Introduction to QAOA

Quantum Finance Meetings

QAOA = Quantum Approximate Optimization Algorithm

Introduced in **arXiv:1411.4028** by Farhi, Goldstone and Gutmann

It is a variational algorithm

Hybrid – i.e. partially quantum, partially classical

Motivated by the Adiabatic Theorem

Convergence ensured* also by the Adiabatic Theorem

Very useful to tackle combinatorial optimization problems

Variational Algorithm

Optimization problem with the solution encoded in the minimum or maximum of some observable

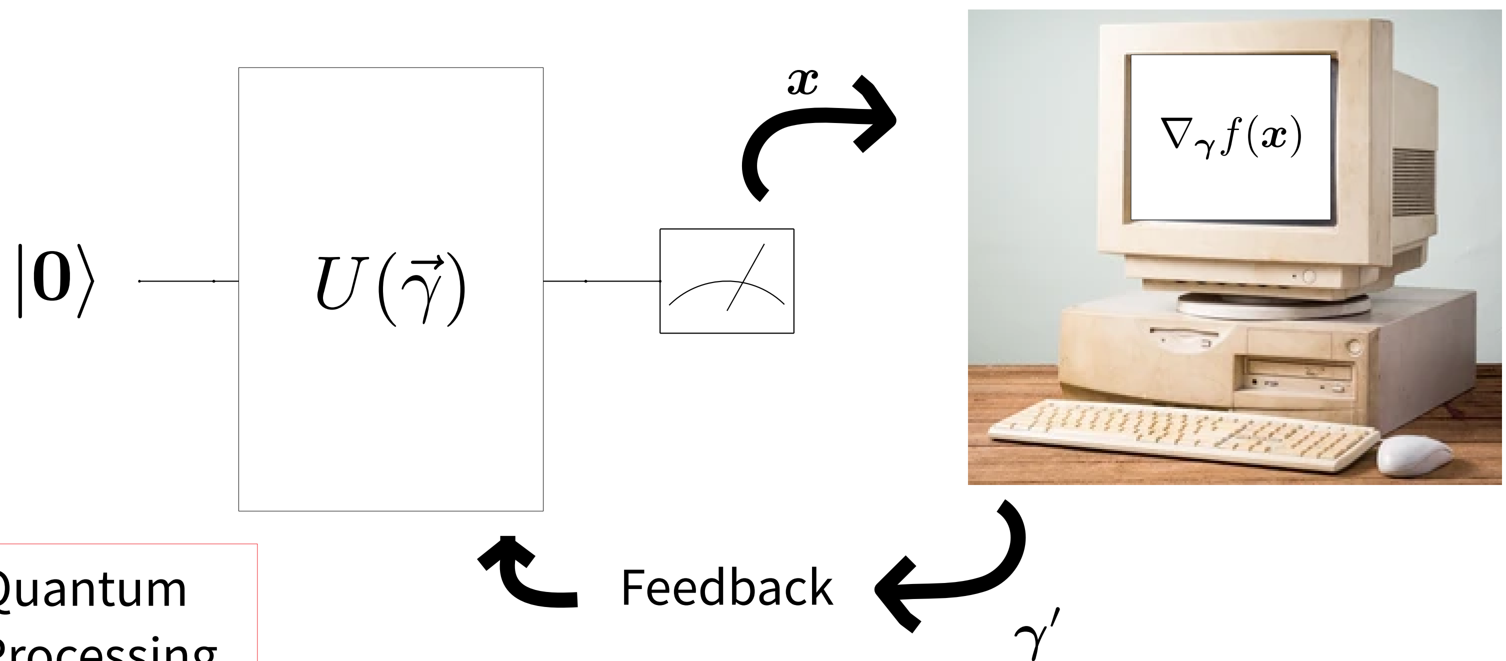
The strategy is to provide a “good” family of **parametric** states in the hope that they include the solution space for some values of the parameter

$$\{\, |\psi(\boldsymbol{\gamma})\rangle \,\}_{\boldsymbol{\gamma}}$$

$$F^* = \min_{\boldsymbol{\gamma}} \langle \psi(\boldsymbol{\gamma}) | H_C | \psi(\boldsymbol{\gamma}) \rangle$$

Hybrid Algorithm:

Partially quantum, partially classical

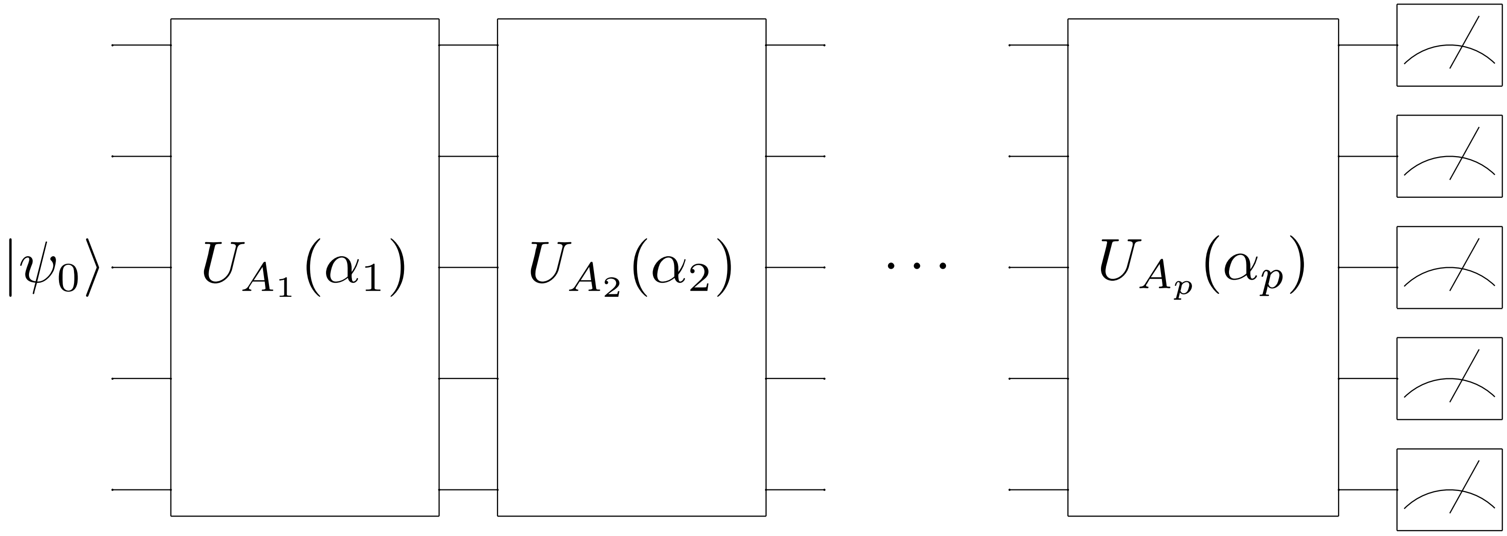


Classical optimization

Quantum Processing

The idea for parametrization:

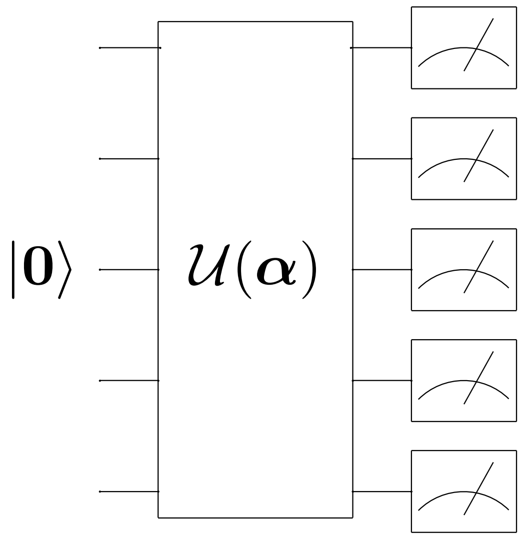
State preparation by a sequence of parameter-dependent unitaries



$$U_{A_k}(\alpha) = \exp(-i\, A_k\, \alpha_k) \qquad |\psi(\boldsymbol{\alpha})\rangle = U_{A_p}(\alpha_p)\ldots U_{A_1}(\alpha_1)|\psi_0\rangle$$

The idea for parametrization:

State preparation by a sequence of parameter-dependent unitaries



$$|\psi(\boldsymbol{\alpha})\rangle = U_{A_p}(\alpha_p)\ldots U_{A_1}(\alpha_1)|\mathbf{0}\rangle$$

$$F_p^* = \min_{\boldsymbol{\alpha}} \langle\psi(\boldsymbol{\alpha})|H_C|\psi(\boldsymbol{\alpha})\rangle$$

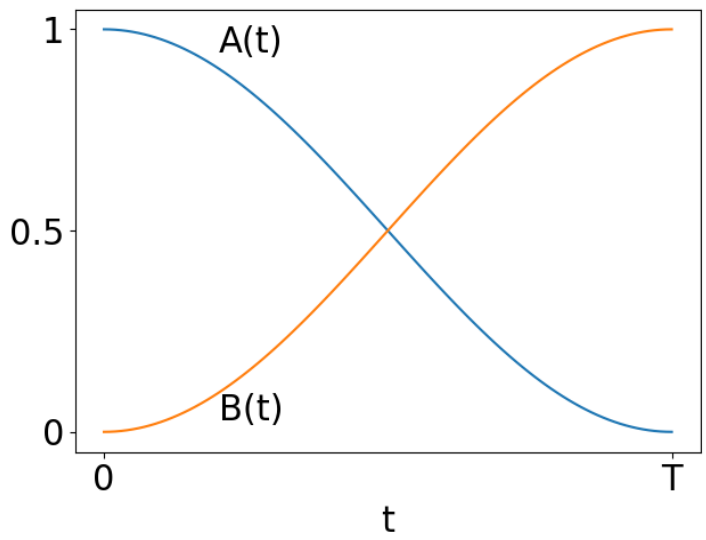
Optimization with p+1 layers include p layers

$$U_{A_k}(0) = \mathbb{1} \;\Rightarrow\; F_{p+1}^* \leq F_p^*$$

Establishing A_i

Inspiration from Adiabatic Quantum Computing

$$H = A(t)\, H_M + B(t)\, H_C$$



Adiabatic Theorem:

For a sufficiently slow transition from H_M to H_C, the system remains at the corresponding eigenstate at all times

Establishing A_i

Inspiration from Adiabatic Quantum Computing

$$H = A(t)\, H_M + B(t)\, H_C$$

The idea:

Encode the solution of the problem in the lowest/highest eigenstate of H_C

Use H_M with a known and easy-to-prepare ground state

Use the adiabatic theorem to go from H_M to the solution of the problem

Establishing A_i – Connecting to our parametric description

Trotterization – Approximate commutativity for small evolutions

$$\exp(A+B) = \lim_{p\to\infty} \left(e^{A/p}\, e^{B/p}\right)^p$$

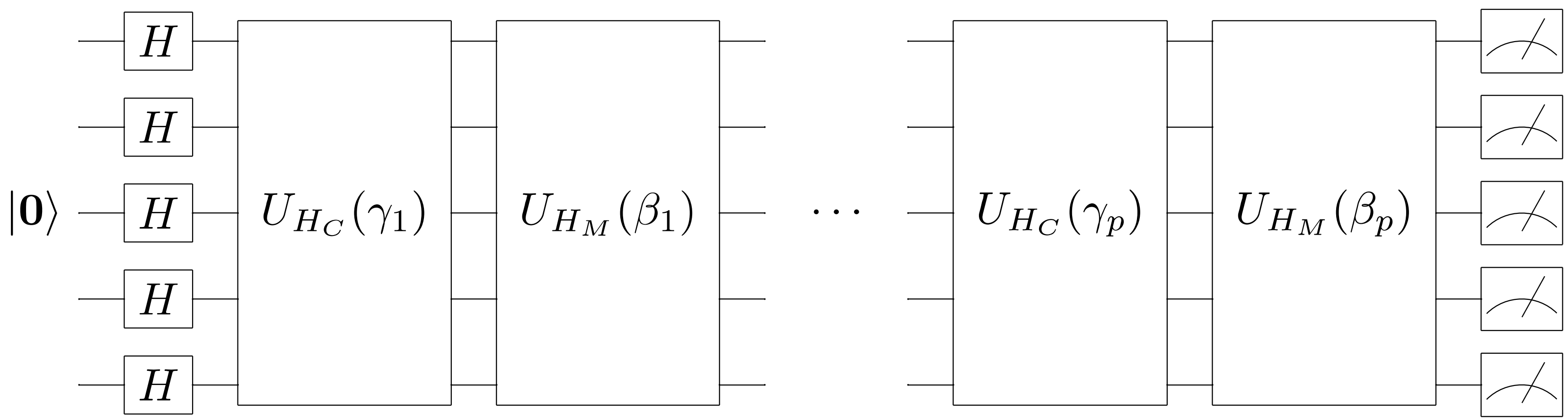
$$A \to H_M \qquad B \to H_C$$

$$\begin{aligned} U(t,\, t+\delta t) &\approx e^{-i\,(A(t)\, H_M\, \delta t + B(t)\, H_C\, \delta t)} \\ &\approx e^{-i\, A(t)\, H_M\, \delta t}\, e^{-i\, B(t)\, H_C\, \delta t} \end{aligned}$$

$$U(0,\, T) = \prod_{i=1}^{N} U(t_i,\, t_i + \delta t_i) \longrightarrow \mathcal{U}(\boldsymbol{\alpha}) = U_{H_C}(\alpha_p) \ldots U_{H_C}(\alpha_2) U_{H_M}(\alpha_1)$$

The QAOA prescription

Alternation between H_M and H_C



$$\boldsymbol{\alpha} \to (\boldsymbol{\gamma}, \boldsymbol{\beta})$$

$$|\psi_0\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n}$$

$$|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = U_{H_M}(\beta_p)\ldots U_{H_C}(\gamma_1)|\psi_0\rangle$$

“Canonical” application – Combinatorial Optimization

Optimization over n binary variables

$$C^* = \max_{\boldsymbol{x}} C(x_1, \ldots, x_n)$$

$$x_i \in \{0,\ 1\}$$

e.g.

$$C(x_1,\ x_2,\ x_3) = (x_1 \vee x_2) \wedge x_3 \ \longrightarrow\ x_3 = 1$$

x_1	x_2	$\mathrm{C}(\mathrm{x}_3 = 1)$
0	0	0
0	1	1
1	0	1
1	1	1

Brute force method: 2^n evaluations of C required

“Canonical” application – Combinatorial Optimization

Optimization over n binary variables

$$C^* = \max_{\boldsymbol{x}} C(x_1, \ldots, x_n)$$

$$x_i \in \{0,\ 1\}$$

The idea for quantization: Each variable is mapped to a qubit

$$x_i \to \hat{x}_i = |1\rangle\langle 1|_i = \frac{\mathbb{1} - Z_i}{2}$$

Thus: $\hat{x}_i|x_1, \ldots, x_n\rangle = x_i\,|x_1, \ldots, x_n\rangle$

“Canonical” application – Combinatorial Optimization

Specific class – Quadratic unconstrained binary optimization (QUBO)

$$C(x_1, \ldots, x_n) = \sum_{i,j=1}^{n} Q_{ij}\, x_i\, x_j + \sum_{i=1}^{n} L_i\, x_i \qquad (\, x_i \in \{0,\ 1\}\,)$$

As a quantum operator:

$$H_C = \sum_{i,j=1}^{n} Q_{ij} \hat{x}_i\, \hat{x}_j + \sum_{i=1}^{n} L_i \hat{x}_i$$

“Canonical” application – Combinatorial Optimization

Specific class – Quadratic unconstrained binary optimization (QUBO)

$$C(x_1, \ldots, x_n) = \sum_{i,j=1}^{n} Q_{ij}\, x_i\, x_j + \sum_{i=1}^{n} L_i\, x_i \qquad (\, x_i \in \{0,\ 1\}\,)$$

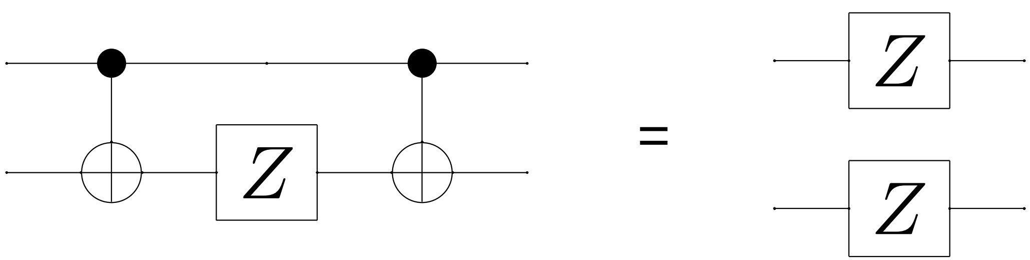
As a quantum operator:

$$H_C = \sum_{i,j=1}^{n} Q_{ij} \hat{x}_i\, \hat{x}_j + \sum_{i=1}^{n} L_i \hat{x}_i \longrightarrow H_C|\boldsymbol{x}\rangle = C(x_1, \ldots, x_n)|\boldsymbol{x}\rangle$$

Extra slides

The “trick” to implement multiple-qubit rotations

For two qubits: Convert a ZZ rotation into a single-Z rotation

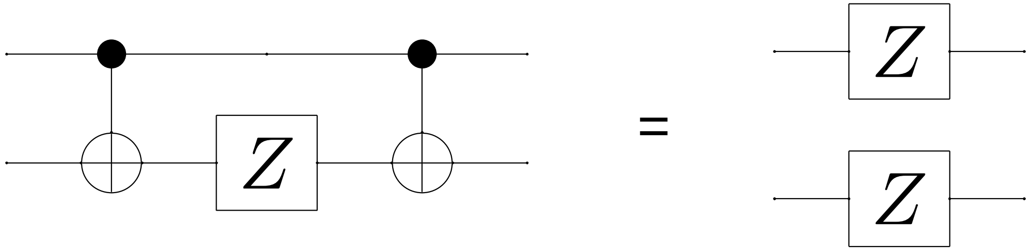


$$\color{darkred}{\mathrm{CNOT}_{ij} = |0\rangle\langle 0|_i \otimes \mathrm{I} + |1\rangle\langle 1|_i \otimes Z_j}$$

$$\mathrm{CNOT}_{ij}\, Z_j\, \mathrm{CNOT}_{ij} = |0\rangle\langle 0|_i \otimes Z_j + |1\rangle\langle 1|_i \otimes (X_j Z_j X_j)$$

The “trick” to implement multiple-qubit rotations

For two qubits: Convert a ZZ rotation into a single-Z rotation



$$\mathrm{CNOT}_{ij} = |0\rangle\langle 0|_i \otimes \mathrm{I} + |1\rangle\langle 1|_i \otimes Z_j$$

$$\mathrm{CNOT}_{ij}\, Z_j\, \mathrm{CNOT}_{ij} = |0\rangle\langle 0|_i \otimes Z_j - |1\rangle\langle 1|_i \otimes Z_j$$

The “trick” to implement multiple-qubit rotations

Using the result in a rotation

Given

$$U_\alpha = \exp(i\,\alpha\,Z_i \otimes Z_j) = \cos(\alpha)\,\mathrm{I} + i\ \sin(\alpha)\,Z_i \otimes Z_j$$

Then

$$U_\alpha = \mathrm{CNOT}_{ij}\,[\cos(\alpha)\,\mathrm{I} + \sin(\alpha)\,Z_j]\,\mathrm{CNOT}_{ij}$$

For n qubits: Cascaded application of CNOTs