NISQ algorithms 1: QAOA

Yipeng Huang

Rutgers University

September 23, 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".

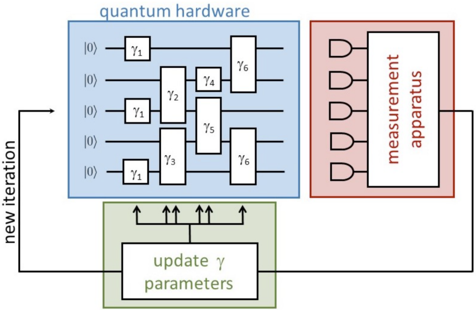

FIG. 1. Illustration of the three common steps of hybrid quantum-classical algorithms. These steps have to be repeated until convergence or when a sufficiently good quality of the solution is reached. 1) State preparation involving the quantum hardware capable of tunable gates characterized by parameters 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: [Guerreschi and Smelyanskiy, 2017]

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):
   Approximate solutions to constraint satisfaction problems (CSPs) [Farhi et al., 2014]

Constraint satisfaction problems $\in$ Combinatorial optimization problems

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

Boolean satisfiability problem

- l-SAT: NP-Complete
- n bits; m constraints/clauses
- A Boolean formula consisting of m clauses
  $C_1 \wedge C_2 \wedge \ldots \wedge C_m$
  For example:
  $(\neg z_0 \vee z_1 \vee z_2) \wedge \underline{(\neg z_1 \vee z_2 \vee z_3)} \wedge \ldots \wedge C_m$
- C_α depends only on l coordinates of $\vec{z}$.
  3 here.
- 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

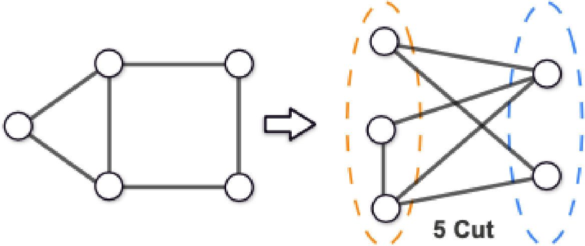

FIG. 39: An illustration of the MaxCut problem.

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 $C(\vec{\sigma})$:
$$\max_{\vec{\sigma}} C(\vec{\sigma}) = \max_{\vec{\sigma}} \sum_{<jk>\in E(G)} C_{<jk>}(\vec{\sigma})$$
- But the constraints are now:
$$C_{<jk>}(\vec{\sigma}) = \frac{1}{2}(1 - \sigma_j \sigma_k) = \begin{cases} 1 & \text{if } \sigma_j \text{ and } \sigma_k \text{ are different} \\ 0 & \text{if } \sigma_j \text{ and } \sigma_k \text{ are the same} \end{cases}$$

QAOA for MAX-CUT: general strategy

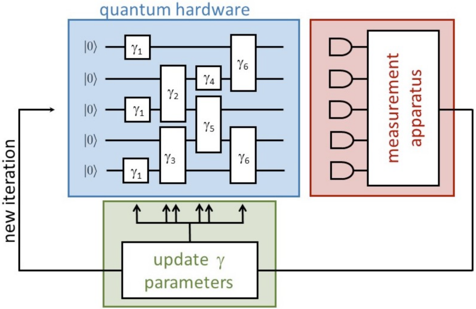

FIG. 1. Illustration of the three common steps of hybrid quantum-classical algorithms. These steps have to be repeated until convergence or when a sufficiently good quality of the solution is reached. 1) State preparation involving the quantum hardware capable of tunable gates characterized by parameters 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: [Guerreschi and Smelyanskiy, 2017]

QAOA for MAX-CUT: general strategy

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

QAOA for MAX-CUT: general strategy

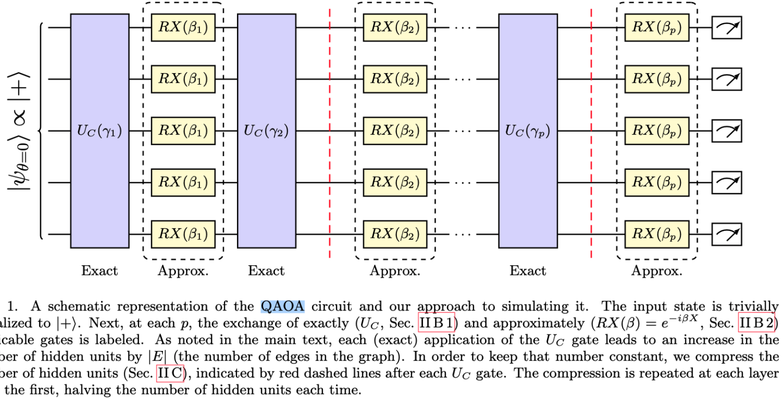

FIG. 1. A schematic representation of the QAOA circuit and our approach to simulating it. The input state is trivially initialized to $|+\rangle$. Next, at each p, the exchange of exactly (U_C, Sec. II B 1) and approximately ($RX(\beta) = e^{-i\beta X}$, Sec. II B 2) applicable gates is labeled. As noted 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.

1. Each node in n nodes of the MAX-CUT graph corresponds to one of n qubits in the quantum circuit.

Let's use an $n = 3$ example in the figure.

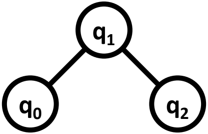

Figure: $G = (V(G), E(G)) = (\{q0, q1, q2\}, \{< q0q1 >, < q1q2 >\})$

$$|\psi\rangle = \alpha_0 |000\rangle + \alpha_1 |001\rangle \ldots + \alpha_7 |111\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \ldots \\ \alpha_7 \end{bmatrix}$$

So now we have quantum amplitudes for each of the basis states.

Probability of measuring outcome z is $|\alpha_z|^2$. $\sum_{z=0}^{n-1} |\alpha_z|^2 = 1$

2. The state vector across the qubits $|\psi\rangle$ encodes a node partitioning

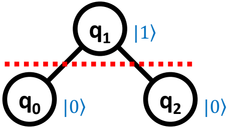

Figure: A max-cut corresponding to $|\psi\rangle = |010\rangle$.

- In our $n = 3$ running example, $|\psi\rangle = |010\rangle$, $\alpha_2 = 1$, $\alpha_0 = \alpha_1 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = \alpha_7 = 0$, is a max-cut.
- The other max-cut is $|\psi\rangle = |101\rangle$.
- A superposition $|\psi\rangle = \frac{1}{\sqrt{2}}|010\rangle + \frac{1}{\sqrt{2}}|101\rangle$ would be an equal superposition of the two max cuts.

3. Put the initial state vector $|\psi_s\rangle$ in a superposition of all possible node partitionings

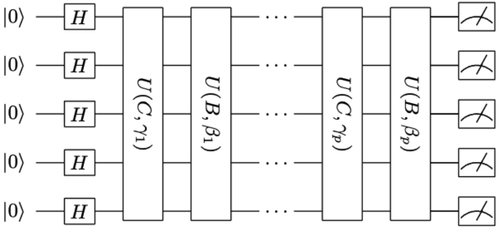

FIG. 2: Framework for a QAOA circuit. Each qubit begins with a Hadamard gate, and then $2p$ gates are performed alternating between applying Hamiltonian C and applying Hamiltonian B.

Figure: Credit: How many qubits are needed for quantum computational supremacy. Dalzell et al.

3. Put the initial state vector $|\psi_s\rangle$ in a superposition of all possible node partitionings

A superposition across all the bitstrings representing partitionings.

$$|\psi_s\rangle = |+\rangle^{\otimes n} = H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} |z\rangle$$

3. Put the initial state vector $|\psi_s\rangle$ in a superposition of all possible node partitionings

In our $n = 3$ running example:

$$|+\rangle^{\otimes n} = H^{\otimes 3}\,|0\rangle^{\otimes 3} = \begin{bmatrix} \frac{+1}{\sqrt{2}} & \frac{+1}{\sqrt{2}} \\ \frac{+1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}^{\otimes 3} \begin{bmatrix} 1 \\ 0 \end{bmatrix}^{\otimes 3} =$$

$$\begin{bmatrix} \frac{+1}{\sqrt{2}} \begin{bmatrix} \frac{+1}{\sqrt{2}} & \frac{+1}{\sqrt{2}} \\ \frac{+1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} & \frac{+1}{\sqrt{2}} \begin{bmatrix} \frac{+1}{\sqrt{2}} & \frac{+1}{\sqrt{2}} \\ \frac{+1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \\ \frac{+1}{\sqrt{2}} \begin{bmatrix} \frac{+1}{\sqrt{2}} & \frac{+1}{\sqrt{2}} \\ \frac{+1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} & \frac{-1}{\sqrt{2}} \begin{bmatrix} \frac{+1}{\sqrt{2}} & \frac{+1}{\sqrt{2}} \\ \frac{+1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \end{bmatrix} \otimes \begin{bmatrix} \frac{+1}{\sqrt{2}} & \frac{+1}{\sqrt{2}} \\ \frac{+1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}^{\otimes 3} =$$

$$\frac{1}{\sqrt{8}} \sum_{z=0}^{7} |z\rangle = [\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}]^{\dagger}$$

4. Need an operator (quantum gate) that encodes an edge $< jk > \in E(G)$

- In the Max-Cut type of CSP, constraints correspond to edges
- $C = \sum_{<jk> \in E(G)} C_{<jk>} = \sum_{<jk> \in E(G)} \frac{1}{2}(1 - \sigma_j^z \otimes \sigma_k^z)$
- σ_i^z is the Pauli-Z operator on qubit i
- $\sigma^z = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$ phase shift.
- Claim: ψ that maximizes $\langle\psi| C |\psi\rangle$ is the graph partition with the maximum cut.

4. Need an operator (quantum gate) that encodes an edge $< jk > \in E(G)$

In our $n = 3$ running example:

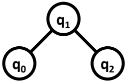

Figure: $G = (V(G), E(G)) = (\{q0, q1, q2\}, \{< q0q1 >, < q1q2 >\})$

$$C = \sum_{<jk>\in E(G)} \frac{1}{2}(1 - \sigma_j^z \otimes \sigma_k^z) = \frac{1}{2}(1 - \sigma_0^z \otimes \sigma_1^z) + \frac{1}{2}(1 - \sigma_1^z \otimes \sigma_2^z) =$$

$$\frac{1}{2}(I - \sigma^z \otimes \sigma^z \otimes I) + \frac{1}{2}(I - I \otimes \sigma^z \otimes \sigma^z) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4. Need an operator (quantum gate) that encodes an edge $< jk > \in E(G)$

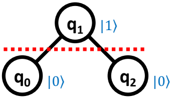

Figure: A max-cut corresponding to $|\psi\rangle = |010\rangle$.

- Claim: ψ that maximizes $\langle\psi| C |\psi\rangle$ is the graph partition with the maximum cut.
- $$\langle 010| C |010\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}^{\dagger} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 2$$

4. Need an operator (quantum gate) that encodes an edge $< jk > \in E(G)$

- $\langle 010 | C | 010 \rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}^{\dagger} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 2$
- Note the size of state vector $|\psi\rangle$ is 2^n.
- Size of constraint matrix C is $2^n \times 2^n$.
- We will enlist a quantum computer to create $|\psi\rangle$ and C.

5. Provide classical parameters such that the classical computer can control quantum partitioning

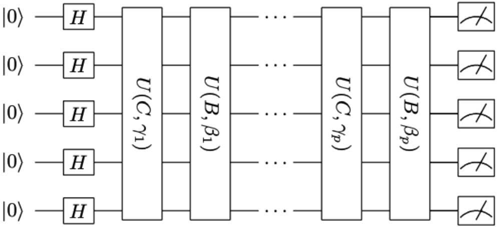

FIG. 2: Framework for a QAOA circuit. Each qubit begins with a Hadamard gate, and then $2p$ gates are performed alternating between applying Hamiltonian C and applying Hamiltonian B.

Figure: Credit: How many qubits are needed for quantum computational supremacy. Dalzell et al.

5. Provide classical parameters such that the classical computer can control quantum partitioning

- A parameter p that controls how many iterations of algorithm and how complete of a graph to see.
- Do optimization across $2p$-dimensional vector of $\vec{\gamma}$ and $\vec{\beta}$ parameters
- $(\vec{\gamma}, \vec{\beta}) = (\gamma_1, \beta_1, ...\gamma_p, \beta_p)$
- $\gamma_i \in [0, 2\pi]$
- $\beta_i \in [0, \pi]$

6. Perform a series of operations parameterized by classical parameters $\vec{\gamma}$ and $\vec{\beta}$ such that the final state vector $|\psi(\vec{\gamma}, \vec{\beta})\rangle$ is a superposition of good partitionings

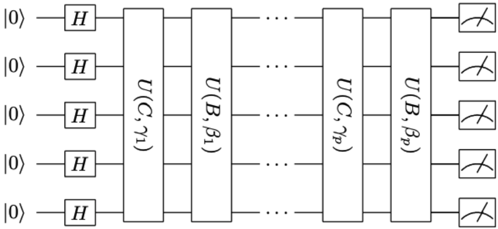

FIG. 2: Framework for a QAOA circuit. Each qubit begins with a Hadamard gate, and then $2p$ gates are performed alternating between applying Hamiltonian C and applying Hamiltonian B.

Figure: Credit: How many qubits are needed for quantum computational supremacy. Dalzell et al.

6. Perform a series of operations parameterized by classical parameters $\vec{\gamma}$ and $\vec{\beta}$ such that the final state vector $|\psi(\vec{\gamma},\vec{\beta})\rangle$ is a superposition of good partitionings

$U(C,\gamma)$, $U(B,\beta)$ are $2^n \times 2^n$ linear operators (Unitary matrices)

1. Problem Hamiltonian enforces constraints.
   A product across all the graph edges.
   $$U(C,\gamma) = e^{-i\gamma C} = \prod_{<jk>\in E(G)} e^{-i\gamma C_{<jk>}}$$
2. Admixing Hamiltonian perturbs the assignments.
   A product across all the qubits representing graph vertices.
   $$U(B,\beta) = e^{-i\beta B} = \prod_{q\in V(G)} e^{-i\beta\sigma_q^x}$$

6. Perform a series of operations parameterized by classical parameters $\vec{\gamma}$ and $\vec{\beta}$ such that the final state vector $|\psi(\vec{\gamma},\vec{\beta})\rangle$ is a superposition of good partitionings

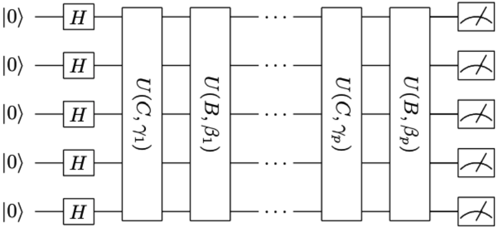

FIG. 2: Framework for a QAOA circuit. Each qubit begins with a Hadamard gate, and then $2p$ gates are performed alternating between applying Hamiltonian C and applying Hamiltonian B.

Figure: Credit: How many qubits are needed for quantum computational supremacy. Dalzell et al.

6. Perform a series of operations parameterized by classical parameters $\vec{\gamma}$ and $\vec{\beta}$ such that the final state vector $|\psi(\vec{\gamma},\vec{\beta})\rangle$ is a superposition of good partitionings

- Create ansatz states $|\psi(\vec{\gamma},\vec{\beta})\rangle$
- $|\psi(\vec{\gamma},\vec{\beta})\rangle = U(B,\beta_p)U(C,\gamma_p)...U(B,\beta_1)U(C,\gamma_1)\,|\psi_s\rangle = \prod_{i=1}^{p}\left(\prod_{q\in V(G)} e^{-i\beta_i\sigma_q^x} \prod_{<jk>\in E(G)} e^{-i\gamma_i C_{<jk>}}\right)|+\rangle^{\otimes n}$

7. Optimize for a good set of $\vec{\gamma}$ and $\vec{\beta}$

- Measure this state to compute the objective function. That is, given the current set of parameters $\vec{\gamma}$ and $\vec{\beta}$, how much of the CSP is satisified
- Use a classical optimization algorithm such as Nelder-Mead to maximize $F_p(\vec{\gamma}, \vec{\beta}) = \langle\psi(\vec{\gamma}, \vec{\beta})| \, C \, |\psi(\vec{\gamma}, \vec{\beta})\rangle$

Evaluation of QAOA for NISQ: Number of iterations?

- Let M_p be the maximum of F_p over the angles:
  $M_p = \max_{\vec{\gamma},\vec{\beta}} F_p(\vec{\gamma}, \vec{\beta})$
- QAOA needs a big parameter p to see the whole graph
- As $p \to \infty$, $\lim_{p\to\infty} M_p = \max_z C(z)$

Evaluation of QAOA for NISQ: Number of iterations?

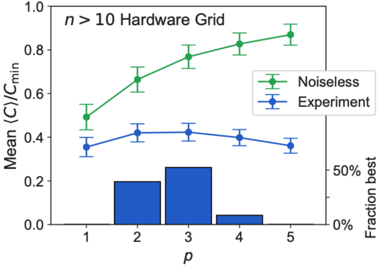

FIG. 5. QAOA performance as a function of depth, p. In ideal simulation, increasing p increases the quality of the solution. For experimental Hardware Grid results, we observe increased performance for $p > 1$ both as measured by the mean over all 10 instances studied for each value of $n \in [11, 23]$ (lines) and statistics of which p value maximizes performance on a per-instance basis (histogram). At larger p, errors overwhelm the theoretical performance increase.

Figure: Credit: [Arute et al., 2020]

Evaluation of QAOA for NISQ: Number of qubits?

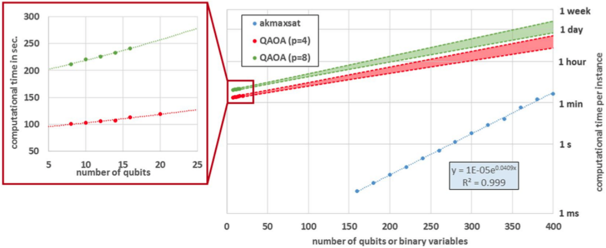

Figure 2. Main panel: Computational cost of solving a single Max-Cut instance on random 3-regular graphs. Blue markers correspond to the classical baseline (AKMAXSAT solver) while red and green marks correspond to the experimental time required by the quantum algorithm QAOA, with $p = 4$ and $p = 8$ respectively. The error bars for the single data points are smaller than the markers (see Supplementary Information). Notice that in the time needed by QAOA to partition graphs with 20 vertices, AKMAXSAT partitions graphs about 20 times larger. The blue dashed line is the result of a fitting procedure with an exponential function. The red and green areas are associated with a 95% confidence interval for the prediction of the QAOA cost based on a linear regression of $\log_{10}(T)$ as a function of the number of qubits (here T is the computational time per instance). This extrapolation should be seen as suggesting a qualitative behavior due to the uncertainty in the extrapolation from relatively small system sizes. Insert panel: Magnification of QAOA datapoints. Notice that exponential curves, and smooth curves in general, locally resemble straight lines and this makes it difficult to exclude other functional forms for the extrapolation. It is, however, believed that even quantum computers will not be able to solve NP-hard problems in polynomial time.

Figure: Credit: [Guerreschi and Matsuura, 2019]

Evaluation of QAOA for NISQ: Number of constraints?

- Findings for MAX-CUT on connected 3-regular graphs [Farhi et al., 2014]
  1. For p = 1, QAOA will always produce a cut whose size is at least 0.6924 times the size of the optimal cut.
  2. This was the best known possible approximation for a few months until classical algorithm found.
  3. For p = 2, the approximation ratio becomes 0.7559 and grows depending on the type of graph.
- Difficulty of solving problems with higher constraint ratios [Akshay et al., 2020]

Evaluation of QAOA for NISQ: Optimization method?

Optimization using gradients [Guerreschi and Smelyanskiy, 2017].

Evaluation of QAOA for NISQ: Generalizations?

- Can be generalized to solve related problems [Hadfield et al., 2019].
- Factoring [Anschuetz et al., 2019].

Read also

- Primary sources: [Farhi et al., 2014]
- Additional source: [Wang and Abdullah, 2018]

Akshay, V., Philathong, H., Morales, M. E. S., and Biamonte, J. D. (2020).
Reachability deficits in quantum approximate optimization.
Phys. Rev. Lett., 124:090504.

Anschuetz, E., Olson, J., Aspuru-Guzik, A., and Cao, Y. (2019).
Variational quantum factoring.
In Feld, S. and Linnhoff-Popien, C., editors, *Quantum Technology and Optimization Problems*, pages 74–85, Cham. Springer International Publishing.

Arute, F. C., Arya, K., Babbush, R., Bacon, D., Bardin, J., Barends, R., Boixo, S., Broughton, M. B., Buckley, B. B., Buell, D. A., Burkett, B., Bushnell, N., Chen, J., Chen, Y., Chiaro, B., Collins, R., Courtney, W., Demura, S., Dunsworth, A., Farhi, E., Fowler, A., Foxen, B. R., Gidney, C. M., Giustina, M., Graff, R., Habegger, S., Harrigan, M., Hong, S., Ioffe, L., Isakov, S., Jeffrey, E., Jiang, Z., Jones, C., Kafri, D., Kechedzhi, K., Kelly, J., Kim, S., Klimov, P., Korotkov, A., Kostritsa, F., Landhuis, D., Laptev, P., Leib, M., Lindmark, M., Lucero, E., Martin, O., Martinis, J., McClean, J. R., McEwen, M., Megrant, A., Mi, X., Mohseni, M., Mruczkiewicz, W., Mutus, J., Naaman, O., Neeley, M., Neill, C., Neukart, F., Neven, H., Niu, M. Y., O'Brien, T. E., O'Gorman, B., Petukhov, A., Putterman, H., Quintana, C., Roushan, P., Rubin, N., Sank, D., Satzinger, K., Skolik, A., Smelyanskiy, V., Strain, D., Streif, M., Sung, K. J., Szalay, M., Vainsencher, A., White, T., Yao, J., Zalcman, A., and Zhou, L. (2020).
Quantum approximate optimization of non-planar graph problems on a planar superconducting processor.
arXiv:2004.04197.

Farhi, E., Goldstone, J., and Gutmann, S. (2014).
A quantum approximate optimization algorithm.

arXiv preprint arXiv:1411.4028.

Guerreschi, G. G. and Matsuura, A. Y. (2019).
Qaoa for max-cut requires hundreds of qubits for quantum speed-up.
Scientific Reports, 9(1):6903.

Guerreschi, G. G. and Smelyanskiy, M. (2017).
Practical optimization for hybrid quantum-classical algorithms.
arXiv preprint arXiv:1701.01450.

Hadfield, S., Wang, Z., O'Gorman, B., Rieffel, E. G., Venturelli, D., and Biswas, R. (2019).
From the quantum approximate optimization algorithm to a quantum alternating operator ansatz.
Algorithms, 12(2):34.

Wang, Q. and Abdullah, T. (2018).
An introduction to quantum optimization approximation algorithm.