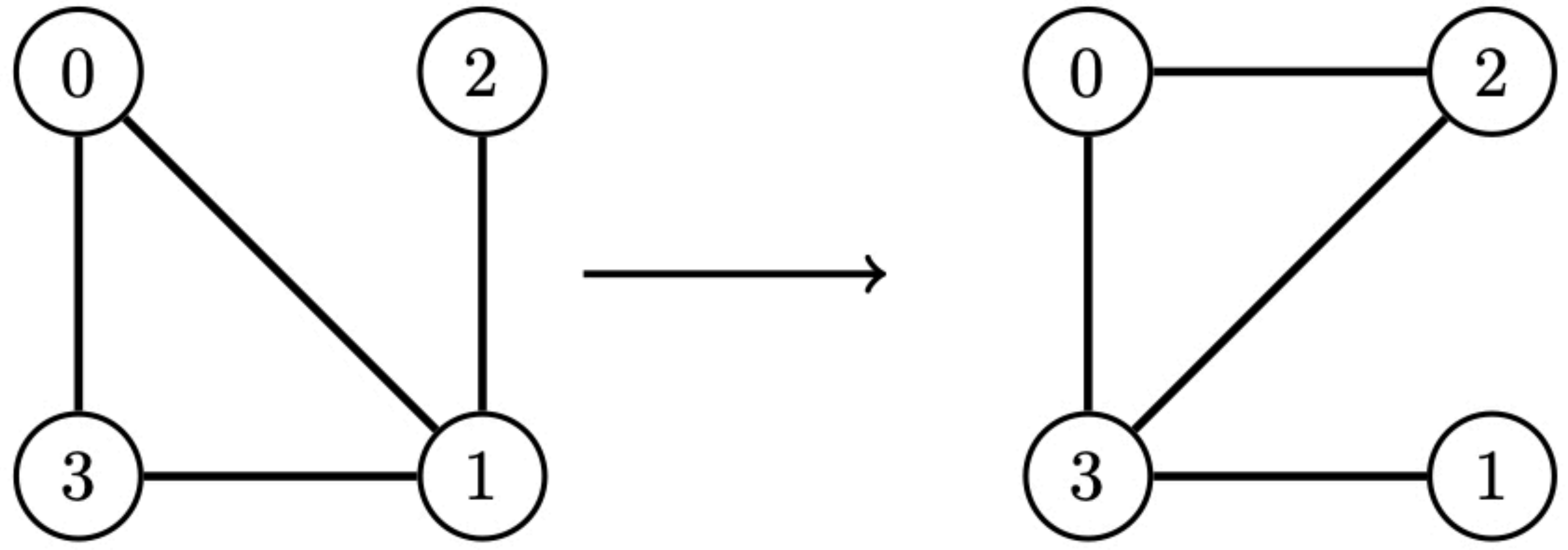
Big strokes in the sky

- Encode the problem in a quantum circuit
- Define the ansatz (parametric circuit) to probe the solution space
- Iterate: quantum/ classical
- $$\min_x f(x) \longrightarrow \min_P V(P) \approx \min_\theta V(P(\theta))$$
- VQE is not the right encoding, can we build better ones?

Finding clues in a specific problem

Sub-graph isomorphism problem/ Graph isomorphism problem

- Focus on the graph isomorphism for simplicity



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

- $\min_{P \in \Pi} ||PAP^T - B||_F^2$

- *A, B* are the adjacency matrices of the graphs of dimension *N, P* is a permutation matrix

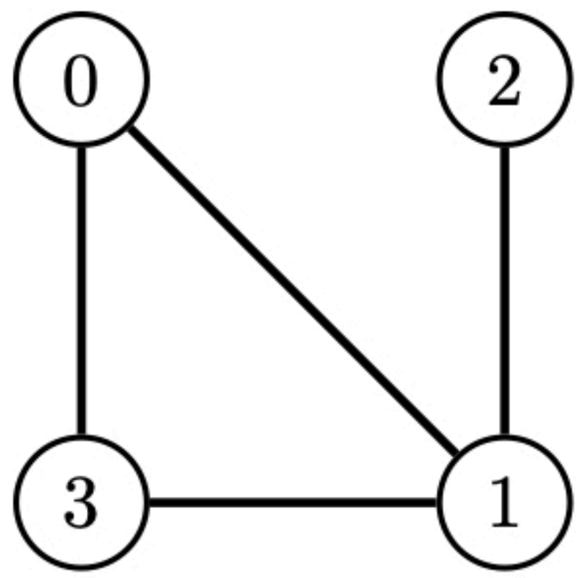
- The problem is finding permutations

Finding clues in a specific problem

Step 1: Encoding

- Encoding: define a transformation, that maps the adjacency matrices into unitary matrices as follows
- $\hat{A} = H^{\otimes(2k+1)} \left(\mathbb{I}_{N^2} \oplus \sum_{i,j} (-1)^{A_{i,j}} |i,j\rangle \langle i,j| \right) H^{\otimes(2k+1)}$

$\mathrm{cexp}(h(A))$



$$\longleftrightarrow \quad A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \quad \longleftrightarrow \quad \widehat{A} = H^{\otimes 5} \begin{pmatrix} \mathbb{I}_{16} & \\ & \begin{pmatrix} 1 & & & & \\ & -1 & & & \\ & & 1 & & \\ & & & -1 & \\ & & & & \ddots \end{pmatrix} \end{pmatrix} H^{\otimes 5}$$

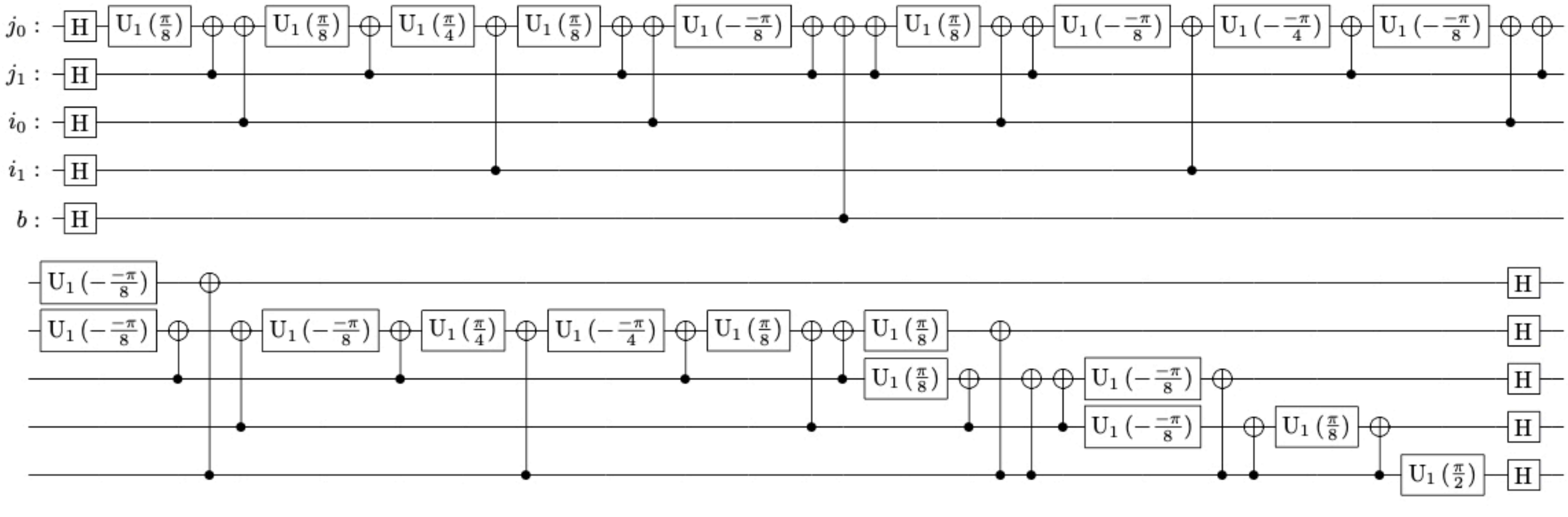
Finding clues in a specific problem

Step 1: Encoding

- Such hat transformation requires $2\log_2(N) + 1 = 2k + 1$ qubits, with *N* the number of vertices (rem, QUBO requires $O(N^2)$)
- **Proof:**
$\text{cexp}(h(A)) \in \{-1,1\}^{2N^2}, \quad H^{\otimes n} \in \mathbb{U}(2^n), \qquad \implies n = 2\log_2(N) + 1$
- And, $\hat{A} \in \mathbb{U}(2^{2k+1})$

Finding clues in a specific problem

Step 1: Encoding



Finding clues in a specific problem

Step 1: Encoding

- The hat transformation has a number of useful algebraic properties, from which, **After some heavy algebra**

Theorem

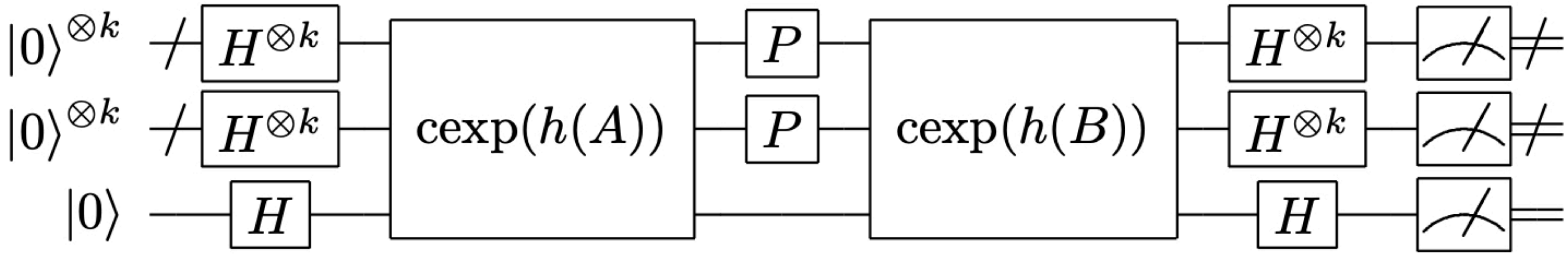
Let: $\check{P} = I_2 \otimes (H^{\otimes 2k} P^{\otimes 2} H^{\otimes 2k}) \in \mathbb{U}(2^{2k+1})$

Then: $PAP^T - B \equiv \check{P}\hat{A}\check{P}^T\hat{B}$

Finding clues in a specific problem

Step 1: Encoding

- The cost then, can be evaluated via a quantum circuit



- $\min_{P \in \Pi} \|PAP^T - B\|_F^2 \quad \equiv \min_{P \in \Pi} \langle \psi | \check{P}\hat{A}\check{P}^T\hat{B} | \psi \rangle$

Finding clues in a specific problem

Step 2: Ansatz design

- Then, one design an ansatz (a parametric circuit, changeable by rotations) to search in the space of permutation matrices as such
- $\min_{P\in\Pi} \|PAP^T - B\|_F^2 \quad \equiv \min_{P\in\Pi}\langle\psi|\check{P}\hat{A}\check{P}^T\hat{B}|\psi\rangle \approx \min_{\theta\in\mathbb{R}^D} \langle\psi|\tilde{P}_\theta\hat{A}\tilde{P}_\theta^T\hat{B}|\psi\rangle$
- Then it's just SGD plus a quantum circuit evaluation !

Finding clues in a specific problem

Step 2: Ansatz design

- Our choice of design is $\tilde{P}_\theta = \prod_{i=1}^{D} R_{P_i}(\theta)$
- $R_{P_i}(\theta)$ are rotations about the permutation *Pi,* and for which

$$\prod_{i=1}^{D} R_{P_i}(m_i \pi) = \prod_{i=1}^{D} P_i^{m_i}, \qquad m_i \in \mathbb{Z}$$

- We span the permutation space with a permutation basis, and allowing for unfeasibility

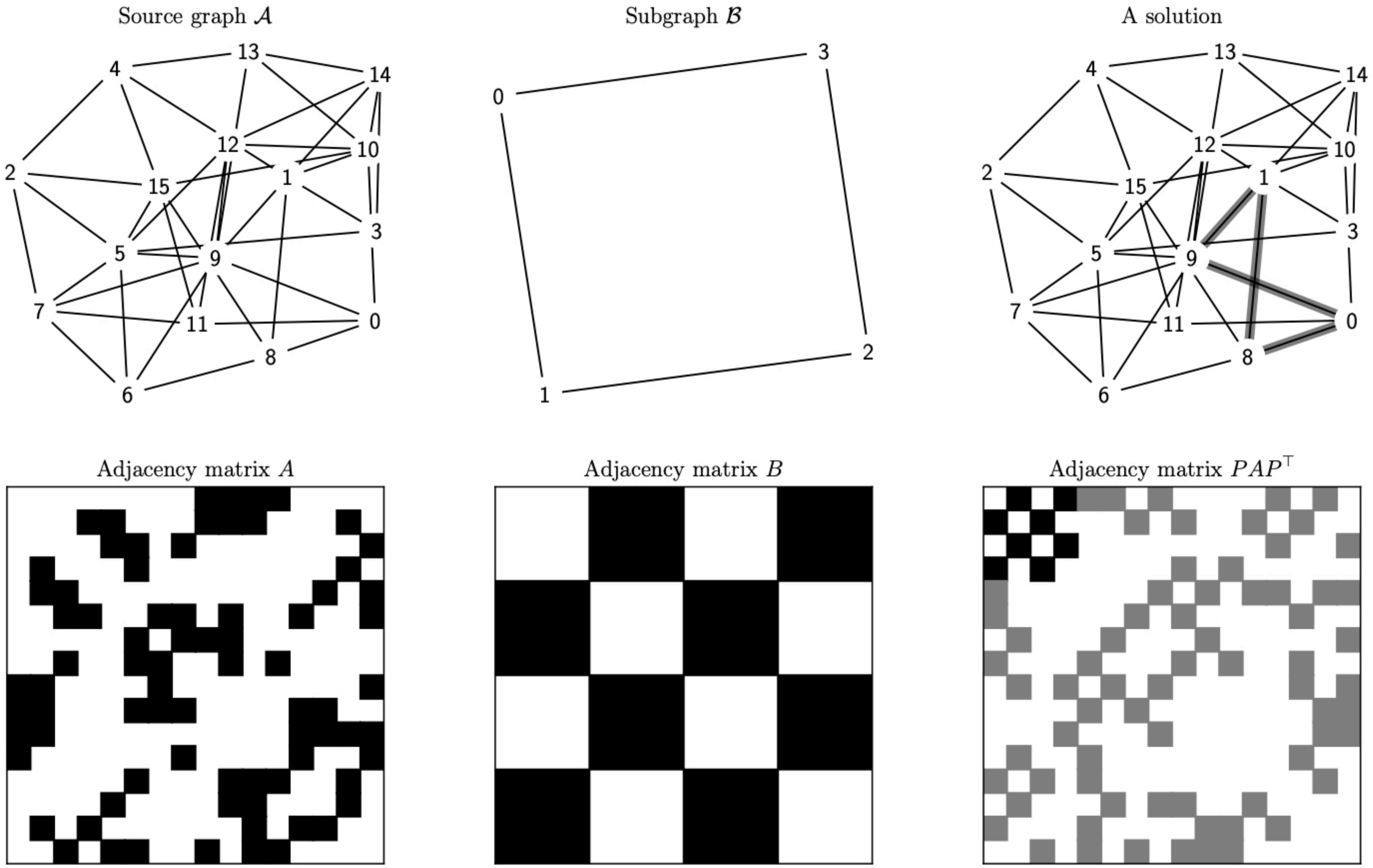
Finding clues in a specific problem

Step 3: Quantum Algorithm

1. Initialize $\theta \in \mathbb{R}^D, \quad |\psi\rangle \in \mathbb{C}^{2^{2k+1}}$ choose an ansatz $\tilde{P}_\theta$
2. Quantum-part : Evaluate the cost/ gradient
   $\langle\psi|\tilde{P}_\theta \hat{A} \tilde{P}_\theta^T \hat{B}|\psi\rangle, \qquad \tilde{\mathbb{E}}[\nabla_\theta \langle\psi|\tilde{P}_\theta \hat{A} \tilde{P}_\theta^T \hat{B}|\psi\rangle]$
3. Classical-part : SGD: $\theta^+ = \theta - \gamma \tilde{\mathbb{E}}[\nabla_\theta \langle\psi|\tilde{P}_\theta \hat{A} \tilde{P}_\theta^T \hat{B}|\psi\rangle]$
4. Probabilistic rounding (in parallel, to map solution to actual permutations)

Finding clues in a specific problem

Sub-graph isomorphism problem/ Graph isomorphism problem



Finding clues in a specific problem

Sub-graph isomorphism problem/ Graph isomorphism problem

- (+) logarithmic scaling in the number of nodes: 100 qubit can encode a $O(10^{15})$ node graph, vs. the best 1M graph classically
- (-) the circuit depth is still proportional to the number of nodes and edges.. (link to the importance of good compilation!)
- (-) it is still an heuristic (dimension *D*)

Extensions?

It's all about algebra and well-played creativity

- It works similarly for max-cut problems
- Generalizations are still unknown

- Different encodings give rise to **different building blocks for optimization**
- Hat transformation seems to indicate to build quantum optimization from matrices and permutations…
- It's exciting: **building optimization from the ground up**

References

A good start

- N. Moll et al., *Quantum optimization using variational algorithms on near-term quantum devices,* 2017 [arXiv: 1710.01022]
- M. Rancic, *An exponentially more efficient optimization algorithm for noisy quantum computers*, 2021 [arXiv: 2110.10788]
- **N. Mariella, A.S., *A Quantum Algorithm for the Sub-Graph Isomorphism Problem,* 2021 [arXiv: 2111.09732]**