Introduction to QAOA

Ruihao Li

Case Western Reserve University, USA

QOSF Mentorship Meeting

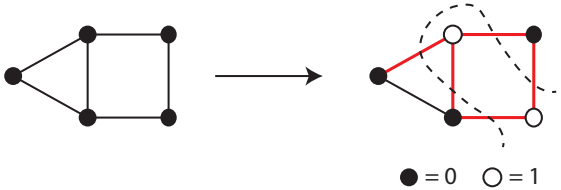
04/29/2022

(Weighted) MaxCut

- Graph: Set of *vertices* (V) connected by (weighted) *edges* (E)
- MaxCut: Partition of vertices into two disjoint subsets (labeled by 0 and 1), such that the total *weight* of the edges between the two subsets is maximized
- For equal-weight edges, the goal is simply to maximize the number of edges connecting the two subsets.

(Weighted) MaxCut

- Graph: Set of *vertices* (V) connected by (weighted) *edges* (E)
- MaxCut: Partition of vertices into two disjoint subsets (labeled by 0 and 1), such that the total *weight* of the edges between the two subsets is maximized
- For equal-weight edges, the goal is simply to maximize the number of edges connecting the two subsets.



Input:
(Weighted) graph $G = (V, E)$

Output:
Maximum cut $x \in (0, 1)^n$
$x = [0, 1, 0, 1, 0]$

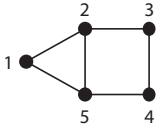
QUBO

- Quadratic Programs: Optimize (maximize or minimize) a quadratic objective function subject to linear constraints on the variables
  e.g., minimize $\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}$
  subject to $A\mathbf{x} \leq \mathbf{b}$
- Special case: Quadratic Unconstrained Binary Optimization (QUBO)
  - Quadratic objective function
  - No constraints on variables
  - Binary variables
- QUBO examples
  - MaxCut
  - Number partitioning
  - Graph coloring
  - ...

MaxCut as QUBO

MaxCut

Weight matrix:



$$W = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Cost function:

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

QUBO

QUBO matrix and vectors:

$$c_i = \sum_{j=1}^{n} W_{ij}, \qquad Q_{ij} = -W_{ij}$$

Cost function:

$$\begin{aligned} C(\mathbf{x}) &= \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ &= \sum_{i,j=1}^{n} Q_{ij} x_i x_j + \sum_{i=1}^{n} c_i x_i \end{aligned}$$

QAOA

- Quantum Approximate Optimization Algorithm (QAOA) first introduced in Farhi, Goldstone and Gutmann (2014) [1]
- Finds approximate solutions to QUBO instances (e.g. MaxCut)
- Can be regarded as a special case of Variational Quantum Eigensolvers (VQE)
- Layerized variational form based on Trotterized adiabatic process (related to adiabatic quantum computing)
- Key idea: encode the cost function of the optimization problem in the cost Hamiltonian H_C

QUBO to Hamiltonian

Goal: Find the cost Hamiltonian operator H_C that encodes the cost function $C(x)$, i.e.,

$$H_C \left|x\right\rangle = C(x) \left|x\right\rangle, \qquad x = \{0,1\}.$$

Using the fact that

$$Z_i \left|x\right\rangle = (-1)^{x_i} \left|x\right\rangle = (1-2x_i)\left|x\right\rangle \quad \Longrightarrow \quad \frac{\mathbb{1}-Z_i}{2}\left|x\right\rangle = x_i \left|x\right\rangle,$$

we have

$$\begin{aligned} C(x) &= \sum_{i,j} Q_{ij} x_i x_j + \sum_i c_i x_i \\ \Longrightarrow \quad H_C &= \sum_{i,j} Q_{ij} \left(\frac{\mathbb{1}-Z_i}{2}\right)\left(\frac{\mathbb{1}-Z_j}{2}\right) + \sum_i c_i \left(\frac{\mathbb{1}-Z_i}{2}\right) \\ &= \sum_{i,j} \frac{1}{4} Q_{ij} Z_i Z_j - \sum_i \frac{1}{2}\left(c_i + \sum_j Q_{ij}\right) Z_i + \left(\sum_{i,j} \frac{Q_{ij}}{4} + \sum_i \frac{c_i}{2}\right). \end{aligned}$$

Trotterized AQC

Typical steps of Trotterized adiabatic quantum computing (AQC): [2]

- Prepare initial state as the highest energy state of some "mixer" Hamiltonian (that does not commute with H_C), $H_M = \sum_i X_i$, i.e., $|\psi_0\rangle = \otimes_i |+\rangle$.
- Set the total Hamiltonian $H(t) = f(t)H_C + g(t)H_M$ with slowly varying control functions $f(t) = t/T$ and $g(t) = 1 - t/T$.
- Through *adiabatic evolution*, the system will end up in the highest energy state of the cost Hamiltonian H_C, which then solves the QUBO problem.

Trotterized AQC

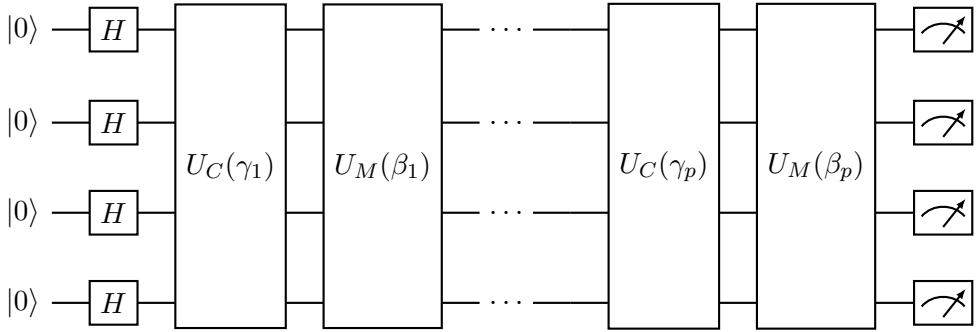
Typical steps of Trotterized adiabatic quantum computing (AQC): [2]

- Prepare initial state as the highest energy state of some "mixer" Hamiltonian (that does not commute with H_C), $H_M = \sum_i X_i$, i.e., $|\psi_0\rangle = \otimes_i |+\rangle$.
- Set the total Hamiltonian $H(t) = f(t)H_C + g(t)H_M$ with slowly varying control functions $f(t) = t/T$ and $g(t) = 1 - t/T$.
- Through *adiabatic evolution*, the system will end up in the highest energy state of the cost Hamiltonian H_C, which then solves the QUBO problem.
- In practice, to implement the adiabatic evolution, one decomposes the time-evolution operator (for a time-dependent Hamiltonian) into a sequence of small steps through the *Trotter-Suzuki formula*:

$$
\begin{aligned}
U(t) := \mathcal{T} \exp\left[-i \int_0^T H(t) dt\right] &\approx \prod_{a=0}^{k-1} \exp\left[-iH(a\tau)\tau\right] \\
&= \prod_{a=0}^{k-1} \exp\left[-if(a\tau)H_C\tau\right] \exp\left[-ig(a\tau)H_M\tau\right].
\end{aligned}
$$

QAOA Variational Form

Inspired by Trotterized AQC, QAOA was designed to be a variational algorithm with repeated cost and mixer layers.



p repetitions of alternating cost and mixer layers:

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

Matrix Exponentiation

Recall that

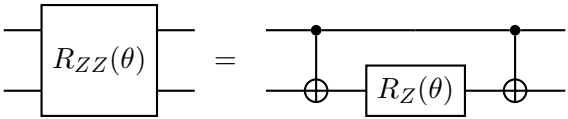
$$H_M = \sum_{i=1}^{n} X_i$$

$$H_C = \sum_{i,j=1}^{n} \frac{1}{4} Q_{ij} Z_i Z_j - \sum_{i=1}^{n} \frac{1}{2} \left(c_i + \sum_{j=1}^{n} Q_{ij} \right) Z_i + \left(\sum_{i,j=1}^{n} \frac{Q_{ij}}{4} + \sum_{i=1}^{n} \frac{c_i}{2} \right)$$

Then upon matrix exponentiation, the mixer and cost layers become

$$U_M(\beta) = e^{-i\beta H_M} = \prod_{i=1}^{n} R_{X_i}(2\beta)$$

$$U_C(\gamma) = e^{-i\gamma H_C} = \prod_{i,j=1}^{n} R_{Z_i Z_j} \left(\frac{1}{2} Q_{ij} \gamma \right) \prod_{i=1}^{n} R_{Z_i} \left(- \left(c_i + \sum_{j=1}^{n} Q_{ij} \right) \gamma \right).$$



QAOA Workflow

1. Initialize $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ with suitable real values.
2. Prepare the state $|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle$ using the QAOA circuit and measure it in the computational basis.
3. Compute the expectation value $\langle\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})|H_C|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle$.
4. Find a new set of parameters $(\boldsymbol{\beta}_{\text{new}}, \boldsymbol{\gamma}_{\text{new}})$ with a classical optimization algorithm (e.g., gradient descent) and use this new set of parameters in the QAOA circuit.
5. Repeat steps 2 - 4 until some suitable convergence criterion is met.
6. The solution is then approximated as $|\psi(\boldsymbol{\beta}_{\text{opt}}, \boldsymbol{\gamma}_{\text{opt}})\rangle$ which maximizes $\langle\psi(\boldsymbol{\beta}_{\text{opt}}, \boldsymbol{\gamma}_{\text{opt}})|H_C|\psi(\boldsymbol{\beta}_{\text{opt}}, \boldsymbol{\gamma}_{\text{opt}})\rangle$

References


E. Farhi, J. Goldstone, and S. Gutmann, *A Quantum Approximate Optimization Algorithm*, arXiv:1411.4028 [quant-ph] (2014).


Y. Sun et al., *Adiabatic Quantum Simulation Using Trotterization*, arXiv:1805.11568 [quant-ph] (2018).

Materials presented also borrow from:

- Qiskit Global Summer School 2021, *Introduction to the Quantum Approximate Optimization Algorithm and Applications*.
- Learn Quantum Computation using Qiskit, Ch. 4.1.3, *Solving combinatorial optimization problems using QAOA*.