NISQ algorithms 1: QAOA

Yipeng Huang

Rutgers University

September 21, 2020

Near-term intermediate-scale quantum (NISQ) computers

The limitations of near term quantum computers

- NISQ systems have limited number of qubits:
  No error correction.
  (In contrast, error corrected Shor's would need a million qubits.)
- NISQ systems have limited coherence time:
  Relative shallow depth of circuits.
  (In contrast, error corrected Shor's would need hundreds of millions of gates.)
- NISQ systems have limited operation accuracy

NISQ variational algorithms

Use a classical algorithm to train a "quantum neural network".

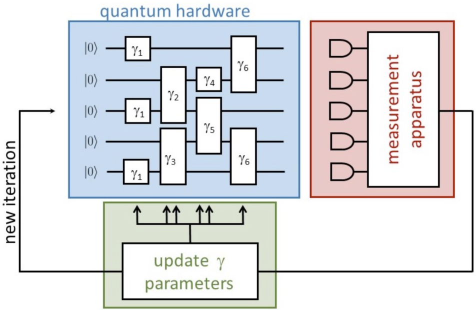

FIG. 1. Illustration of the three common steps of hybrid quantum-classical algorithms. These steps have to be repeated until convergence or when a sufficiently good quality of the solution is reached. 1) State preparation involving the quantum hardware capable of tunable gates characterized by parameters γ_n (blue), 2) measurement of the quantum state and evaluation of the objective function (red), 3) iteration of the optimization method to determine promising changes in the state preparation (green). Notice that a single parameter γ_n may characterize more than one gate, for example see γ_1 and γ_6 in the blue box. In practice, many state preparations and measurements are necessary before proceeding with a single update of the parameters.

Figure: Credit: Practical optimization for hybrid quantum-classical algorithms. Guerreschi and Smelyanskiy.

NISQ variational algorithms

Use a classical algorithm to train a "quantum neural network".

1. Quantum computer prepares a quantum state that is a function of classical parameters.
2. Quantum computer measures quantum state to provide classical observations.
3. Classical computer uses observations to calculate an objective function.
4. Classical computer uses optimization routine to propose new classical parameters to maximize objective function.
5. Repeating steps 1 through 4, the algorithm leads to better approximations to underlying problem.

NISQ variational algorithms

Benefits of quantum-classical scheme:

1. Provides meaningful results even without error correction
2. Shallow circuits (not many operations on each qubit)
3. Draws on strengths of quantum and classical:
4. Prepare and measure a quantum state
5. Optimize for a set of optimal parameters based on classical measurements

NISQ variational algorithms

Great! Can NISQ variational algorithms solve useful problems?

1. Variational quantum eigensolver (VQE):
   Simulate quantum mechanics
2. Quantum approximate optimization algorithm (QAOA):
   Solve constraint satisfaction problems (CSPs)

Constraint satisfaction problems $\in$ Combinatorial optimization problems

- Knapsack
- Traveling salesman
- Graph coloring
- MAX-SAT
- MAX-CUT

Boolean satisfiability problem

- l-SAT: NP-Complete
- A Boolean formula consisting of m clauses
  $C_1 \wedge C_2 \wedge \ldots \wedge C_m$
  For example:
  $(\neg z_0 \vee z_1 \vee z_2) \wedge (\neg z_1 \vee z_2 \vee z_3) \wedge \ldots \wedge C_m$
- C_α depends only on l coordinates of $\vec{z}$.
- Each clause C_α is either True or False.
  for each constraint $\alpha \in [m]$ and each n-bit string $\vec{z} \in \{0,1\}^n$, define
  $$C_\alpha(\vec{z}) = \begin{cases} 1 & \text{if } \vec{z} \text{ satisfies the constraint } \alpha \\ 0 & \text{if } \vec{z} \text{ does not} \end{cases}$$

Constraint satisfaction problem (CSP): MAX-SAT

- MAX-SAT: NP-Hard
- A Boolean formula consisting of m clauses
  $C_1 \wedge C_2 \wedge ... \wedge C_m$
- Each clause C_α is either True or False.
  for each constraint $\alpha \in [m]$ and each n-bit string $\vec{z} \in \{0,1\}^n$, define
  $$C_\alpha(\vec{z}) = \begin{cases} 1 & \text{if } \vec{z} \text{ satisfies the constraint } \alpha \\ 0 & \text{if } \vec{z} \text{ does not} \end{cases}$$
- Satisfy as many clauses as possible to maximize objective function $C(z)$:
  $$\max_{\vec{z}} C(\vec{z}) = \max_{\vec{z}} \sum_{\alpha=1}^{m} C_\alpha(\vec{z})$$

Approximate MAX-SAT

Approximate the maximum:

$$\max_{\vec{z}} C(\vec{z}) = \max_{\vec{z}} \sum_{\alpha=1}^{m} C_\alpha(\vec{z})$$

Constraint satisfaction problem (CSP): MAX-CUT

- Given an arbitrary undirected graph $G = (V(G), E(G))$
- goal of MAX-CUT is to assign one of two partitions to each node so as to maximize the number of cuts

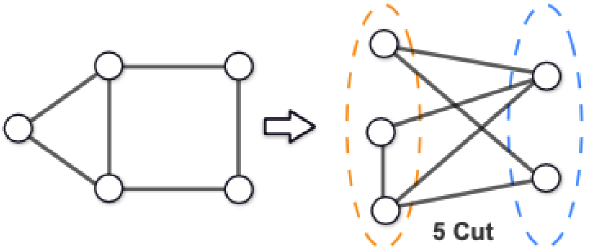

FIG. 39: An illustration of the MaxCut problem.

Figure: Credit: Quantum Algorithm Implementations for Beginners Coles.

Constraint satisfaction problem (CSP): MAX-CUT

- Given an arbitrary undirected graph $G = (V(G), E(G))$
- goal of MAX-CUT is to assign one of two partitions $\sigma_i \in \{-1, +1\}$ to each node $i \in V(G)$ so as to maximize the number of cuts
- Identical form to the MAX-SAT problem with objective function $C(\sigma)$:
  $$\max_{\sigma} C(\sigma) = \max_{\sigma} \sum_{<jk>\in E(G)} C_{<jk>}(\sigma)$$
- But the constraints are now:
  $$C_{<jk>}(\sigma) = \frac{1}{2}(1 - \sigma_j \sigma_k) = \begin{cases} 1 & \text{if } \sigma_j \text{ and } \sigma_k \text{ are different} \\ 0 & \text{if } \sigma_j \text{ and } \sigma_k \text{ are the same} \end{cases}$$

QAOA for MAX-CUT: general strategy

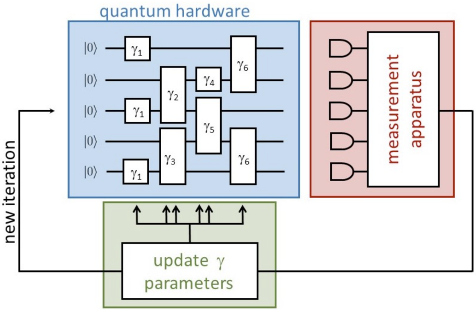

FIG. 1. Illustration of the three common steps of hybrid quantum-classical algorithms. These steps have to be repeated until convergence or when a sufficiently good quality of the solution is reached. 1) State preparation involving the quantum hardware capable of tunable gates characterized by parameters γ_n (blue), 2) measurement of the quantum state and evaluation of the objective function (red), 3) iteration of the optimization method to determine promising changes in the state preparation (green). Notice that a single parameter γ_n may characterize more than one gate, for example see γ_1 and γ_6 in the blue box. In practice, many state preparations and measurements are necessary before proceeding with a single update of the parameters.

Figure: Credit: Practical optimization for hybrid quantum-classical algorithms. Guerreschi and Smelyanskiy.

QAOA for MAX-CUT: general strategy

- Each node in n nodes of the MAX-CUT graph corresponds to one of n qubits in the quantum circuit.
- The state vector across the qubits $|\psi\rangle$ encodes a node partitioning $\sigma \in \{-1, +1\}^n$
- Put the initial state vector $|\psi_s\rangle$ in a superposition of all possible node partitionings
- Need an operator (quantum gate) that encodes an edge $< jk > \in E(G)$
- Provide classical parameters such that the classical computer can control quantum partitioning
- Perform a series of operations parameterized by classical parameters β and γ such that the final state vector $|\psi(\beta, \gamma)\rangle$ is a superposition of good partitionings
- Optimize for a good set of β and γ

QAOA for MAX-CUT: general strategy

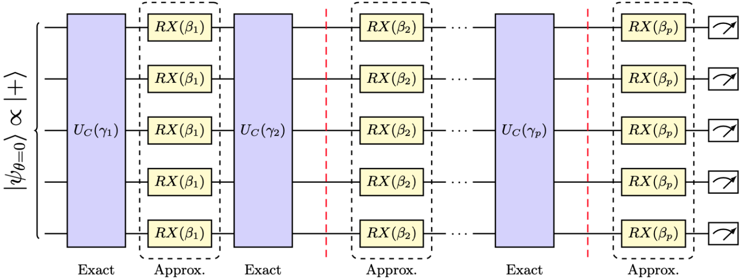

FIG. 1. A schematic representation of the QAOA circuit and our approach to simulating it. The input state is trivially initialized to $|+\rangle$. Next, at each p, the exchange of exactly (U_C, Sec. II B 1) and approximately ($RX(\beta) = e^{-i\beta X}$, Sec. II B 2) applicable gates is labeled. As noted in the main text, each (exact) application of the U_C gate leads to an increase in the number of hidden units by $|E|$ (the number of edges in the graph). In order to keep that number constant, we compress the number of hidden units (Sec. II C), indicated by red dashed lines after each U_C gate. The compression is repeated at each layer after the first, halving the number of hidden units each time.

Figure: Credit: Classical variational simulation of the Quantum Approximate Optimization Algorithm. Medvidovic and Carleo.