Project Report

On Quantum Approximate Optimization Algorithm

Contents

1 Introduction

The Quantum Approximate Optimization Algorithm(QAOA)[FGG14] is a hybrid quantum-classical algorithm that outputs approximate solutions for combinatorial optimization problems. Given a problem instance QAOA first computes a set of parameters on a classical computer, and then runs the main quantum algorithm on a quantum hardware with these parameters. QAOA and Variational Quantum Eeigensolver(VQE)[MRBAG16] algorithm are at the core of NISQ era[Pre18] algorithms.

2 Preliminaries

We denote the Pauli Matrices as $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$.

Definition 2.1. *For a matrix M, the matrix exponential is defined as the power series* $e^M = \sum_{k=0}^{\infty} \frac{1}{k!} M^k$.

Exponetiating the pauli matrices, we obtain the corresponding rotation matrices as following.

$$e^{-i\theta X} = R_X(2\theta) = \begin{bmatrix} \cos(\theta) & -i\sin(\theta) \\ -i\sin(\theta) & \cos(\theta) \end{bmatrix}$$
$$e^{-i\theta Y} = R_Y(2\theta) = \begin{bmatrix} \cos(\theta) & -i\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$
$$e^{-i\theta Z} = R_Z(2\theta) = \begin{bmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{bmatrix}$$

, where $R_X, R_Y, and\ R_Z$ are single qubit Rotation Gates around X-axis, Y-axis and Z-axis, respectively.

The Hadamard gate H sends the $|0\rangle$ state to the $|+\rangle$ $(= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle))$ state. That is

$$H : |0\rangle \to |+\rangle$$

3 The QAOA Algorithm

We now study the framework for QAOA for combinatorial optimization problem. Let $C(z)$ be the objective function defined on n bit strings for the given otimization problem, such that,

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

where $C_\alpha(z) = 1\ if$ assignment z satisfies the clause α, and 0 otherwise.

In the quantum setting, we look at the n-lenth strings z as computational basis of a 2^n dimensional Hilbert Space. We also observe that we can view C in Eq. (1) as an operator which is diagonal in the computational basis. We call it the problem operator.

Next we define

$$B = \sum_{j\in[n]} X_j \tag{2}$$

where X_j is single qubit Pauli's X operator acting upon qubit j. We call it the mixer operator.

Now, we define two unitary operators using operators from Eq. (1) and Eq. (2) respectively,

$$U(C,\gamma) = e^{-i\gamma C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha} \tag{3}$$

,where parameter $\gamma \in [0, 2\pi]$(because C has integer eigenvalues).

$$U(B, \beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta X_j} \tag{4}$$

,where parameter $\beta \in [0, \pi]$.

The initial state $|s\rangle$ is the uniform superposition of computational basis states.

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{z} |z\rangle \tag{5}$$

$|s\rangle$ is obtained using n qubits initialized in $|0\rangle$, and then applying a single qubit Hadamard Gate on each qubit.

Next, We define an angle dependent quantum state for a set of $2p$ angles $\{\gamma_1, \gamma_2, .., \gamma_p, \beta_1, \beta_2, .., \beta_p\}$,

$$|\gamma, \beta\rangle = U(B, \beta_p)U(C, \gamma_p)...U(B, \beta_1)U(C, \gamma_1) |s\rangle \tag{6}$$

Thus we have p layers of alternating Mixer and Problem Unitaries from Eq. (3) and Eq. (4) applied onto the initial state of Eq. (5).

Let $F_p(\gamma, \beta)$ be the expectation of operator C in the state $|\gamma, \beta\rangle$, then we know

$$F_p(\gamma, \beta) = \langle \gamma, \beta | C | \gamma, \beta \rangle \tag{7}$$

Let M_p be the maximum of F_p.

$$M_p = \max_{\gamma, \beta} F_p(\gamma, \beta) \tag{8}$$

That is we want to find angles $\gamma, \beta = \{\gamma_1, \gamma_2, .., \gamma_p, \beta_1, \beta_2, .., \beta_p\}$ such that this expectation is maximum.

For a fixed p, we can run the quantum computer multiple time and can achieve a maximum value in $O(m^2 + mn)$ searches. But if p does not grow with n, and each bit is involved only in a bounded number of clauses, then we can classically compute the good angles γ and β in advance.

Furthermore, the maximization at $p - 1$ can be viewd as a constrained maximization at p, therefore

$$M_p \geq M_{p-1}$$

4 A Special Case : Fixed p and Locality

We now study a special case where the optimization problem has locality(i.e. each bit can appear in a bounded number of clauses). We study the MaxCut problem on a graph with bounded degree. We show that in case of bounded degree and fixed p, the parameters γ and β can be computed efficiently on a classical computer.

Given a graph $G = (V, E), |V| = n, |E| = m$, we want to find an assignment $z = \{0, 1\}^n$, such that $|\{(u, v) \in E : z_u \neq z_v\}|$ is maximized. Thus we can define the objective function(and hence the problem operator) as below:

$$C = \frac{1}{2} \sum_{jk \in E} (I - Z_j Z_k) \tag{9}$$

From Eq. (2) we obtain,

$$C = \sum_{jk \in E} C_{\langle jk \rangle} \tag{10}$$

where $C_{\langle jk \rangle} = \frac{1}{2}(I - Z_j Z_k)$.
Using Eq. (10), we rewrite equation Eq. (8) as,

$$F_p(\gamma, \beta) = \sum_{jk \in E} \langle s | U^\dagger(C, \gamma_1)U^\dagger(B, \beta_1)...U^\dagger(B, \beta_p)C_{\langle jk \rangle}U(B, \beta_p)U(C, \gamma_p)...U(C, \gamma_1) |s\rangle \tag{11}$$

Consider the operator associated with an edge $\langle jk \rangle$ in $F_p(\gamma, \beta)$.

$$U^\dagger(C,\gamma_1)U^\dagger(B,\beta_1)...U^\dagger(B,\beta_p)C_{\langle jk\rangle}U(B,\beta_p)U(C,\gamma_p)...U(C,\gamma_1) \tag{12}$$

We consider case $p = 1$, and observe that any factors in $U^\dagger(B,\beta_1)$ which does not involve qubit j or qubit k cancels out with corresponding factor in its hermitian, as they commutes through $C_{\langle jk\rangle}$.

$$\begin{aligned} &U^\dagger(C,\gamma_1)U^\dagger(B,\beta_1)C_{\langle jk\rangle}U(B,\beta_1)U(C,\gamma_1) \\ &= U^\dagger(C,\gamma_1)\prod_{t=1}^{n} e^{i\beta_1 X_t}C_{\langle jk\rangle}\prod_{t=1}^{n} e^{-i\beta_1 X_t}U(C,\gamma_1) \\ &= U^\dagger(C,\gamma_1)e^{i\beta_1(X_j+X_k)}C_{\langle jk\rangle}e^{-i\beta_1(X_j+X_k)}U(C,\gamma_1) \end{aligned}$$

Similarly, any factor in $U^\dagger(C,\gamma_1)$ which do not involve qubit j or qubit k cancels out with the corresponding factor in hermitian. That is for $p = 1$, only qubits that effect the operator for edge $\langle jk \rangle$ are at most $p = 1$ edge away from it(i.e. all qubits on this edge or any edge adjacant to it).

Furthermore, we generalize this observation for any value of p, such that only edges(and thus qubits on those edges) which are at most p distance away from the edge $\langle jk \rangle$ effect the operator for edge(described in Eq. (12)). It means that each term corresponding to $\langle jk \rangle$ in Eq. (11) depends on a subgraph containing qubits j, k and qubits which are at most p distance away from qubit j or qubit k. Since the degree of our graph is bounded, the number of qubits in each of these subgraphs is independent of n. This allows us to compute F_p in terms of quantum subsystems whose sizes are independent of n.

For example, p = 1, and 3-regular graph, for an edge $\langle j, k \rangle$, these are the only possible subgraphs.

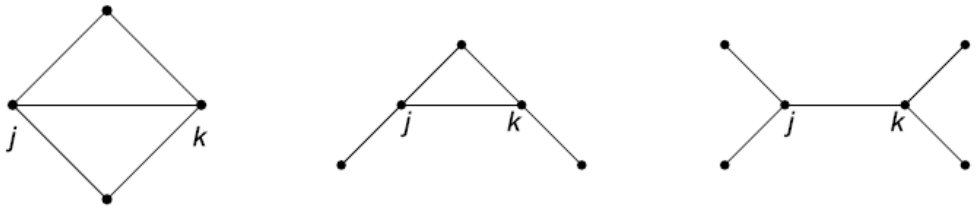

Figure 1: Subgraph Types

For any subgraph g, we define the operator C_g which is C restricted to g,

$$C_g = \sum_{\langle j'k'\rangle \in g} C_{\langle j'k'\rangle}$$

The associated unitary operator

$$U(C_g,\gamma) = e^{-i\gamma C_g}$$

$$B_g = \sum_{j' \in g} X_{j'}$$

The associated unitary operator

$$U(B_g,\beta) = e^{-i\beta B_g}$$

We define a state $|s, g\rangle$ as following (s.t. we are considering qubits only inside the subgraph g)

$$|s,g\rangle = \prod_{l \in g} |+\rangle_l$$

From Eq. (1), each edge $\langle jk \rangle$ associated with a subgraph $g(j,k)$ makes a contribution to F_p of

$$\begin{aligned}\langle s, g(j,k)|\, U^\dagger(C_{g(j,k)},\gamma_1)U^\dagger(B_{g(j,k)},\beta_1)...U^\dagger(B_{g(j,k)},\beta_p)C_{\langle jk\rangle}U(B_{g(j,k)},\beta_p)U(C_{g(j,k)},\gamma_p)\\ ...U(C_{g(j,k)},\gamma_1)\,|s, g(j,k)\rangle\end{aligned} \tag{13}$$

In case two edge gives rise to isomorphic sugraphs, then the corresponding contributions to F_p are same. Therefore, we see the sum in Eq. (11) as sum over subgraph types. We now define

$$\begin{aligned}f_g(\gamma,\beta) = \langle s, g(j,k)|\, U^\dagger(C_{g(j,k)},\gamma_1)U^\dagger(B_{g(j,k)},\beta_1)...U^\dagger(B_{g(j,k)},\beta_p)C_{\langle jk\rangle}U(B_{g(j,k)},\beta_p)U(C_{g(j,k)},\gamma_p)\\ ...U(C_{g(j,k)},\gamma_1)\,|s, g(j,k)\rangle\end{aligned} \tag{14}$$

where g(j,k) is a subgraph of type g. Let w_g be the number of occurences of the subgraph type g in the original edge sum of Eq. (11), we then obtain

$$F_p(\gamma,\beta) = \sum_g w_g f_g(\gamma,\beta) \tag{15}$$

Note that the terms $f_g(\gamma,\beta)$ do not depend on either n or m. The dependence of F_p on n and m comes through only w_g, which can be evaluated from the original graph efficiently (polynomial time classically). The number of qubits in Eq. (13) are maximum when the subgraph is a tree. The maximum number of qubits in this tree are given by

$$q_{tree} = 2\frac{(d-1)^{p+1}-1}{d-2} \tag{16}$$

where p is the fixed parameter for QAOA. For each p, there are only finitely many subgraph types.

Thus, for a fixed p(not large, because q_{tree} is growing exponetially with p) and bounded degree graph, we first compute $F_p(\gamma,\beta)$ on a classical computer using Eq. (15), and find the good values for parameters (γ,β) which maximize the value for F_p(this only gives us the approximate optimum value for F_p but it does not give us the solution i.e. a mapping of string z which achieves this value). Now using these (γ,β), we turn to the quantum hardware and produce the state $|\gamma,\beta\rangle$ given in Eq. (6). We then make the measurement in the computational basis, gets a string z, and evaluate the objective function $C_{obj}(z)$. With a given $F_p, \gamma, and\ \beta$, if we repeat this procedure involving quantum hardware for $O(m\log m)$ repitition, then we get an outcome of at least $F_p - 1$ with high probability$(= 1 - \frac{1}{m})$.

Input: Graph G = (V, E), $|V| = n, |E| = m$; the degree of each vertex is bounded
Output: A string z = $\{0,1\}^V$

Let p = 2
Classically Compute (γ,β) to maximize F_p in Eq. (7), Let $F_p(\gamma,\beta),\gamma,\beta$ be the output of classical processing.
for *i = 0 to i = $O(m\log m)$* **do**
    Prepare State $|\gamma,\beta\rangle$ on Quantum hardware
    Make Measurement in computational basis, Let z be the measured string
    Compute $C_{obj}(z)$
    if $C_{obj}(z) \geq F_p(\gamma,\beta) - 1$ **then**
        return z
return Failure

Algorithm 1: QAOA for MaxCUT with fixed p = 2

Note that with more calls to quantum hardware, i.e. more iterations of the *for* loop in Algorithm 1, we can further boost the success probability.

5 Resource Estimate : Circuit for QAOA to solve MaxCut

We now give an estimete for quantum resources required to run the qaoa on a graph(with maximum degree 3). To represent each vertex, we require one qubit. For each edge $\langle j,k\rangle$, the unitary operator is

$$U(C_{\langle j,k\rangle},\gamma) = e^{-i\gamma C_{\langle j,k\rangle}} = e^{-i\gamma\frac{1}{2}(I-Z_jZ_k)}$$

This can be achieved using 2 CNOT gates and 1 $R_Z(2\gamma)$ gate as below circuit.

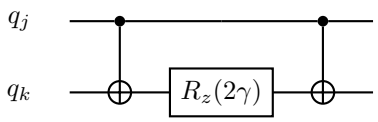

Figure 2: Problem Unitary

Therefore we add above circuit for each edge.
To implement mixer unitray from Eq. (4), we add below circuit to each qubit.

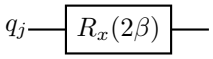

Figure 3: Mixer Unitary

For a fixed $p = 1$, parameters γ, β, below graph G=(V, E);V=$\{0, 1, 2, 3, 5\}$,E=$\{(0, 1), (1, 2), (2, 3), (3, 4), (4, 0)\}$), we generate the state $|\gamma, \beta\rangle$as below :

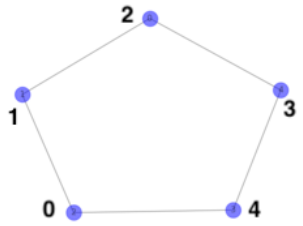

Figure 4: Graph G

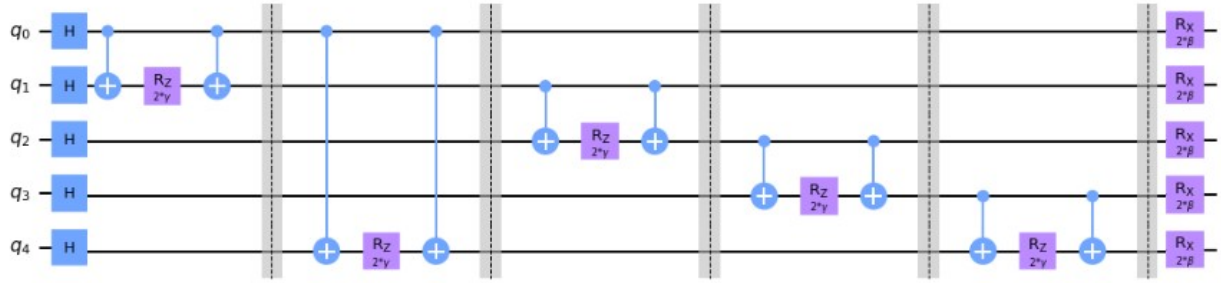

Figure 5: Circuit to run QAOA for MaxCut(generated using [Qis21])

Generally, for a given graph $G = (V, E), |V| = n, |E| = m$, s.t. the degree is bounded, we can build the QAOA circuit with:

$$\begin{aligned} Number\ of\ H\ gates &= n \\ Number\ of\ CNOT\ gates &= 2m \\ Number\ of\ R_Z\ gates &= m \\ Number\ of\ R_X\ gates &= n \end{aligned}$$

For $p = p$, we need to apply the Problem Hamiltonian and the Mixer Hamiltonian p times(with different angles). Thus the total number of CNOT, R_Z, and R_X is p times(than earlier). Thus QAOA circuit requires:

$$\begin{aligned} Number\ of\ H\ gates &= n \\ Number\ of\ CNOT\ gates &= p \cdot 2m \\ Number\ of\ R_Z\ gates &= p \cdot m \\ Number\ of\ R_X\ gates &= p \cdot n \end{aligned}$$

Since H gates are only used to initialize the state, there is no change in their number as p increases.

Can we improve upon this? A recent result [MMB$^+$21] improves upon the number of CNOT gates required for depth-p QAOA(a reduction of upto $\lfloor n/2 \rfloor$ gates).

6 Approximation Ratio for 3-Regular graph

We now study the approximation ratio of QAOA to solve MaxCut on 3-regular graphs. A 3-regular graph can be partitioned into (i) a set of isolated triangles, and (ii) a set of crossed squares. (Note we do not include the 4 vertex 3 regular graph in this analysis, although the approximation ratio is actually better than the general 3-regular case)

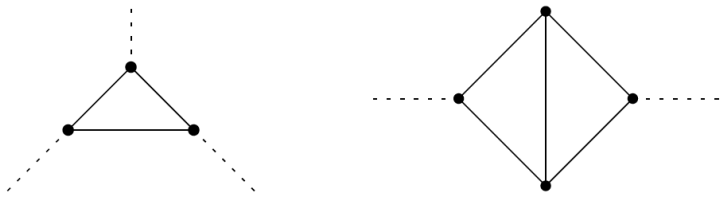

Figure 6: Isolated Triangles and Crossed Squares for 3-regular graph

Let the 3 subgaph types in Fig. 1 be $g4$, $g5$, and $g6$ (with 4,5, and 6 vertices respectively). Given a 3-regular graph, we first decompose it into T isolated triangles and S crossed squares. Then we represent each isolated triangle and crossed square in terms of subgraphs $g4, g5, and\ g6$. We now express the sum over edges in Eq. (11)

$$F_{p=1}(\gamma, \beta) = S \cdot f_{g4}(\gamma, \beta) + (4S + 3T) \cdot f_{g5}(\gamma, \beta) + (\frac{3n}{2} - 5S - 3T) \cdot f_{g6}(\gamma, \beta) \tag{17}$$

We observe that for any cut there is at least one unsatisfied edge per crossed square and at least one unsatisfied edge per isolated triangle. Thus the approximation ratios is

$$approximation\ ratio = \frac{\max_{\gamma,\beta} F_{p=1}(\gamma, \beta)}{\frac{3n}{2} - S - T} \tag{18}$$

The term $\max_{\gamma,\beta} F_{p=1}(\gamma,\beta)$ from Eq. (17) is a function of (n,S,T). We rewrite Eq. (18) as:

$$approximation\ ratio = \frac{M(n,S,T)}{\frac{3n}{2} - S - T}$$

Scaling out the n, we obtain

$$approximation\ ratio = \frac{M(1,s,t)}{\frac{3}{2} - s - t} \tag{19}$$

where $s,t \geq 0$ and $4s+3t \leq 1$. Numerically evaluating this, we get the approximation ratio to be 0.6924. Thus we know that on 3-regular graphs, with p = 1, the QAOA outputs a solution which is at least 0.6924 approximate to the optimal. For $p = 2$, the calculation of approximation ratio is more complex, but it comes to be 0.7559 for 3-regular graphs.

We note that this approximation ratio is worse than ≈ 0.878 of classical algorithm of [GW95].

7 An application to Independent Set

Consider the problem of Maximum Independent Set, where given a graph we want to find the largest subset of vertices such that no two vertices in the subset has an edge between them. As done earlier, we represent each vertex with a qubit, and a string $z = z_1 z_2 ... z_n$ is an assignment of 1 or 0 to each vertex if that vertex belongs to independent set or not(respectively). We denote our objective function(and the problem operator) as:

$$C(z) = \sum_{j=1}^{n} z_j, \tag{20}$$

Next we define a mixer quantum operator B that connects the computational basis states as:

$$\langle z|B|z'\rangle = \begin{cases} 1 : z \text{ and } z' \text{ differ in one bit} \\ 0 : \text{otherwise} \end{cases} \tag{21}$$

We then define the two unitaries $U(C,\gamma)$, and $U(B,\beta)$ as below:

$$U(C,\gamma) = e^{-i\gamma C} \tag{22}$$

,where parameter $\gamma \in [0, 2\pi]$(because C has integer eigenvalues).

$$U(B,\delta) = e^{-i\delta B} \tag{23}$$

,where parameter δ is a real number.

We start with the initial state $|s\rangle = |0\rangle$, which has the minimum value of C. Then similar to the MaxCut problem, we choose p values of δ and $p-1$ values of γ, and define a quantum state $|\gamma,\delta\rangle$ as:

$$|\gamma,\delta\rangle = U(B,\delta_p)U(C,\delta_{p-1})...U(B,\delta_1)\,|0\rangle \tag{24}$$

Let F_p be the expectation of operator C in state $|\gamma,\delta\rangle$, then we know

$$F_p(\gamma,\delta) = \langle\gamma,\delta|\,C\,|\gamma.\delta\rangle \tag{25}$$

And we want to find parameters, such that this expectation is maximum. We define :

$$M_p = \max_{\gamma,\delta} F_p(\gamma,\delta) \tag{26}$$

Also, the maximization at $p-1$ can be viewd as a constrained maximization at p with $\delta_p = 0$ and $\gamma_{p-1} = 0$, therefore

$$M_p \geq M_{p-1} \tag{27}$$

Furthermore, Eq. (24) to Eq. (27) suggests a quantum algorithm for the independent set problem. For a given p and given (γ, δ), produce the quantum state $|\gamma, \delta\rangle$ of Eq. (24). Then make the measurement in computational basis state to get a string z which is equivalent to an independent set of size of the Hamming weight of z. Repeat with same parameters (γ, δ) to get an estimate of $F_p(\gamma, \delta)$ in Eq. (25). We repeat this procedure to maximize F_p(i.e. obtain M_p of Eq. (26)).

For $p = 1$, the quantum algorithm can be thought of as evolving the initial state $|z = 0\rangle$ with the Hamiltonian B for a time δ. B is the adjacancy matrix of a big graph(not the input graph) whose vertices corresponds to the independent sets of the input graph and whose edges are from the Eq. (21)

In more general case, we can view Eq. (24) as a succession of quantum walks punctuated by applications of C dependent unitaries which aid the walk in achieving its objective. (Similar view also applies to Eq. (6))

8 Limitations: Performance Comparison with Classical Algorithms

A recent result from Bravyi[BKKT20] shows that QAOA can not outperform Goemans-Williamson[GW95] algorithm for certain instances of MaxCut for any value of p. [BKKT20] shows that for bipartite graphs with large degree, the QAOA can not exceed an approximation ratio of $5/6 \approx 0.833$, while the Goemans-Williamson algorithm has an approximation ratio of ≈ 0.878. Another recent work [Has19] provides strong evidence that local classical algorithms are likely to be as promising as the QAOA for approximate optimization.

It is believed that the QAOA highly depends on the locality and symmetry of problem instances, and hence its performance(in terms of its approximation ration) is also limited by these factors. The QAOA also has a key feature that its approximation increases as p increases, but then computing the parameters for QAOA on a classical computer requires a lot more resources as p increases (as evident from Eq. (16) for MaxCut on 3-regular graphs).

On its application to real world problems, given a large scale problem instance(for e.g. a graph with millions of nodes), it seems many years in future before we could use QAOA practically over classical algorithms(as it would require millions of qubits,and quantum gates), but it does build a foundation for a family of quantum algorithms. There are no known examples of NP complete problems where QAOA outperforms the classical approximation algorithms.

8.1 Relation to Quantum Adiabatic Algorithm

In the QAOA, we want to find an approximate solution whereas the quantum adiabatic algorithm[FGGS00] attempts to find the optimal solution if the run time for quantum computer is long enough. In [FGG14] they also show one example, where QAA fails and QAOA succeeds. In this example the objective function(minimization) is symmetric in the n bits. This again shows that QAOA's performance is really connected to the symmetry and locality of problem. It would be interesting to do a more in-depth study of the relation between QAOA and the Quantum Adiabatic Algorithm.

9 Conclusion

The execution of QAOA on a quantum computer requires a set of parameters(the angles γ, β), there exists good values of these parameters, where the QAOA outputs a good solution. We saw that in some specific cases, i.e. for fixed p and locality(each bit belongs to bounded number of clauses), it is possible to efficiently compute the good values for γ,and β on a classical computer. And under these circumstances the QAO algorithm is guaranteed to return a solution which is approximate to the optimal solution upto a factor(varies based on problem structure). The QAOA crucially depends on locality and symmetry[BKKT20]. It would be interesting if one can find problem instances(perhaps with locality) where QAOA can be proved to be

outperforming the approximation ratios of known classical algorithm.

Also, on the other hand, we would want to find instances (perhaps with some global properties) for optimization problems(other than MaxCut), and prove that QAOA can not outperform known classical approximation algorithms for any value of p.

References

[BKKT20] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang. Obstacles to variational quantum optimization from symmetry protection. *Physical Review Letters*, 125(26), dec 2020.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[FGGS00] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution, 2000.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, nov 1995.

[Has19] M. B. Hastings. Classical and quantum bounded depth approximation algorithms, 2019.

[MMB$^+$21] Ritajit Majumdar, Dhiraj Madan, Debasmita Bhoumik, Dhinakaran Vinayagamurthy, Shesha Raghunathan, and Susmita Sur-Kolay. Optimizing ansatz design in qaoa for max-cut, 2021.

[MRBAG16] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alá n Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, feb 2016.

[Pre18] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, aug 2018.

[Qis21] Qiskit. Qiskit: An open-source framework for quantum computing, 2021.

A Appendix

A.1 Jupyter Notebook for QAOA Circuit using [Qis21]