

# Quantum approximate optimization algorithm with random and subgraph phase operators

Anthony Wilkie, Igor Gaidai , James Ostrowski, and Rebekah Herrman 

Department of Industrial and Systems Engineering *University of Tennessee at Knoxville* Knoxville, Tennessee 37996, USA



(Received 31 May 2024; accepted 29 July 2024; published 27 August 2024)

The quantum approximate optimization algorithm (QAOA) is a promising quantum algorithm that can be used to approximately solve combinatorial optimization problems. The usual QAOA ansatz consists of an alternating application of the cost and mixer Hamiltonians. In this work, we study how using Hamiltonians other than the usual cost Hamiltonian, dubbed custom phase operators, can affect the performance of the QAOA. We derive an expected value formula for the QAOA with custom phase operators at  $p = 1$  and show numerically that some of these custom phase operators can achieve higher approximation ratios than the original algorithm implementation. Out of all the graphs tested at  $p = 1$ , 0.036% of the random custom phase operators, 75.9% of the subgraph custom phase operators, 95.1% of the triangle-removed custom phase operators, and 93.9% of the maximal degree edge-removed custom phase operators have a higher approximation ratio than the original QAOA implementation. Furthermore, we numerically simulate these phase operators for  $p = 2$  and  $p = 3$  levels of QAOA and find that there exist a large number of subgraph, triangle-removed, and maximal degree edge-removed custom phase operators that have a higher approximation ratio than the QAOA at the same depth. These findings open up the question of whether better phase operators can be designed to further improve the performance of the QAOA.

DOI: [10.1103/PhysRevA.110.022441](https://doi.org/10.1103/PhysRevA.110.022441)

## I. INTRODUCTION

The quantum approximate optimization algorithm (QAOA) is an algorithm consisting of a parametrized quantum circuit that is well-suited to approximate solutions to combinatorial optimization (CO) problems [1]. The algorithm evolves a given input state under the alternating action of cost and mixing Hamiltonians for variable amounts of time that are selected to optimize the expectation of the cost Hamiltonian. The QAOA can be used to find an approximate solution to any CO problem that can be formulated as an Ising problem [2], among which the most studied is the MaxCut problem [3–7]. The goal of MaxCut is to partition the vertices of a given graph  $G = (V, E)$  into two sets to maximize the number of edges between the sets.

Initial QAOA implementation on noisy intermediate scale quantum (NISQ) devices was used for some small-scale problems [8–10] due to the challenges of extending it to larger problems related to the depth limitations of the current generation of NISQ devices. However, with advancements in error mitigation the QAOA can be implemented on NISQ devices for problems in some cases involving more than 100 qubits [11–15]. Several variations of the QAOA have been developed that attempt to reduce the required hardware resources, such as multiangle QAOA (MA-QAOA) and XQAOA [16–19], but hardware implementation remains challenging. As a result, the performance of QAOA is often studied through the classical simulations of the algorithm. In particular, closed-form equations for the expected value of the MaxCut cost Hamiltonian  $C$  have been derived for the special case when only one layer of the QAOA is applied [7,16]. Furthermore, more

general closed-form equations for all combinatorial optimization problems that have an Ising formulation have also been derived by Ozaeta *et al.* [2]. Closer inspection of these equations reveals that the expectation value of the cost Hamiltonian may be negatively affected by the presence of triangles in the graph or may be impacted by the maximum degree of the graph. As such, removing one of the terms that correspond to an edge in a triangle or an edge incident to a maximal degree vertex of the graph might improve the performance of the QAOA. Furthermore, one can consider other modifications of the QAOA circuit, detailed below, and their effect on the performance of the QAOA. One such modification is to use a subgraph of the original graph to generate the circuit. Using subgraphs for the QAOA has been explored in Refs. [20–22] in order to solve large instances of the MaxCut problem using fewer quantum resources.

Generally, the state vector prepared by the QAOA circuit can be written as

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

where  $U(B, \beta)$  and  $U(C, \gamma)$  are the unitary evolution operators under the action of the corresponding Hamiltonians  $B$  and  $C$ , and  $|s\rangle$  is a maximum (or minimum) energy eigenstate of  $B$ . The mixer Hamiltonian  $B$  is usually defined as a sum of Pauli  $X$  operators on each qubit:

$$B = \sum X_i,$$

although other mixers have been considered [23,24]. The *phase operator* Hamiltonian  $C$  is usually taken to be equal to the *cost* (or *phase*) Hamiltonian  $C'$ , i.e., the Hamiltonian whose expectation needs to be minimized by the QAOA,

$$\min \langle \vec{\gamma}, \vec{\beta} | C' | \vec{\gamma}, \vec{\beta} \rangle.$$

\*Contact author: rherrma2@utk.edu

However, the ideas described above make the phase operator different from the cost Hamiltonian ( $C \neq C'$ ), which may potentially improve the performance of the QAOA for some choices of the phase operators.

One similar development has been considered in Ref. [25], where the authors used a Hamiltonian based on a hardware connectivity graph as a phase operator and concluded that a worse, but still nontrivial, performance can be achieved with such a phase operator. However, they did not try to search for other phase operators that could potentially achieve better performance than the usual cost Hamiltonian.

Other works have tried using different phase operators. In particular, the authors of Ref. [26] suggested removing the terms of the phase operator corresponding to the edges that are not likely to be in the MaxCut and found that this indeed may improve the performance of the QAOA, while also reducing the number of gates in its circuit implementation. Another independent work [27] considered a different set of strategies to choose the terms to remove and arrived at similar conclusions for the case of not-all-equal 3-SAT problems. However, both of these works rely on starting from a classical solution, which results in a poor performance if such a solution is far from optimal. There has also been work on using random gate activation in the setting of a variational quantum eigensolver, where in each step of optimization they activate or unfreeze the trainable parameters of the two-qubit gates [28].

In this work, we focus instead on strategies that are based directly on graph structure and do not require classical solutions. Another important contribution of this work is the analytical formula for the expected value of MaxCut for arbitrary subgraph phase operators in the  $p = 1$  case, which we derive in Sec. II.

In Sec. III, we start by comparing the standard ( $C = C'$ ) formulation to a “random” phase operator formulation, where each term in the cost Hamiltonian is selected randomly. Note that one can imagine a phase operator with  $Z_i Z_j$  interactions as a graph with edges  $Z_i Z_j$ ; hence we may also talk about phase operator graphs occasionally throughout the text. We also consider the case where some fraction of the terms in the cost  $C'$  Hamiltonian are used to generate the phase operator  $C_{\text{sub}}^\alpha$ , where  $\alpha$  represents the fraction of terms selected from the original Hamiltonian  $C'$ . If  $G$  is quite dense and  $\alpha$  is small, the number of terms and, correspondingly, gates in the circuit can be significantly reduced, therefore lowering the error in the circuit. Lastly, there is evidence that triangles in problem graphs affect the QAOA MaxCut approximation ratio [3,4,29,30]. Thus, we consider a special case of the phase operator that corresponds to the original graph  $G$  with a subset of triangles removed. The resulting graph is then used to generate the phase operator.

Finally, we discuss this work and future research directions in Sec. IV.

## II. DERIVATION OF ANALYTICAL MAXCUT EXPECTED VALUE WITH CUSTOM PHASE OPERATORS

For the MaxCut problem, the cost Hamiltonian is defined as

$$C' = \sum_{uv \in E} C'_{uv} = \sum_{uv \in E} \frac{1}{2} (\mathbb{I} - Z_u Z_v). \quad (1)$$

Using the Pauli-solver algorithm detailed in Ref. [31], the expected value of  $\langle C'_{uv} \rangle$  from Eq. (1) after one layer of the QAOA with a random Hamiltonian is

$$\begin{aligned} & \langle s | e^{i\gamma C_{\text{rand}}} e^{i\beta B} C_{uv} e^{-i\beta B} e^{-i\gamma C_{\text{rand}}} | s \rangle \\ &= \frac{1}{2} - \frac{1}{2} \langle s | e^{i\gamma C_{\text{rand}}} e^{i\beta B} Z_u Z_v e^{-i\beta B} e^{-i\gamma C_{\text{rand}}} | s \rangle. \end{aligned} \quad (2)$$

Most terms of the mixer commute to leave

$$\begin{aligned} e^{i\beta B} Z_u Z_v e^{-i\beta B} &= \cos^2(2\beta) Z_u Z_v + \sin(2\beta) \cos(2\beta) \\ &\quad \times (Y_u Z_v + Z_u Y_v) + \sin^2(2\beta) Y_u Y_v. \end{aligned} \quad (3)$$

The first term of this sum commutes with  $e^{-i\gamma C_{\text{rand}}}$  and does not contribute to the expected value of the sum, i.e.,

$$\langle s | e^{i\gamma C_{\text{rand}}} Z_u Z_v e^{-i\gamma C_{\text{rand}}} | s \rangle = 0.$$

Let us look at the effect of conjugation on the second term of Eq. (3):

$$\langle s | e^{i\gamma C_{\text{rand}}} Y_u Z_v e^{-i\gamma C_{\text{rand}}} | s \rangle.$$

$C_{\text{rand}}$  has terms of the form  $\frac{1}{2} (\mathbb{I} - Z_a Z_b)$ ; however, all terms that do not have the form  $Z_u Z_c$  for some  $c$  commute and cancel, so

$$\langle s | e^{i\gamma C_{\text{rand}}} Y_u Z_v e^{-i\gamma C_{\text{rand}}} | s \rangle = \langle s | e^{-i\gamma \sum_c (Z_u Z_c)} Y_u Z_v | s \rangle.$$

If  $Z_u Z_v$  is not a term of  $C_{\text{rand}}$ , then there is no contribution to the expected value and

$$\langle s | e^{-i\gamma \sum_c (Z_u Z_c)} Y_u Z_v | s \rangle = 0.$$

Otherwise,

$$\begin{aligned} \langle s | e^{-i\gamma \sum_c (Z_u Z_c)} Y_u Z_v | s \rangle &= \langle s | e^{-i\gamma Z_u Z_v} e^{-i\gamma \sum_{c \neq v} (Z_u Z_c)} Y_u Z_v | s \rangle \\ &= \langle s | [\cos(\gamma) \mathbb{I} - i \sin(\gamma) Z_u Z_v] \prod_{i=1}^d \\ &\quad \times [\cos(\gamma) \mathbb{I} - i \sin(\gamma) Z_u Z_{c_i}] Y_u Z_v | s \rangle, \end{aligned}$$

where  $d$  is the number of terms of the form  $Z_u Z_c$  in  $C_{\text{rand}}$ , with  $c \neq v$ . The only term that contributes to the expected value is  $-i \sin(\gamma) Z_u Z_v \cos^d(\gamma) Y_u Z_v = -\sin(\gamma) \cos^d(\gamma) X_u$ , which implies

$$\begin{aligned} \langle s | e^{i\gamma C_{\text{rand}}} Y_u Z_v e^{-i\gamma C_{\text{rand}}} | s \rangle &= \langle s | -\sin(\gamma) \cos^d(\gamma) X_u | s \rangle \\ &= -\sin(\gamma) \cos^d(\gamma). \end{aligned}$$

By symmetry, the third term of Eq. (3) is also 0 if  $Z_u Z_v$  is not a term of  $C_{\text{rand}}$ ; otherwise,

$$\begin{aligned} \langle s | e^{i\gamma C_{\text{rand}}} Z_u Y_v e^{-i\gamma C_{\text{rand}}} | s \rangle &= \langle s | -\sin(\gamma) \cos^e(\gamma) X_v | s \rangle \\ &= -\sin(\gamma) \cos^e(\gamma), \end{aligned}$$

where  $e$  is the number of terms of the form  $Z_h Z_v$  in  $C_{\text{rand}}$ , where  $h \neq u$ .

Hence, we have

$$\begin{aligned} \langle s | e^{i\gamma C_{\text{rand}}} (Y_u Z_v + Z_u Y_v) e^{-i\gamma C_{\text{rand}}} | s \rangle &= -\chi_{uv} \sin(\gamma) [\cos^d(\gamma) \\ &\quad + \cos^e(\gamma)], \end{aligned}$$

where  $\chi_{uv} = 1$  if  $Z_u Z_v$  is a term in  $C_{\text{rand}}$  and 0 otherwise.

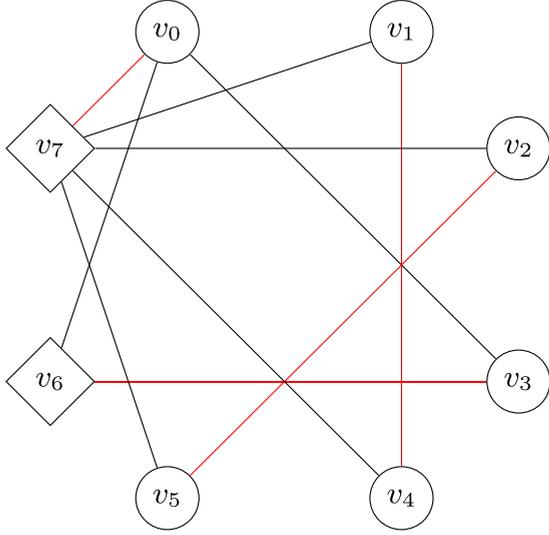


FIG. 1. An eight-vertex graph where a subgraph phase operator achieves an approximation ratio of 1. The subset of edges included in the subgraph phase operator is marked by the color red. The diamond and round vertices mark the two subsets that achieve the maximum cut. Note that the red edges form a perfect matching.

Let us look at the effect of conjugation on the last term of Eq. (3):

$$\begin{aligned} \langle s | e^{i\gamma C_{\text{rand}}} Y_u Y_v e^{-i\gamma C_{\text{rand}}} | s \rangle &= \langle s | e^{i\gamma \sum_{b \neq v} Z_u Z_b} e^{i\gamma \sum_{a \neq u} Z_a Z_v} Y_u Y_v | s \rangle \\ &= \langle s | \prod_{i=1}^d [\cos(\gamma) \mathbb{I} - i \sin(\gamma) Z_u Z_{b_i}] \prod_{j=1}^e \\ &\quad \times [\cos(\gamma) \mathbb{I} - i \sin(\gamma) Z_{a_j} Z_v] Y_u Y_v | s \rangle. \end{aligned}$$

By approaches similar to those in Ref. [31], the expected value contribution from this term is  $\frac{1}{2} \cos^{d+e-2f}(\gamma) [1 - \cos^f(2\gamma)]$ , where  $f$  is the number of pairs of terms that satisfy the triangle condition for edge  $(u, v)$ . We say two terms,  $Z_u Z_w$  and  $Z_w Z_v$ , for  $w \neq u, v$  in the phase operator  $C_{\text{rand}}$  satisfy the *triangle condition* if  $(u, v)$  is an edge in the graph of interest.

Combining the above terms yields

$$\begin{aligned} \langle C'_{uv} \rangle &= \frac{1}{2} + \frac{\chi_{uv}}{4} \sin(4\beta) \sin(\gamma) [\cos^d(\gamma) + \cos^e(\gamma)] \\ &\quad - \frac{1}{4} \sin^2(\beta) \cos^{d+e-2f}(\gamma) [1 - \cos^f(2\gamma)]. \end{aligned} \quad (4)$$

**A. Example: An eight-vertex graph where a subgraph phase operator achieves an approximation ratio of 1**

Figure 1 shows a graph  $G$  where there exists a subgraph  $G'$  consisting of only the red edges such that using the corresponding phase operator yields a higher approximation ratio compared to the usual QAOA phase operator. The approximation ratio for this graph using the original QAOA phase operator is 0.934, as found with the code from Ref. [32].

The phase operator in this case is given by

$$C = Z_0 Z_7 + Z_1 Z_4 + Z_2 Z_5 + Z_3 Z_6.$$

From Eq. (4), one can see that the cost expectation of edges in  $G$  that are not in  $G'$  is 0.5. The edges that are in  $G'$  do

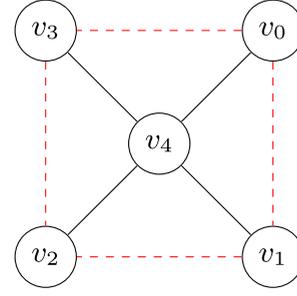


FIG. 2. The star graph, which consists of vertices and the solid edges. The dashed red edges are the terms included in the phase operator. This choice of phase operator yields an approximation ratio of 0.5.

not form triangles; therefore, it is easy to see that their cost expectation becomes equal to 1 for  $\gamma = \pi/2$  and  $\beta = \pi/8$ . There are six edges that are in  $G$  but not in  $G'$  and four edges in  $G'$ ; therefore, the total cost expectation, 7, in this case coincides with the maximum cut for this graph.

**B. Example: A family of graphs and circuits with an approximation ratio of 0.5**

The star graph on  $n$  vertices is a connected  $n$ -vertex graph that has exactly one vertex of degree  $n - 1$  and  $n - 1$  vertices of degree 1. An example of a star graph is depicted in Fig. 2.

Consider a star graph with three or more leaves. Label the center vertex of the star  $n - 1$  and the other vertices 0 to  $n - 2$  in clockwise order, starting at some arbitrary leaf. Now consider the phase operator with the terms of the form  $Z_i Z_{i+1}$  modulo  $n - 2$ , which has  $n - 1$  terms, the same as the cost Hamiltonian. With this choice of the phase operator, the last term of Eq. (4) for all edges of the graph is 0, since none of them satisfy the triangle condition. Furthermore, the second term in Eq. (4) is also 0 since none of edges are chosen as a term in the phase operator. Therefore, all edges in the graph have the cost expectation of 0.5, and the overall cost expectation for a star graph with this phase operator is  $\frac{|E|}{2}$ . In general, if the edges of the graph do not correspond to the terms of the phase operator and do not satisfy the triangle condition, the approximation ratio will always be 0.5. Note if the star has two leaves, this phase operator construction will contain a multiedge and thus require additional parameter optimization considerations.

**III. RESULTS**

In this work, we considered a QAOA with random phase operators, subgraph phase operators, a special case of subgraph phase operators in which a subset of triangles are removed (abbreviated TR phase operator for triangle-removed), and a subclass of random subgraphs where edges that are incident to maximal degree vertices are removed (abbreviated MDER phase operator for maximum-degree edge removal) for solving the MaxCut problem on all 11 117 nonisomorphic eight-vertex graphs. The simulations at  $p = 1$  were performed by maximizing the expected value formula in Eq. (4) over the parameters  $\gamma$  and  $\beta$ . The  $p = 2$  and  $p = 3$  code and

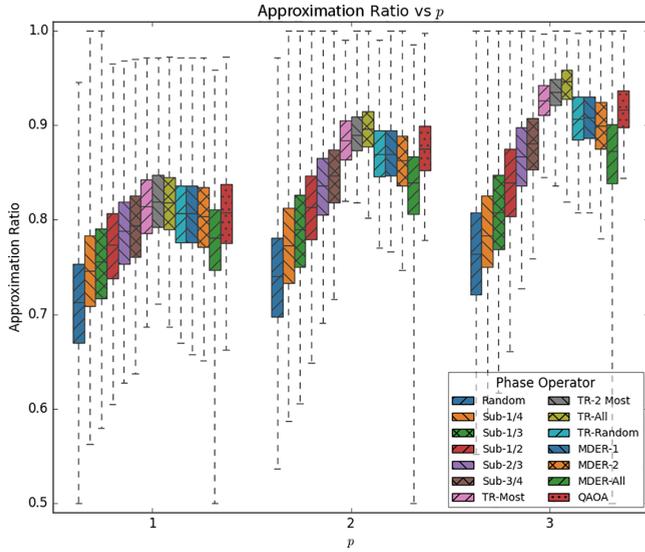


FIG. 3. Comparison of the approximation ratio for each phase operator and  $p$  tested. For each  $p$ , the random phase operator performs the worst, while one of the TR phase operators performs the best, with the regular QAOA in the middle.

classical optimization subroutine can be found in Ref. [33]. The optimizer used was the L-BFGS algorithm with 100 random parameter initialization seeds.

**A. Random phase operators**

First, we consider random phase operators and analyze the effect of phase operator structure on the approximation ratio. In order to create the phase operator graphs, we randomly selected  $m$  of the  $\binom{8}{2}$  possible edges of the complete graph on eight vertices, where  $m$  is the number of edges in the original graph. Mathematically, this is given by

$$C_{\text{rand}} = \sum_{ab} \frac{1}{2} (\mathbb{I} - Z_a Z_b),$$

where the indices  $(a, b)$  are random. For each graph, up to ten nonisomorphic random phase operators were considered. Note that in some cases, such as the complete graph  $K_8$ , we cannot select ten nonisomorphic phase operators, since we keep the same number of edges in the phase operator graph. However, when possible, we generated ten nonisomorphic random phase operators and evaluated the approximation ratio for each.

From Fig. 4, we can see that in almost no cases did a random phase operator outperform the cost phase operator. Specifically 0.036% of the graphs tested at  $p = 1$  had a random phase operator that outperformed the cost phase operator. The random graphs are not guaranteed to have any structure with respect to the original graph, so it is not surprising that the random phase operators typically perform worse. Furthermore, there are far more than ten nonisomorphic phase operators that can be generated in most cases, so it is of course possible that the remaining graphs have better phase operators. Figure 3 shows the performance of the random phase operators as compared to the cost phase operator for each  $p$ . On average, the random phase operators not only performed

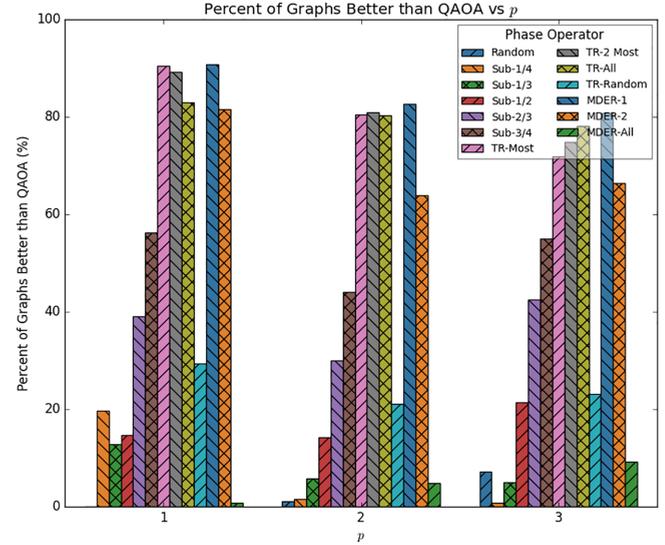


FIG. 4. The percentage of eight-vertex graphs that had at least one instance of the phase operator perform better than the regular QAOA in terms of the approximation ratio.

worse than the cost phase operator, but also worse than all the other phase operators considered in this work, with the lowest average approximation ratio. This trend continued for all  $p$  tested.

**B. Subgraph phase operators**

The subgraph phase operators differ from the purely random ones in that we chose a random subgraph of the original graph  $G$  and added to the phase operator  $Z_i Z_j$  only the terms corresponding to edges of this subgraph. In other words, only a subset of terms from the cost Hamiltonian can be considered for the phase operator.

In this experiment, we tested five cases of the subgraph phase operator, where the subgraphs chosen contain  $\lceil \alpha |E(G)| \rceil$  edges of  $G$  for one  $\alpha \in \{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}\}$ . We labeled each case as Sub- $\alpha$ , for the corresponding  $\alpha$ . Similar to before, we chose up to ten nonisomorphic subgraphs for each  $\alpha$  and used 100 parameter initialization seeds.

Figure 3 shows the performance of the subgraph phase operators as compared to the cost Hamiltonian for MaxCut for each  $p$ . We can see that the subgraph phase operators performed better than the random phase operators, but not as well as the TR phase or the MDER phase operators. What is more interesting to note is that all the subgraph phase operators performed worse than the normal QAOA for every  $p$  tested. We can see from Fig. 4 that for Sub- $\frac{1}{4}$  and Sub- $\frac{1}{3}$  the percentage of graphs that had at least one phase operator that performed better than the normal QAOA decreases with  $p$ , while for the other cases there is a decrease at  $p = 2$  and an increase at  $p = 3$ . Out of all the subgraph phase operators tested, 75.9% of the considered graphs had at least one subgraph phase operator with a higher approximation ratio at  $p = 1$ . These results suggest that randomly choosing a subgraph to create the phase operator from does not always yield a better approximation ratio. Since only ten

nonisomorphic subgraph phase operators were tested, it is possible that there exist phase operators that have even better performance. Specifically, as we will see with the TR phase operators in Sec. III C, subgraphs that have specific edges removed to reduce the number of triangles can improve the approximation ratio.

One interesting note is that there were ten phase operator subgraphs that achieved an approximation ratio of 1, meaning that the optimal MaxCut value was found. The structures of these subgraphs all have the same form, where the subgraph is a perfect matching as in Fig. 1. A perfect matching  $M \subset E$  is a subset of graph edges such that every vertex in  $G$  is incident to exactly one edge in  $M$ . While interesting to note, this condition is not necessary or sufficient to define subgraph phase operators that achieve an approximation ratio of 1. However, if the MaxCut value is  $\frac{|V|}{4} + \frac{|E|}{2}$ , then a perfect matching phase operator, if it exists, does yield an approximation ratio of 1. In general, this information is not known *a priori*.

### C. TR phase operators

The TR phase operators are a special case of the subgraph phase operators in which one or more triangles are removed from the original graph. All edges that are not in triangles remain in the phase operator. The importance of this case is to reduce the number of triangles so that the last term of Eq. (4) is 0 for a larger number of edges in the graph, with the hope that this increases the overall expected value. This experiment examines four types of TR phase operators. The first type, named TR-Most, removes a single edge that is contained in most triangles in the graph. The second type, named TR-2 Most, implements the same “most” rule, but applied consecutively 2 times. The third type, named TR-All, iteratively removes edges until there are no more triangles left in the graph. The final type, named TR-Random, just chooses a single random edge included in a random triangle and removes it.

Figure 3 shows the performance of the TR phase operators as compared to the cost phase operators for each  $p$ . These results indicate that removing triangles from the cost phase operator can improve the approximation ratio, while also using fewer gates in the circuit. The TR phase operators (excluding TR-Random) performed the best over all compared to the other phase operators, which becomes more apparent as  $p$  increases. Out of all the TR phase operators tested at  $p = 1$ , 95.1% of the considered graphs had at least one TR phase operator that yielded a higher approximation ratio. From Fig. 4, we see the same trend that TR-All begins to perform better on more graphs than TR-Most and TR-2 Most as  $p$  increases. However, comparing the performance of TR-Most, TR-2 Most, and TR-All, we see that removing all triangles did not yield the best performance initially, but at  $p > 2$  TR-All outperforms the other two. This may be due to the fact that removing all triangles may result in new structure, such as an increase in the diameter of the graph, which may be difficult for the QAOA to see. Additionally, the expectation of the removed edges tends to decrease, which offsets the gain from the remaining edges. We can also see that for each  $p$ , removing at least two edges performs better on average than removing only a single edge. In every case, the TR-Random

phase operator performed the worst among the TR phase operators, but was still comparable to the regular QAOA, which implies that edges should be strategically removed.

### D. MDER phase operators

The MDER phase operators are subcases of the subgraph phase operator in which at least one edge that is incident to maximum-degree vertices is removed from the original graph. We examine this type of phase operator because the exponents of the trigonometric functions in the expected value equation depend on the degree of the vertex incident to each edge. Removing an edge with a high degree drives down the degree of a subset of the trigonometric functions in Eq. (4). This may slightly increase the expected value of all edges that share a common vertex with the removed edge at the cost of setting the expected value of the removed edge to 0.5 plus possibly the triangle term in the expected value. Thus, edges incident to high-degree vertices are removed. This experiment considers three types of MDER phase operators: the MDER-1 phase operator where we find the maximal degree vertex and remove a random edge incident to it, the MDER-2 phase operator, which consecutively applies the previous strategy 2 times, and the MDER-all phase operator where we find the maximal degree vertex and remove all the edges incident to it.

As shown in Fig. 4, over 60% of the tested graphs had at least one MDER-1 and MDER-2 instance that gave a higher approximation ratio than the QAOA, while almost no MDER-All phase operators did. Furthermore, MDER-1 and MDER-2 result in approximation ratios comparable to those of the QAOA, while MDER-All performed worse at  $p = 1$ , as seen in Fig. 3. However, as  $p$  increases the MDER phase operators begin to perform increasingly worse than the regular QAOA. These results indicate that removing edges incident to maximum-degree vertices can yield comparable performance at lower depth to QAOA while using fewer gates. However, removing all the edges incident to the maximum-degree vertex can lead to worse performance.

## IV. DISCUSSION

In this work, we examined the impact of the QAOA phase operator design on the approximation ratio on all nonisomorphic, connected eight-vertex graphs and up to three layers of the QAOA. In general, removing a subset of triangles from the phase operator tends to yield a higher average approximation ratio, and using a random phase operator tends to result in an approximation ratio lower than the approximation ratio yielded by the cost phase operator.

The main implication of this work is that, on average, fewer gates can be used in the QAOA circuit and a comparable approximation ratio can be attained. This can provide a modest noise reduction when implementing the algorithm on NISQ devices. Future work may include multiple directions. First, one can examine how the subgraph phase operators affect the optimization landscape, as there could be a risk that modifying the phase operator results in more barren plateaus. Second, it is important to consider the scalability of the custom phase operators with respect to the number of layers of the QAOA. Another direction includes characterizing of when removing

gates will improve the approximation ratio. We will also consider using the MA-QAOA with the subgraph phase operators to see if the approximation ratio of a given problem can be increased even further. Since the subgraph phase operators have fewer gates, there would be fewer angles to optimize. Thus, the parameter optimization of these multiangle circuits should be easier, which overcomes one of the challenges of MA-QAOA implementation [34]. It would also be interesting to choose other types of phase operators. In particular, at iteration  $p$  of the QAOA, cycles of length  $2p + 1$  contribute to the expected value of edges in the cycle. One could study how removing edges from appropriate length cycles at each iteration of the QAOA impacts the approximation ratio. We believe this type of dropout method would result in even higher approximation ratios than just removing edges in triangles, as in this work.

One possible application of this work is in the area of privacy. In Ref. [35], the authors explore creating multiple circuits with various gates removed from the QAOA cost Hamiltonian in order to protect the privacy of the problem at hand at the cost of having to run multiple circuits on different third-party hardware and recombining the results. Our work shows that we can remove certain gates from the cost Hamiltonian, thus increasing the privacy of the original problem, and have comparable results without relying on multiple circuits.

### ACKNOWLEDGMENTS

J.O. acknowledges DARPA ONISQ program under Award No. W911NF-20-2-0051. R.H., J.O., and A.W. acknowledge NSF Grant No. CCF 2210063.

- 
- [1] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](#).
- [2] A. Ozaeta, W. van Dam, and P. L. McMahon, Expectation values from the single-layer quantum approximate optimization algorithm on Ising problems, *Quantum Sci. Technol.* **7**, 045036 (2022).
- [3] R. Herrman, L. Treffert, J. Ostrowski, P. C. Lotshaw, T. S. Humble, and G. Siopsis, Impact of graph structures for QAOA on MaxCut, *Quantum Inf. Process.* **20**, 289 (2021).
- [4] E. Farhi, D. Gamarnik, and S. Gutmann, The quantum approximate optimization algorithm needs to see the whole graph: Worst case examples, [arXiv:2005.08747](#).
- [5] J. Wurtz and D. Lykov, The fixed angle conjecture for QAOA on regular MaxCut graphs, [arXiv:2107.00677](#).
- [6] G. G. Guerreschi and A. Y. Matsuura, QAOA for Max-Cut requires hundreds of qubits for quantum speed-up, *Sci. Rep.* **9**, 6903 (2019).
- [7] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, Quantum approximate optimization algorithm for MaxCut: A fermionic view, *Phys. Rev. A* **97**, 022304 (2018).
- [8] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices, *Phys. Rev. X* **10**, 021067 (2020).
- [9] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo *et al.*, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, *Nat. Phys.* **17**, 332 (2021).
- [10] N. Earnest, C. Tornow, and D. J. Egger, Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware, *Phys. Rev. Res.* **3**, 043088 (2021).
- [11] E. Pelofske, A. Bärttschi, and S. Eidenbenz, Quantum annealing vs. QAOA: 127 qubit higher-order ising problems on NISQ computers, in *International Conference on High Performance Computing* (Springer, Berlin, 2023), pp. 240–258.
- [12] S. H. Sack and D. J. Egger, Large-scale quantum approximate optimization on nonplanar graphs with machine learning noise mitigation, *Phys. Rev. Res.* **6**, 013223 (2024).
- [13] S. V. Barron, D. J. Egger, E. Pelofske, A. Bärttschi, S. Eidenbenz, M. Lehmkuehler, and S. Woerner, Provable bounds for noise-free expectation values computed from noisy samples, [arXiv:2312.00733](#).
- [14] A. Miessen, D. J. Egger, I. Tavernelli, and G. Mazzola, Benchmarking digital quantum simulations and optimization above hundreds of qubits using quantum critical dynamics, [arXiv:2404.08053](#).
- [15] J. A. Montanez-Barrera and K. Michielsen, Towards a universal QAOA protocol: Evidence of quantum advantage in solving combinatorial optimization problems, [arXiv:2405.09169](#).
- [16] R. Herrman, P. C. Lotshaw, J. Ostrowski, T. S. Humble, and G. Siopsis, Multi-angle quantum approximate optimization algorithm, *Sci. Rep.* **12**, 1 (2022).
- [17] J. Wurtz and P. J. Love, Classically optimal variational quantum algorithms, *IEEE Trans. Quantum Eng.* **2**, 3104107 (2021).
- [18] V. Vijendran, A. Das, D. E. Koh, S. M. Assad, and P. K. Lam, An expressive ansatz for low-depth quantum optimisation, *Quantum Sci. Technol.* **9**, 025010 (2024).
- [19] K. Shi, R. Herrman, R. Shaydulin, S. Chakrabarti, M. Pistoia, and J. Larson, Multiangle QAOA does not always need all its angles, in *Proceedings of the 2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)* (IEEE, New York, 2022), pp. 414–419.
- [20] Z. Zhou, Y. Du, X. Tian, and D. Tao, Qaoa-in-qaoa: solving large-scale maxcut problems on small quantum machines, *Phys. Rev. Appl.* **19**, 024027 (2023).
- [21] J. Li, M. Alam, and S. Ghosh, Large-scale quantum approximate optimization via divide-and-conquer, in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (IEEE, New York, 2022).
- [22] M. Ponce, R. Herrman, P. C. Lotshaw, S. Powers, G. Siopsis, T. Humble, and J. Ostrowski, Graph decomposition techniques for solving combinatorial optimization problems with variational quantum algorithms, [arXiv:2306.00494](#).

- [23] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, *XY mixers: Analytical and numerical results for the quantum alternating operator ansatz*, *Phys. Rev. A* **101**, 012320 (2020).
- [24] A. Bärttschi and S. Eidenbenz, Grover mixers for QAOA: Shifting complexity from mixer design to state preparation, in *Proceedings of the 2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, New York, 2020), pp. 72–82.
- [25] E. Farhi, J. Goldstone, S. Gutmann, and H. Neven, Quantum algorithms for fixed qubit architectures, [arXiv:1703.06199](https://arxiv.org/abs/1703.06199).
- [26] X. Liu, R. Shaydulin, and I. Safro, Quantum approximate optimization algorithm with sparsified phase operator, in *Proceeding of the 2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, New York, 2022), pp. 133–141.
- [27] Z. Wang, P.-L. Zheng, B. Wu, Y. Zhang *et al.*, Quantum dropout: On and over the hardness of quantum approximate optimization algorithm, *Phys. Rev. Res.* **5**, 023171 (2023).
- [28] S. Liu, S.-X. Zhang, S.-K. Jian, and H. Yao, Training variational quantum algorithms with random gate activation, *Phys. Rev. Res.* **5**, L032040 (2023).
- [29] E. Farhi, D. Gamarnik, and S. Gutmann, The quantum approximate optimization algorithm needs to see the whole graph: A typical case, [arXiv:2004.09002](https://arxiv.org/abs/2004.09002).
- [30] J. Wurtz and P. Love, Maxcut quantum approximate optimization algorithm performance guarantees for  $p > 1$ , *Phys. Rev. A* **103**, 042612 (2021).
- [31] S. Hadfield, *Quantum Algorithms for Scientific Computing and Approximate Optimization* (Columbia University, 2018).
- [32] <https://github.com/GaidaiIgor/MA-QAOA>.
- [33] A. Wilkie, Random circuit QAOA, 2024, available at <https://github.com/Vilcius/Random-Driver-QAOA>.
- [34] I. Gaidai and R. Herrman, Performance analysis of multi-angle QAOA for  $p > 1$ , *Sci. Rep.* **14**, 18911 (2024).
- [35] S. Upadhyay and S. Ghosh, Obfuscating quantum hybrid-classical algorithms for security and privacy, in *2024 25th International Symposium on Quality Electronic Design (ISQED)* (IEEE, Piscataway, NJ, 2024), pp. 1–8.