Multiangle QAOA Does Not Always Need All Its Angles

Kaiyan Shi*†, Rebekah Herrman§, Ruslan Shaydulin‡, Shouvanik Chakrabarti‡, Marco Pistoia‡, and Jeffrey Larson*
*Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL, USA
‡Global Technology Applied Research, JPMorgan Chase, New York, NY, USA
§Industrial and Systems Engineering, University of Tennessee at Knoxville, Knoxville, TN, USA
†Dept. of Computer Science, University of Maryland, College Park, MD, USA
Email: *kshi12@umd.edu, *jmlarson@anl.gov

2022 IEEE/ACM 7th Symposium on Edge Computing (SEC) | 978-1-6654-8611-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SEC54971.2022.00062

***Abstract*—Introducing additional tunable parameters to quantum circuits is a powerful way of improving performance without increasing hardware requirements. A recently introduced multiangle extension of the quantum approximate optimization algorithm (ma-QAOA) significantly improves the solution quality compared with QAOA by allowing the parameters for each term in the Hamiltonian to vary independently. Prior results suggest, however, considerable redundancy in parameters, the removal of which would reduce the cost of parameter optimization. In this work we show numerically the connection between the problem symmetries and the parameter redundancy by demonstrating that symmetries can be used to reduce the number of parameters used by ma-QAOA without decreasing the solution quality. We study Max-Cut on all 7,565 connected, non-isomorphic 8-node graphs with a nontrivial symmetry group and show numerically that in 67.4% of these graphs, symmetry can be used to reduce the number of parameters with no decrease in the objective, with the average ratio of parameters reduced by 28.1%. Moreover, we show that in 35.9% of the graphs this reduction can be achieved by simply using the largest symmetry. For the graphs where reducing the number of parameters leads to a decrease in the objective, the largest symmetry can be used to reduce the parameter count by 37.1% at the cost of only a 6.1% decrease in the objective. We demonstrate the central role of symmetries by showing that a random parameter reduction strategy leads to much worse performance.**

***Index Terms*—QAOA, Max-Cut, Graph automorphism**

I. Introduction

Quantum hardware has improved rapidly in recent years [1]–[4], opening up the possibility of demonstrating quantum advantage on a relevant practical problem. Combinatorial optimization problems are commonly considered targets for near-term quantum devices [5], [6], with the quantum approximate optimization algorithm (QAOA) [7], [8] as a promising candidate algorithm because of its low hardware resource requirements [9]–[12].

QAOA solves optimization problems using a parameterized circuit composed of layers of alternating operators, with two operators being evolutions with a Hamiltonian encoding the objective function and a problem-instance-independent mixer Hamiltonian. The evolution times are free parameters (often called angles), which are optimized with the goal of maximizing the expected quality of the measurement outcomes. The success of variational quantum algorithms with a large number of trainable parameters such as quantum neural networks and the variational quantum eigensolver [13] motivated the introduction of additional parameters in QAOA. Intuitively, adding additional parameters to the algorithm based on the structure of the problem can only increase the circuit expressiveness and thereby can only improve the algorithm's performance.

Multiangle QAOA (ma-QAOA) is a modification of QAOA that incorporates additional parameters [14] by allowing the parameter associated with each term in the problem and mixer Hamiltonian to vary independently. ma-QAOA has been shown to solve Max-Cut on star graphs exactly using only one layer, whereas QAOA achieves an approximation ratio of only 0.75. The improvement in the quality of the solution achieved by the introduction of the parameters is modest, however, suggesting that the large number of parameters does not translate to a highly expressive circuit. Moreover, preliminary ma-QAOA research has shown that parameters tend to cluster around multiples of 0.25π [15]. Together, these observations suggest that the number of parameters in ma-QAOA can be reduced without affecting the solution quality.

In this work we demonstrate the connection between the redundancy in ma-QAOA parameters and the problem symmetries. Specifically, we reduce the number of

This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research for Quantum Computing program. R.H. acknowledges the NSF award CCF-2210063 and the DARPA ONISQ program under award W911NF-20-2-0051. The opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of JPMorgan Chase.

978-1-6654-8611-8/22/$31.00 ©2022 IEEE
DOI 10.1109/SEC54971.2022.00062

parameters by setting the parameters connected by a chosen symmetry to be equal. We consider the problem of Max-Cut and show numerically that on 68.0% of graphs that have a nontrivial symmetry group, the number of parameters can be reduced on average by 28.1% by using one of the symmetries without decreasing the objective function value. Inspired by this observation, we propose a modification of ma-QAOA that uses the full symmetry group (max-sym-QAOA). The full symmetry group can be obtained efficiently for many classes of graphs, and fast heuristic solvers can be used in practice [16]. We show that max-sym-QAOA reduces the number of parameters by 37.1% at the cost of only a 6.1% decrease in the objective. Moreover, we provide evidence of the centrality of the symmetries by showing that a random strategy with the same number of parameters yields much worse performance.

This paper is organized as follows. First, we introduce binary optimization, QAOA, and graph symmetry background material in Sec. II. In Sec. III we then discuss the methods used in this work. In Sec. IV we discuss our results, and we conclude with a discussion in Sec. V.

II. Background

We first briefly review the relevant background material and introduce the notation.

A. Binary Optimization Problem

We consider binary optimization(BO) problems of the form $\max_{x\in\{0,1\}^n} f(x)$, where $f(x)$ is a non-negative objective function over the Boolean cube $\{0,1\}^n$. It is often a sum of other functions that describe the system, called clauses.

When solving BO problems on quantum hardware, we construct a cost Hamiltonian H_c that encodes $f(x)$, so that $H_c |x\rangle = f(x) |x\rangle$. Then the optimization problem becomes

$$\max_{x\in\{0,1\}^n} \langle x| H_c |x\rangle .$$

The outcome of the algorithm is marked as x^*, and algorithm performance is typically quantified by the approximation ratio $r \in [0,1]$ given by

$$r := \frac{f(x^*)}{\max f(x)} = \frac{\langle x^*| H_c |x^*\rangle}{\max \langle x| H_c |x\rangle}. \tag{1}$$

B. QAOA

QAOA is a hybrid quantum-classical algorithm that finds approximate solutions to combinatorial optimization problems [8]. To solve a given optimization problem with QAOA, one must construct a cost Hamiltonian H_c that encodes the objective function and a mixer Hamiltonian H_m. Let $U(\gamma, C) = e^{-iH_c\gamma}$ and $U(\beta, B) = e^{-iH_m\beta}$, where γ and β are free parameters. These two unitaries are applied to an initial state $|s\rangle$, which is an eigenvector of H_m. The outcome of p iterations of the algorithm is denoted $|\vec{\gamma},\vec{\beta}\rangle_p$ and is

$$\begin{aligned}|\vec{\gamma},\vec{\beta}\rangle_p =&U(\beta_p, H_m)U(\gamma_p, H_c)\dots\\ &U(\beta_1, H_m)U(\gamma_1, H_c)|s\rangle .\end{aligned}$$

The parameters γ and β are chosen to maximize

$$E(\vec{\gamma},\vec{\beta}) = \langle \vec{\gamma},\vec{\beta}| H_c |\vec{\gamma},\vec{\beta}\rangle .$$

Measuring the state $|\vec{\gamma},\vec{\beta}\rangle$ gives an approximate solution to the BO problem encoded by H_c.

ma-QAOA is similar to QAOA; however, the definitions of $U(\gamma, C)$ and $U(\beta, B)$ are changed to

$$U(\vec{\gamma}, C) = e^{-i\sum_a C_a\gamma_a}$$

and

$$U(\vec{\beta}, B) = e^{-i\sum_b X_b\beta_b},$$

where C_a is a clause in the objective function and X_b is the Pauli-x operator acting on qubit b. Throughout this work, r_{y} refers to the approximation of y-QAOA, where y is a variation of QAOA.

C. QAOA on Max-Cut Problem

The Max-Cut problem is well studied in QAOA literature (e.g. [9], [17]) and is thus a natural problem to consider when studying QAOA variants. Given a simple graph $G = (V,E)$, the Max-Cut problem aims to partition V into two disjoint sets so that the number of edges with endpoints in both sets is maximized. This problem is NP-hard to solve exactly.

When solving the Max-Cut problem using QAOA, the cost Hamiltonian is

$$H_c = \sum_{uv\in E} \frac{1}{2}(I - Z_u Z_v),$$

and the mixer Hamiltonian is typically

$$H_m = \sum_{v\in V} X_v,$$

where Z_v and X_v are single-qubit Pauli operators acting on qubit v. Each layer of the QAOA circuit has one γ for all edges and one β for all vertices. The number of tunable parameters is just $2p$, independent of the graph size.

In ma-QAOA, each vertex and each edge has its own angle. Thus, there are $(|E| + |V|)\cdot p$ parameters to optimize in this modified algorithm. One drawback to this approach is that finding $(|E| + |V|)\cdot p$ parameters can be difficult, especially as the size of the problems grows. However, the algorithm has better performance than QAOA has on average [14].

D. Graph Symmetries

A graph automorphism is a permutation of the vertex set of a graph, $\sigma : V \to V$, that satisfies the condition that a pair of vertices, (u, v), forms an edge if and only if the pair $(\sigma(u), \sigma(v))$ also forms an edge. Automorphisms can be represented by products of disjoint cycles, $\sigma = \pi_1 \ldots \pi_k$, where $\pi_l = (i_1, i_2, \ldots, i_j)$ and each entry in the cycle is a unique integer. In this notation, $\sigma(i_a) = i_{a+1}$ modulo j. Any set of automorphisms can generate a corresponding vertex and edge orbit, which are the equivalence classes of the vertices (edges) of a graph G under the action of the automorphism.

A generator of a group is a set of automorphisms $(\sigma_1, \ldots, \sigma_n)$ containing group elements so that (possibly, repeated) application of the generators on themselves and each other can produce all elements in the group. In this work, we call the group generator of the automorphism group of the graph G the symmetry generator, and it can generate corresponding vertex and edge orbits. In this paper we denote the vertex (edge) orbit of the symmetry generator as the *maximum vertex (edge) orbit*, written as $\mathcal{O}_v(\mathcal{O}_e)$.

III. Methods

We use the symmetry structure of the problem to reduce the number of parameters in ma-QAOA. Symmetry is known to impact QAOA performance [16]. To analyze the role of symmetries, we introduce three modifications called best-1sym-QAOA, max-sym-QAOA, and rand-group-QAOA.

sym-QAOA selects a single automorphism of the target graph and assigns the same angle to all vertices in the same vertex orbit and the same angle to all edges in the same edge orbit. It then optimizes the QAOA parameters and executes the resulting QAOA circuit. best-1sym-QAOA runs sym-QAOA over all automorphisms of G and selects the one that gives the largest Max-Cut value.

When using max-sym-QAOA to solve Max-Cut on a graph G, the first step is to compute the symmetry generator of G and determine the corresponding maximum vertex orbit, $\mathcal{O}_v$, and edge orbit, $\mathcal{O}_e$. The algorithm requires $|\mathcal{O}_v| + |\mathcal{O}_e|$ parameters, where each element in the same vertex orbit or edge orbit receives the same parameter. The algorithm samples $|\mathcal{O}_v|+|\mathcal{O}_e|$ parameters randomly as initial parameters and runs the QAOA variational quantum circuit as a subroutine using those parameters. QAOA optimization steps are applied until the solution converges to the optimal solution or until the desired number of iterations has been performed. The formal algorithm is described in Alg. 1.

Finding the generating set of the automorphism group of a graph is an extra step in max-sym-QAOA compared with ma-QAOA. The time complexity of this step is at most quasi-polynomial since its polynomial time equivalent graph isomorphism problem can be solved by a quasi-polynomial algorithm [18]. Also, many polynomial time heuristics exist for specific classes of graphs such as **nauty** [19], [20] used in this work.

Algorithm 1: max-sym-QAOA

Input : Graph G, number of layers p.
Output: Optimized $\{\vec{\beta}, \vec{\gamma}\}$ approx. max-cut $\tilde{A}$

Construct cost Hamiltonian H_c from G
Find the symmetry generator of G and corresponding maximum vertex/edge orbit sets $\mathcal{O}_v$, $\mathcal{O}_e$
Sample $|\mathcal{O}_v| + |\mathcal{O}_e|$ initial parameters $\{\vec{\beta}, \vec{\gamma}\} = \{\beta_0, \ldots, \beta_{|\mathcal{O}_v - 1|}, \gamma_0, \ldots, \gamma_{|\mathcal{O}_e - 1|}\}$. Fix vertices/edges in the same orbit to have the same value $|\psi(\vec{\beta}, \vec{\gamma})\rangle \leftarrow$ QAOAcirc$(\{\vec{\beta}, \vec{\gamma}\}, p)$
$E(\vec{\beta}, \vec{\gamma}) \leftarrow \langle\psi(\vec{\beta}, \vec{\gamma})| H_c |\psi(\vec{\beta}, \vec{\gamma})\rangle$
$\{\vec{\beta}, \vec{\gamma}\}, x^* \leftarrow$ classical optimization algorithms to optimize $E(\vec{\beta}, \vec{\gamma})$

rand-group-QAOA groups vertices in the problem graph randomly into $|\mathcal{O}_v|$ sets and edges randomly into $|\mathcal{O}_e|$ sets, so that the number of parameters is the same as that of max-sym-QAOA.

The generator used in max-sym-QAOA is usually a set of automorphisms, so max-sym-QAOA is not always contained in best-1sym-QAOA, which ranges over all single automorphisms. Thus, max-sym-QAOA may perform better than best-1sym-QAOA.

IV. Results

In this work we implement one iteration of max-sym-QAOA, rand-group-QAOA, and best-1sym-QAOA, using the graph descriptions from [21]. We then compare the algorithms with one another and ma-QAOA using the data found in [22]. We use COBYLA to optimize the QAOA parameters, although we expect similar results to be obtained with other gradient-free and gradient-based local methods.

A. best-1sym-QAOA vs. ma-QAOA

Among all $7,565$ graphs with nontrivial symmetries, best-1sym-QAOA has fewer parameters than ma-QAOA has on $5,918$ graphs. Thus, we analyze only best-1sym-QAOA on those graphs. Figure 1 shows the difference in approximation ratios between best-1sym-QAOA and ma-QAOA for these $5,918$ graphs. best-1sym-QAOA has the same approximation ratio as ma-QAOA has on $5,097$ graphs, which is approximately 86.1% of the studied graphs, while using on average 28.1% fewer parameters than ma-QAOA uses.

We also quantify the ratio of the difference in the Max-Cut (approximation ratio) values of ma-QAOA and

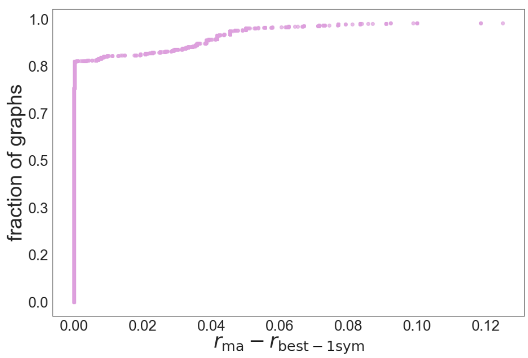

Fig. 1: Difference of approximation ratio r between ma-QAOA and best-1sym-QAOA, as defined in (1). For most graphs with symmetry, one symmetry can be used to reduce the number of parameters without affecting the solution quality.

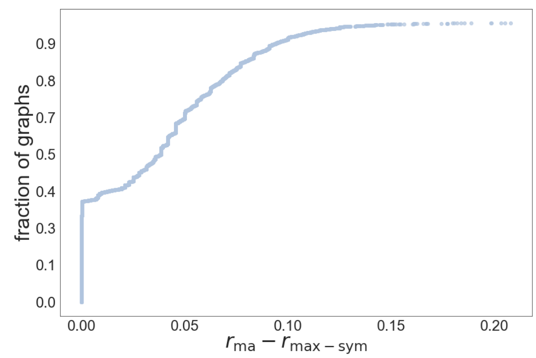

Fig. 2: Difference of approximation ratio r between ma-QAOA and max-sym-QAOA, as defined in (1). In a plurality of graphs with symmetry, simply using the largest symmetry leads to a reduction in the number of parameters with no impact on solution quality.

best-1sym-QAOA to the difference in the Max-Cut (approximation ratio) values of ma-QAOA and QAOA as

$$k_{\text{best-1sym-QAOA}} := \frac{f(x^*_{\text{ma}}) - f(x^*_{\text{best-1sym-QAOA}})}{f(x^*_{\text{ma}}) - f(x^*_{\text{QAOA}})}, \quad (2)$$

where $f(x^*_{\text{y}})$ denotes the approximate Max-Cut found by $\text{y} - QAOA$. When $k = 0$, best-1sym-QAOA recovers ma-QAOA; and when $k = 1$, best-1sym-QAOA performs the same as QAOA. Thus, this ratio indicates whether best-1sym-QAOA performance is closer to ma-QAOA performance or QAOA performance. In this study, the denominator of the ratio is always nonzero.

It is encouraging that the approximation ratios for best-1sym-QAOA and ma-QAOA are similar. But this finding is of limited value if both methods use the same number of parameters. We therefore consider the quantity $l_{\text{best-1sym-QAOA}}$, which is the relative difference in parameters between ma-QAOA and best-1sym-QAOA (as compared with the difference in the number of parameters between ma-QAOA and QAOA). That is,

$$l_{\text{best-1sym-QAOA}} := \frac{|E| + |V| - (|\mathcal{O}_e(\sigma)| + |\mathcal{O}_v(\sigma)|)}{|E| + |V| - 2}, \quad (3)$$

where $|\mathcal{O}_v(\sigma)|$ and $|\mathcal{O}_e(\sigma)|$ are the number of vertex orbits and edge orbits, respectively, induced by the automorphism σ. This ratio determines how close the number of parameters in best-1sym-QAOA is to either ma-QAOA or QAOA, depending on whether the ratio is closer to 0 or not.

B. *max-sym-QAOA vs. ma-QAOA*

Of the 11,117 connected, non-isomorphic 8-vertex graphs, $3,552$ have only trivial symmetries. In these cases max-sym-QAOA is ma-QAOA. Thus, our analysis focuses on the $7,565$ graphs that contain nontrivial symmetry, where max-sym-QAOA has fewer parameters to optimize over. As shown in Fig. 2, max-sym-QAOA performs as well as ma-QAOA on 2,713 of these graphs. Furthermore, it performs the same as QAOA on 30 graphs, which is only about 0.4% of the graphs with nontrivial symmetry. These results indicate that max-sym-QAOA performance is comparable, even though it requires fewer parameters.

We define the ratio of the difference in the Max-Cut values of ma-QAOA and max-sym-QAOA to the difference in Max-Cut values of ma-QAOA and QAOA as

$$k_{\text{max-sym}} := \frac{f(x^*_{\text{ma}}) - f(x^*_{\text{max-sym}})}{f(x^*_{\text{ma}}) - f(x^*_{\text{QAOA}})}. \quad (4)$$

When $k = 0$, max-sym-QAOA recovers ma-QAOA; and when $k = 1$, max-sym-QAOA performs the same as QAOA. Thus, this ratio indicates whether max-sym-QAOA's performance is closer to ma-QAOA's performance or QAOA's performance. In this study, the denominator of the ratio is always nonzero.

We quantify the ratio of the reduction of parameters from ma-QAOA to max-sym-QAOA over the difference in parameters between ma-QAOA and QAOA as

$$l_{\text{max-sym}} := \frac{|E| + |V| - (|\mathcal{O}_e| + |\mathcal{O}_v|)}{|E| + |V| - 2}, \quad (5)$$

since ma-QAOA uses $|E| + |V|$ parameters, max-sym-QAOA requires $|\mathcal{O}_e| + |\mathcal{O}_v|$ parameters, and QAOA uses two parameters.

Figure 3 shows that a positive correlation exists between the quantities k and l. Among results that achieve the result equivalent to ma-QAOA, the number of parameters reduced is spread nearly evenly over $[0.1, 0.8]$.

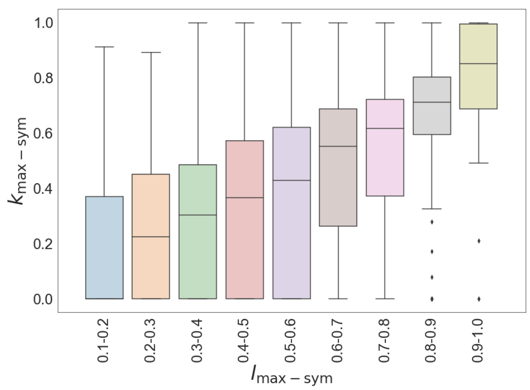

Fig. 3: Comparison of $k_{\text{max-sym}}$ and $l_{\text{max-sym}}$ for max-sym-QAOA, as defined in (4) and (5). There is a positive correlation between these two variables.

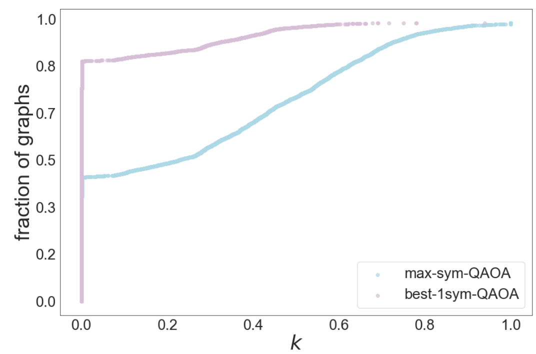

Fig. 5: Fraction of graphs achieving ratio k for max-sym-QAOA and best-1sym-QAOA, as defined in (2) and (4).

Since there are almost no points between $[0, 0.1]$, max-sym-QAOA almost always requires at least 10% fewer parameters than ma-QAOA requires. $l_{\text{max-sym}}$ averaged over all those graphs with nontrivial symmetries is 0.37.

Note that specifically for those graphs where reducing the number of parameters leads to a decrease in the objective, max-sym-QAOA can reduce the parameter count by 37.1% at the cost of only a 6.1% decrease in the objective.

C. Evidence for Central Role of Symmetries

max-sym-QAOA is also compared with rand-group-QAOA, which groups vertices and edges randomly so that the corresponding number of parameters is the same as that of max-sym-QAOA. Figure 4 demonstrates the centrality of symmetries to ma-QAOA parameter redundancy by showing that the parameter reduction strategy of max-sym-QAOA has a clear advantage over rand-group-QAOA in terms of solution quality.

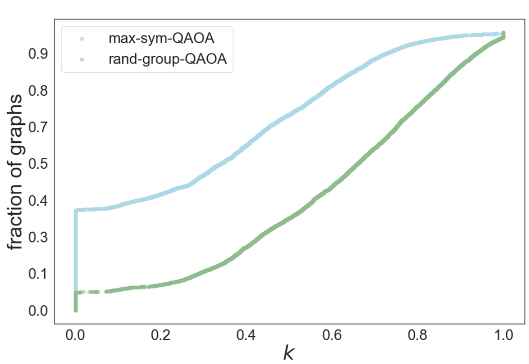

Fig. 4: Fraction of graphs achieving ratio k for max-sym-QAOA and rand-group-QAOA. If a random parameter reduction is used, the performance deteriorates significantly, suggesting the central role of symmetries.

D. max-sym-QAOA *vs.* best-1sym-QAOA

In this section we compare max-sym-QAOA and best-1sym-QAOA on the 5,918 graphs for which best-sym-QAOA requires fewer parameters than does ma-QAOA.

Figure 5 indicates that best-1sym-QAOA has $k = 0$ on nearly twice as many graphs as max-sym-QAOA. Additionally, $k < 0.6$ for the majority of graphs solved with best-1sym-QAOA while k is spread over $[0, 1]$ with max-sym-QAOA.

Although max-sym-QAOA does not perform as well as best-1sym-QAOA on the majority of graphs, the average $l_{\text{max-sym}}$ is around 0.39 while the average $l_{\text{best-sym}}$ is only 0.31, so best-1sym-QAOA has more parameters than max-sym-QAOA, on average.

V. Discussion

In this work we demonstrate the connection between the parameter redundancy in ma-QAOA and the symmetries of the problem to be optimized. Specifically, we show that the number of parameters in ma-QAOA can often be dramatically reduced without affecting the solution quality. To that end, we introduce three QAOA variations that require fewer parameters than ma-QAOA: best-1sym-QAOA, max-sym-QAOA, and rand-group-QAOA. The three algorithms assign classical parameters based on the symmetries (automorphisms) of the underlying problem graph. We evaluate these algorithms on all connected, non-isomorphic 8-vertex graphs and compare the results with those of ma-QAOA.

In most cases, max-sym-QAOA requires at least 10% fewer parameters than ma-QAOA does, while maintaining a comparable approximation ratio, which is the primary metric of QAOA success. In fact, in over one-third of the connected 8-vertex graphs with nontrivial symmetry, max-sym-QAOA finds the same approximate Max-Cut as does ma-QAOA. Furthermore, a positive correlation exists between the number of parameters reduced and the reduction in the approximation ratio, as expected. Additionally, significantly more graphs have $k = 0$ when solved with max-sym-QAOA than rand-group-QAOA, implying that max-sym-QAOA outperforms rand-group-QAOA in general. On the other hand, significantly more graphs have $k = 0$ when solved with best-1sym-QAOA than max-sym-QAOA, yet max-sym-QAOA needs fewer parameters (on average) than does best-1sym-QAOA.

Thus, out of best-1sym-QAOA, max-sym-QAOA, and rand-group-QAOA, rand-group-QAOA appears to have the worst performance while best-1sym-QAOA appears to have the closest performance to ma-QAOA on these small graphs. The failure of rand-group-QAOA demonstrates the importance of symmetry to parameter setting in QAOA.

Approximately one-third of the graphs considered in this study had only trivial symmetry, and max-sym-QAOA is equivalent to ma-QAOA in these cases. Nonetheless, numerical evidence in [14] suggests that for these graphs the redundancy in parameters is also present. Therefore, an interesting future direction is understanding how the redundancy in parameters can be reduced for graphs with no symmetries.

Sauvage et al. proposed using symmetries to improve the performance of variational quantum algorithms [23]. They observed that in ma-QAOA applied to the Max-Cut problem, the number of parameters can be reduced and the trainability improved by using an approach equivalent to max-sym-QAOA described in this work. Unlike [23], we highlight the role of symmetries in quantum optimization by showing that symmetry-based parameter reduction leads to much better performance than does a random approach. Moreover, we consider utilizing a part of the symmetry group (best-1sym-QAOA).

References

[1] F. Arute *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[2] Y. Wu *et al.*, "Strong quantum computational advantage using a superconducting quantum processor," *Physical Review Letters*, vol. 127, no. 18, p. 180501, 2021.

[3] L. S. Madsen *et al.*, "Quantum computational advantage with a programmable photonic processor," *Nature*, vol. 606, no. 7912, pp. 75–81, 2022.

[4] M. Ringbauer *et al.*, "A universal qudit quantum processor with trapped ions," *Nature Physics*, pp. 1–5, 2022.

[5] C. C. McGeoch and C. Wang, "Experimental evaluation of an adiabiatic quantum system for combinatorial optimization," in *Proceedings of the ACM International Conference on Computing Frontiers*, 2013, pp. 1–11.

[6] N. Mohseni *et al.*, "Ising machines as hardware solvers of combinatorial optimization problems," *Nature Reviews Physics*, vol. 4, no. 6, pp. 363–379, 2022.

[7] T. Hogg and D. Portnov, "Quantum optimization," *Information Sciences*, vol. 128, no. 3-4, pp. 181–197, 2000.

[8] E. Farhi *et al.*, "A quantum approximate optimization algorithm," *arXiv:1411.4028*, 2014.

[9] M. P. Harrigan *et al.*, "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor," *Nature Physics*, vol. 17, no. 3, pp. 332–336, 2021.

[10] A. Kakkar *et al.*, "Characterizing error mitigation by symmetry verification in QAOA," *arXiv:2204.05852*, 2022.

[11] P. Niroula *et al.*, "Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer," 2022.

[12] R. Shaydulin and A. Galda, "Error mitigation for deep quantum optimization circuits by leveraging problem symmetries," in *International Conference on Quantum Computing and Engineering*. IEEE, 2021.

[13] M. Cerezo *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[14] R. Herrman *et al.*, "Multi-angle quantum approximate optimization algorithm," *Scientific Reports*, vol. 12, no. 1, pp. 1–10, 2022.

[15] R. Herrman, "Investigating parameters in the multi-angle approximate optimization algorithm," in *APS March Meeting Abstracts*, vol. 2022, 2022, pp. G00–075.

[16] R. Shaydulin *et al.*, "Classical symmetries and the quantum approximate optimization algorithm," *Quantum Information Processing*, vol. 20, no. 11, pp. 1–28, 2021.

[17] R. Shaydulin and Y. Alexeev, "Evaluating quantum approximate optimization algorithm: A case study," in *International Green and Sustainable Computing Conference*. IEEE, 2019, pp. 1–6.

[18] L. Babai, "Graph isomorphism in quasipolynomial time," in *Proceedings of the ACM Symposium on Theory of Computing*, 2016, pp. 684–697.

[19] B. D. McKay, "Nauty user's guide (version 2.4)," *Computer Science Dept., Australian National University*, pp. 225–239, 2007.

[20] B. D. McKay and A. Piperno, "Nauty and Traces user's guide (Version 2.5)," *Computer Science Department, Australian National University, Canberra, Australia*, 2013.

[21] B. McKay, "Graphs [data set]," found at http://users.cecs.anu.edu.au/ bdm/data/graphs.html.

[22] P. C. Lotshaw and T. S. Humble, "QAOA dataset," found at https://code.ornl.gov/qci/qaoa-dataset-version1.

[23] F. Sauvage *et al.*, "Building spatial symmetries into parameterized quantum circuits for faster training," *arXiv:2207.14413*, 2022.