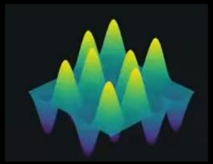
QAOA

The Quantum Approximate Optimization Algorithm

Matthias Pawlik

Presentation QAR-Lab 10.04.2024

Classification of the QAOA

- The QAOA is a quantum algorithm that solves optimization problems with arbitrary accuracy
- The QAOA is a heuristic quantum algorithm that may produce acceptable solutions fast in practice
  However, in the worst-case scenario, the algorithm might still scale exponentially
- The QAOA is classified as a variational algorithm which can be thought of as an approximation to adiabatic quantum computation
- Derivation of the QAOA in three steps: I. VQE II. QUBO III. AQC $\longrightarrow$ QAOA 
- Key idea of the quantum optimization algorithm:

Reformulate the optimization problem in terms of a Hamiltonian whose ground state corresponds to the optimal solution

I. Variational Quantum Eigensolvers (VQE)

- Energy expectation value of a pure state $|\Psi\rangle \in \mathcal{H}$: $E = \langle\Psi|H|\Psi\rangle$ with Hamiltonian $H = H^\dagger$
- Approximate the total Hilbert space with a family of quantum states $|\Psi(\theta)\rangle$ that depend on a variational parameter θ
- Find the optimal parameter θ^* as the approximate ground state $|\Psi^*\rangle$ of the quantum system with lowest energy E^*:

$$|\Psi^*\rangle = \underset{|\Psi(\theta)\rangle \in \mathcal{H}}{\operatorname{argmin}} \langle\Psi(\theta)|H|\Psi(\theta)\rangle \qquad \theta^* = \underset{\theta}{\operatorname{argmin}} \langle\Psi(\theta)|H|\Psi(\theta)\rangle \qquad E^* = \langle\Psi(\theta^*)|H|\Psi(\theta^*)\rangle$$

- With the help of Variational Quantum Circuits one can determine upper bounds to the ground state energy:

prepare quantum state

$|\Psi(\theta)\rangle = U(\theta)|\Psi_0\rangle$

$\boxed{U(\theta)}$

→ energy measurement $E(\theta)$ →

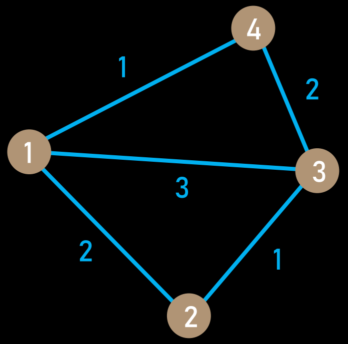
← parameter update $\theta_i \to \theta_{i+1}$ ←

Classical Optimizer

II. Quadratic Unconstrained Binary Optimization Problem (QUBO)

- Optimization problem with quadratic objective function: $x^T Q x + c^T x$ with $Q \in \mathbb{R}^{nxn}$ and $c \in \mathbb{R}^n$
- Minimize the expression with binary-valued vectors $x \in \{0,1\}^n$ and no further constraints
- Example: MaxCut problem

edges E vertices V



graph G(V,E) $\longrightarrow$ output: maximal cut $x = [x_1 x_2 x_3 x_4]$

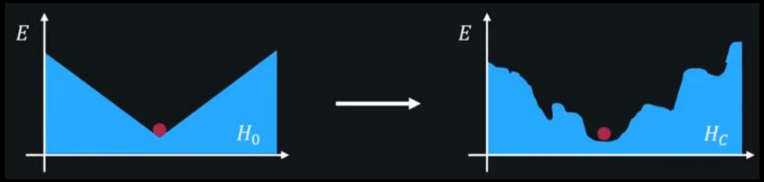
weight matrix $W = \begin{pmatrix} 0 & 2 & 3 & 1 \\ 2 & 0 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{pmatrix}$

- Maximize the cost function $C(x) = \sum_{i,j=1}^{4} W_{ij} x_i (1 - x_j)$
- Reformulate the MaxCut problem into a QUBO: $Q_{ij} = -W_{ij}$ and $c_i = \sum_{j=1}^{4} W_{ij}$
- Encode the cost function into a Hamiltonian operator $H_C = \sum_{i,j=1}^{4} \frac{1}{4} Q_{ij} Z_i Z_j - \sum_{i=1}^{4} \frac{1}{2}(c_i + \sum_{j=1}^{4} Q_{ij}) Z_i$ $+ const.$

  with Schrödinger equation $H_C|x\rangle = C(x)|x\rangle$

III. Adiabatic Quantum Computing (AQC)

- Evolution of a quantum system: $H|\Psi(t)\rangle = i\hbar\partial_t|\Psi(t)\rangle$ Schrödinger equation
- Solution for time independent H: $|\Psi(t)\rangle = e^{-iHt/\hbar}|\Psi(0)\rangle = U(t)|\Psi(0)\rangle$
- Adiabatic theorem: If the Hamiltonian of a quantum system in the ground state is perturbed slowly enough, the quantum system remains in its ground state



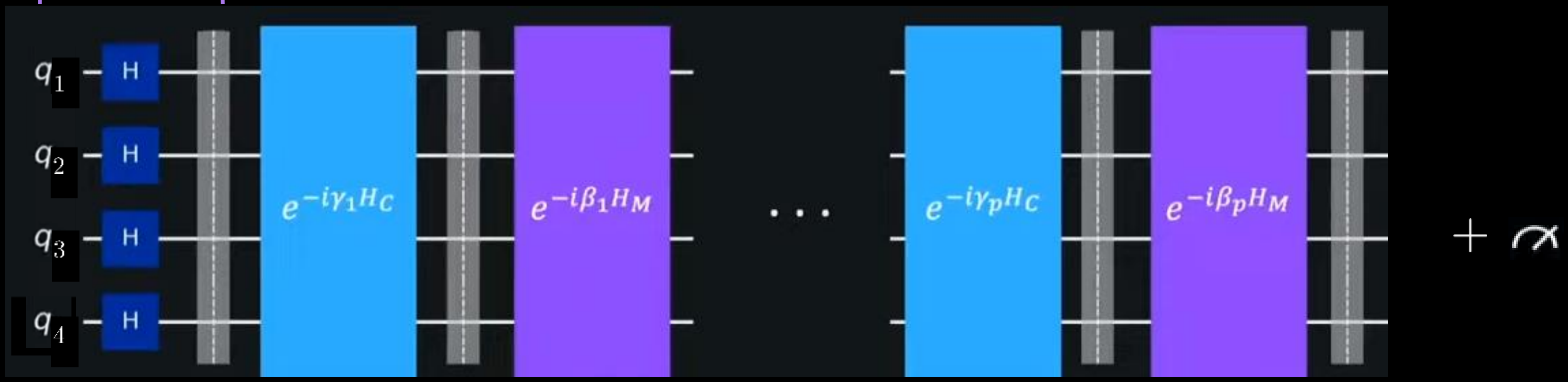
- Trotterization: Discretization of the time evolution operator of the Hamiltonian $H_C = H_1 + H_2$ by applying the Trotter Suzuki Formula $e^{-i(H_1+H_2)t/\hbar} \approx (e^{-iH_1t/\hbar r}e^{-iH_2t/\hbar r})^r$
- The QAOA is an adiabatic schedule: The initial quantum state is an easy-to-prepare eigenstate of the so-called Mixer Hamiltonian H_M which is adiabatically changed into the ground state of the Cost Hamiltonian H_C

$H(t) = \frac{t}{T}H_C + (1-\frac{t}{T})H_M$ for a total run time T

IV. Quantum Approximate Optimization Algorithm (QAOA) - Pattern

- The QAOA was first introduced by Farhi, Goldstone and Gutmann in 2014
- Idea: Quantum optimization algorithm to find an approximate solution for QUBO instances
- QAOA as a special case of VQE: layerized variational form based on a trotterized adiabatic process

preparation of equal superposition state $|+\rangle^n = \sum_{x \in \{0,1\}^n} \frac{1}{\sqrt{2^n}} |x\rangle$ as the highest eigenenergy state of $H_M = \sum_{i=1}^{n} X_i$



$U_C(\gamma_i) = e^{-i\gamma_i H_C}$

encoding of the specific optimization problem

$U_M(\beta_i) = e^{-i\beta_i H_M}$

mixing through single rotational gates

p layers, variational parameters $\gamma_1 \cdots \gamma_p$ and $\beta_1 \ldots \beta_p$

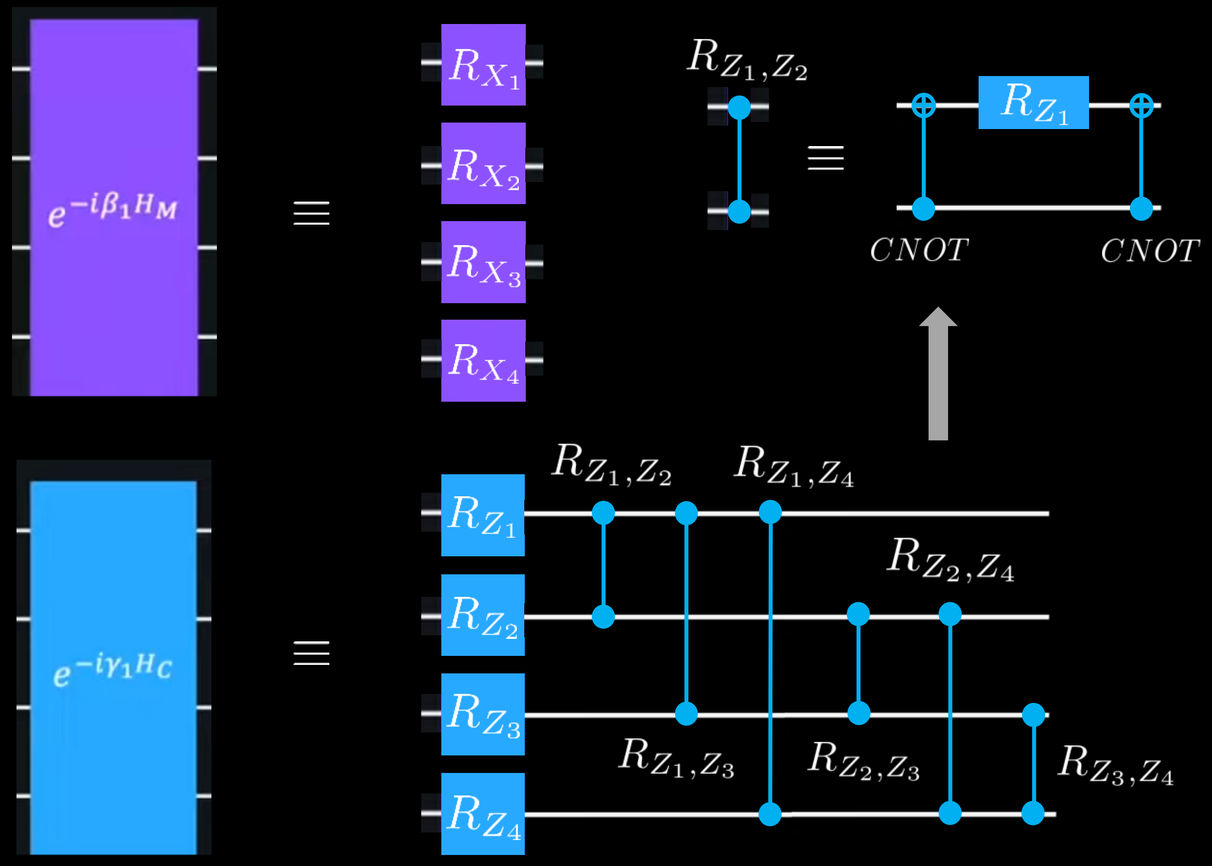
IV. Quantum Approximate Optimization Algorithm (QAOA) – Gates

Time evolution operator of the Mixer Hamiltonian

$$e^{-i\beta_1 H_M} = \prod_{i=1}^{4} e^{-i\beta_1 X_i} = \prod_{i=1}^{4} R_{X_i}(\beta_1)$$

Time evolution operator of the Cost Hamiltonian

$$e^{-i\gamma_1 H_C} = \prod_{i,j=1}^{4} R_{Z_i,Z_j}\left(\tfrac{1}{4}Q_{ij}\gamma_1\right) \cdot \prod_{i=1}^{4} R_{Z_i}\left(-\tfrac{1}{2}\left(c_i + \sum_{j=1}^{4} Q_{ij}\right)\gamma_1\right)$$



IV. Quantum Approximate Optimization Algorithm (QAOA) – Implementation

- Implementation of the QAOA algorithm in practice (Qiskit Runtime):

Setup: Import of numpy, qiskit libraries, qiskit runtime, rustworkx graph and SciPy libraries

Step 1 Map classical inputs to a quantum problem with rustworkx library → construct Cost Hamiltonian → QAOAAnsatz(Hamiltonian, repetitions) → verify initialization with circuit plotting features

Step 2 Optimize problem for quantum execution: transpiler reduces the overall gate count needed to run the quantum algorithm on the hardware (less error rates and decoherence over time)

Step 3 Execute using Qiskit Primitives (Estimator): define the cost function(parameters, ansatz, Hamiltonian, estimator) over which to minimize → use classical optimizer routine (e.g. SciPy minimize)

Step 4 Post-process, return result in classical format: plug in solution vector of parameters into the ansatz circuit → Obtain probability distribution of the most probable bit-strings → draw solution cut

Sources

- QST Lecture Notes Quantum Information (Adiabatic and Variational Algorithms): Chapter Quantum Algorithms
- IBM Quantum Learning: https://learning.quantum.ibm.com/tutorial/quantum-approximate-optimization-algorithm
- Qiskit QML 2021 Summer School Lecture 5.2 – Introduction to the Quantum Approximate Optimization Algorithm and Applications: https://www.youtube.com/watch?v=YpLzSQPrgSc&t=647s