An upper bound on the Universality of the Quantum Approximate Optimization Algorithm

J CEASAR AGUMA

Using lie algebra, this brief text provides an upper bound on the universality of QAOA. That is, we prove that the upper bound for the number of alterations of QAOA required to approximate a universal gate set is within $O(n)$. 

Manuscript submitted for review to the 22nd ACM Conference on Economics & Computation (EC’21).
1 INTRODUCTION
Adding to Farhi’s introduction of QAOA[1], Seth Lloyd presented a proof that QAOA is universal by defining new \( H_z \) and \( H_x \) operators that facilitate nearest neighbor interactions between qubits on a 1D lattice[2]. This is done by "turning on and off" terms in the operators using threshold wavelengths (coefficients \( w_x \) ) on the terms of the Hamiltonian governing the system’s interactions. In a follow up paper using lie algebra, a proof was given for how these operators and others can be generated from a set of \( H_z, H_x \). This text attempts to answer a follow up question from the universal setting of QAOA.

1.1 Research Question
For a problem size \( n \) and a choice of \( H_z \) and \( H_x \) acting on \( n \) qubits we say QAOA is universal if any element in the full unitary group \( U(2^n) \) is approximated to arbitrary precision (up to a phase) from the lie algebra of \( H_z, H_x \). Therefore, we ask, for what sequence length \( p \) (number of alterations) is the set = \( H_z, H_x \) universal under QAOA dynamics.

2 PROOF
2.1 Method
The method is quite simple, we look to at the length required to generate, Pauli-X,Y,Z operators, on a 1D lattice[2]. This is done by "turning on and off" terms in the operators using threshold wavelengths (coefficients \( w_x \) ) on the terms of the Hamiltonian governing the system’s interactions.

2.2 Long-range-CNOT
This section finds the length required to approximate a CNOT on two qubits at a distance \( n \) from each other on the 1-dimensional line.

2.2.1 Step 1 : Represent the \textit{CNOT} in diagonal form. we know that the CNOT on nearest neighbors is,

\[
\text{CNOT}_{k,k+1} = |0\rangle_k \langle 0| \bigotimes I_{k+1} + |1\rangle_k \langle 1| \bigotimes X_{k+1}
\]

If we expand the length between the two qubits \( k \) and \( k+d \), where \( d = 0,1,2,...,n \), we see that

\[
\text{CNOT}_{k,k+d} = |0\rangle_k \langle 0| \bigotimes I^{n-1} + |1\rangle_k \langle 1| \bigotimes I^{n-2} \bigotimes X_{k+d}
\]

which is equivalent to,

\[
\text{CNOT}_{k,k+d} = \frac{1}{2} (I^{n} + Z_k \bigotimes I^{n-1} + I^{n-1} \bigotimes X_{k+d} - Z_k \bigotimes I^{n-2} \bigotimes X_{k+d})
\]

From intuitive inspection, the term \( Z_k \bigotimes I^{n-2} \bigotimes X_{k+d} \) is the most expensive in the \( \text{CNOT}_{k,k+d} \) expression. We will therefore use this term for finding an upper bound.

2.2.2 Step 2 : Show that \textit{required operators in the definition of CNOT can be generated}
from \( H_z, H_x \). We want to show that we can generate \( Z_k \bigotimes I^{n-2} \bigotimes X_{k+d} \) from \( H_z, H_x \).

We will start by showing how to generate \( Z_k I_{k+1} X_{k+2} \) then a show an iterative step to generate \( Z_k I_{k+1} I_{k+2} X_{k+3} \). The same step can used to generate up to \( Z_k I_{k+1} I_{k+2} ... X_{k+n} \).

We need to generate \( Z_k Z_{k+1} Z_{k+2} \cdot X_{k+1} Y_{k+2} \), commute the two to generate \( Z_k I_{k+1} X_{k+2} \).

From the properties of lie algebra, we know that if \( R \) and \( S \) are in the lie algebra then their commutators \([R,S]\) and \([S,R]\) are also in the lie algebra.
if \( n \) is odd: From \( H_z \), we can separate out the term

\[
H_{z2} = \gamma_{AB}H_{AB} + \gamma_{BA}H_{BA} = \gamma_{AB} \sum_{j=1}^{n-1} Z_{2j}Z_{2j+1} + \gamma_{BA} \sum_{j=0}^{n-3} Z_{2j+1}Z_{2j+2}
\]

[3]

Then from \( H_x \), we can generate

\[
X_{even} = \sum_{j=1}^{n-1} X_{2j}
\]
as shown in [3].

Now commuting these two,

\[
[H_{z2}, X_{even}] = \gamma_{AB} \sum_{j=1}^{n-1} Y_{2j}Z_{2j+1} + \gamma_{BA} \sum_{j=0}^{n-3} Z_{2j+1}Y_{2j+2} \pm H_{yz}^e
\]

Again commuting

\[
[H_{yz}^e, H_{z2}] = \gamma_{AB}^2 \sum_{j=1}^{n-1} X_{2j} + 2\gamma_{AB}Y_{BA} \sum_{j=1}^{n-1} Z_{2j-1}X_{2j}Z_{2j+1} + \gamma_{BA}^2 \sum_{j=0}^{n-3} X_{2j+2}
\]

From the above we can separate out

\[
H_{zxx} = 2\gamma_{AB}Y_{BA} \sum_{j=1}^{n-1} Z_{2j-1}X_{2j}Z_{2j+1}
\]

and commute again

\[
\frac{1}{2\mu} [H_{zxx}, X_{k+1}] = Z_k I_{k+1}X_{k+2}
\]

(Note that \( X_{k+1} \) and \( Y_{k+2} \) can easily be generated from \( H_{AB} \) and \( X_k \) which are generated in [3].)

With \( Z_k I_{k+1}X_{k+2} \), we use \( Y_{k+2}Z_{k+3} \) and \( Z_{k+2}Y_{k+3} \) in an iterative step to generate terms up to \( Z_k I_{k+1}I_{k+2} \ldots x_{k+n} \).

if \( n \) is even: From \( H_z \), we can separate out the term

\[
H_{z2} = \gamma_{AB}H_{AB} + \gamma_{BA}H_{BA} = \gamma_{AB} \sum_{j=1}^{\frac{n}{2}-1} Z_{2j}Z_{2j+1} + \gamma_{BA} \sum_{j=0}^{\frac{n}{2}-1} Z_{2j+1}Z_{2j+2}
\]

And from \( H_x \), we can generate

\[
X_{odd} = \sum_{j=0}^{\frac{n}{2}-1} X_{2j}
\]
as shown in [3].

Now commuting these two,

\[
[H_{z2}, X_{odd}] = \gamma_{AB} \sum_{j=1}^{\frac{n}{2}-1} Y_{2j}Z_{2j+1} + \gamma_{BA} \sum_{j=0}^{\frac{n}{2}-1} Y_{2j+1}Z_{2j+2} \pm H_{yz}^0
\]

Again commuting

\[
[H_{yz}^0, H_{z2}] = \gamma_{AB}^2 \sum_{j=1}^{\frac{n}{2}-1} X_{2j+1} + 2\gamma_{AB}Y_{BA} \sum_{j=1}^{\frac{n}{2}-1} Z_{2j-1}X_{2j}Z_{2j+1} + \gamma_{BA}^2 \sum_{j=0}^{\frac{n}{2}-1} X_{2j+1}
\]
From the above we can separate out
\[ H_{xz} = 2y_{AB}y_{BA} \sum_{j=1}^{n-1} z_{2j-1} x_{2j} z_{2j+1} \]
and commute again
\[ \frac{1}{2^n} [[H_{xz}, x_{k+1}], y_{k+2}] = z_k l_{k+1} x_{k+2} \]
(Note that \( x_{k+1} \) and \( y_{k+2} \) can easily be generated from \( H_{AB} \) and \( x_k \) which are generated in [3].)
With \( z_k l_{k+1} x_{k+2} \), we use \( y_{k+2} z_{k+3} \) and \( z_{k+2} y_{k+3} \) in an iterative step to generate terms up to \( z_k l_{k+1} l_{k+2} \ldots \ldots x_{k+n} \).

2.2.3 Step 3: find an upper bound on the sequence length required to approximate any of the operators. From counting, we find that it takes length of \( p = 3 \) to generate \( z_k l_{k+1} x_{k+2} \) and \( p \leq 10 \) to get \( x_k, y_k, z_k x_k y_{k+1}, z_k y_{K+1}, \) and \( y_k z_{K+1} \). Therefore for the iterative step we need a constant length of \( p \leq 12 = t \) which we repeat at most \( n - 2 \) times. We can conclude that we need length \( p = tn \) alternations which implies \( O(n) \) upper bound on alternations to generate \( CNOT_{k,k+x} \) using QAOA dynamics.

3 CONCLUSION
Intuitively, it is likely that the lower bound too is within \( o(n) \) but a complete proof has not availed itself yet. If one wishes to collaborate on this idea, please reach out at jaguma@uci.edu

REFERENCES
[1] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A Quantum Approximate Optimization Algorithm. arXiv:1411.4028 [quant-ph]
[2] Seth Lloyd. 2018. Quantum approximate optimization is computationally universal. arXiv preprint arXiv:1812.11075 (2018).
[3] Mauro ES Morales, Jacob Biamonte, and Zoltán Zimborás. 2019. On the Universality of the Quantum Approximate Optimization Algorithm. arXiv preprint arXiv:1909.03123 (2019).