Classical symmetries and QAOA

Ruslan Shaydulin* [†‡] Stuart Hadfield [†§] Tad Hogg [†] Ilya Safro [¶]

Abstract

We study the relationship between the Quantum Approximate Optimization Algorithm (QAOA) and the underlying symmetries of the objective function to be optimized. Our approach formalizes the connection between quantum symmetry properties of the QAOA dynamics and the group of classical symmetries of the objective function. The connection is general and includes but is not limited to problems defined on graphs. We show a series of results exploring the connection and highlight examples of hard problem classes where a nontrivial symmetry subgroup can be obtained efficiently. In particular we show how classical objective function symmetries lead to invariant measurement outcome probabilities across states connected by such symmetries, independent of the choice of algorithm parameters or number of layers. To illustrate the power of the developed connection, we apply machine learning techniques towards predicting QAOA performance based on symmetry considerations. We provide numerical evidence that a small set of graph symmetry properties suffices to predict the minimum QAOA depth required to achieve a target approximation ratio on the MaxCut problem, in a practically important setting where QAOA parameter schedules are constrained to be linear and hence easier to optimize.

1 Introduction

Advances in quantum hardware such as the recent demonstration of quantum supremacy [1] have paved the way for the deployment of algorithms that potentially offer quantum advantage in applications, that is, outperform state-of-the-art classical algorithms for some computationally challenging problem. Combinatorial optimization is a particularly attractive and important target domain as optimization problems are omnipresent in science and industry and the difficulty of such problems means one must often settle for approximation algorithms or heuristics in practice. The Quantum Approximate Optimization Algorithm (QAOA) [2, 3] is a leading candidate algorithmic paradigm leveraging near-term quantum hardware to approximately solve certain classes of hard combinatorial optimization problems. Published results to date have shown experimental demonstrations of QAOA on quantum hardware using up to 23 qubits [4]. Nevertheless, although a number of recent results have begun to shed some light on the potential of QAOA as compared with classical algorithms and heuristics, the power of QAOA is not yet well understood.

Nontrivial performance guarantees for QAOA circuits have thus far been obtained only in very limited settings, with the majority of rigorous bounds to the approximation ratio achievable by QAOA obtained only for small (high-level QAOA circuit depth) p, primarily $p = 1$ or $p = 2$ [2, 5, 6, 7, 8]. A notable exception is [9], which shows that QAOA-related circuits can recover the well-known quadratic speedup for Grover's unstructured search problem. Furthermore, such performance results are typically worst case and do not take advantage of the structure of a

*Argonne National Laboratory, Lemont, IL 60439, USA
†Quantum Artificial Intelligence Lab (QuAIL), NASA Ames Research Center, Moffett Field, CA 94035, USA
‡KBR, Houston, TX 77002, USA
§USRA Research Institute for Advanced Computer Science (RIACS), Mountain View, CA 94043, USA
¶University of Delaware, Newark, DE 19716, USA

given problem instance, nor do they give much insight into the performance of deeper QAOA circuits. Indeed, initial numerical studies on small problem instances have indicated that going beyond small p appears necessary to obtain potential quantum advantage [10, 6, 11, 12]. From the perspective of lower bounds, a recent paper of Hastings [13] demonstrates that for QAOA with $p = 1$, classical local algorithms can achieve the same (or better) performance guarantees as QAOA; the author argues that for some problems increasing the QAOA depth by any bounded amount would not lead to significant improvements in performance. Farhi et al. [14] show an upper bound for the approximation ratio attainable by a particular realization of QAOA for the MaxIndependentSet problem on d-regular graphs with depth p growing slower than logarithmically with the number of nodes, implying that p must grow at least logarithmically for QAOA to outperform a particular approximation ratio. They note the observed relationship between the required depth and the graph diameter, suggesting that for graphs with larger diameter (e.g., 2d lattices) the depth required to obtain a quantum advantage in this case may be polynomially scaling in the problem size.

While clearly important, these findings leave many practical questions unresolved, such as "What is the QAOA depth required to adequately solve a given problem instance?" For example, Farhi et al. [14] note that for a class of 3-regular graphs with 2 million qubits, the upper bound they prove yields a necessary depth of only $p = 7$.

A promising approach to overcoming limitations of the previous analyses is the application of symmetry ideas and analysis to QAOA. Understanding symmetry has long been central to the analyses of topological phases of matter [15], the hardness of ground state preparation for practically interesting local Hamiltonians [16, 17], mathematical programming [18, 19], and quantum query complexity [20], among other important applications in quantum computing. Much insight likewise may be gained through the connection between the symmetries of the QAOA ansatz (that is, the symmetries shared by the QAOA quantum circuit and initial state) and the symmetries of the target classical optimization problem. Indeed, recently Bravyi et al. [21] have utilized the $\mathbb{Z}_2$ symmetry exhibited by the MaxCut problem and its corresponding QAOA ansatz together with the properties of bipartite expander graphs to show that constant-depth QAOA for MaxCut is outperformed by the classical Goemans–Williamson algorithm for some families of graphs. This result highlights the potential of symmetry insights in advancing our understanding of the performance and limitations of QAOA and related approaches to quantum optimization.

In this paper we focus on QAOA as originally proposed in [2] for clarity of presentation, although our approach and analysis serve as a template that may be adapted to more general QAOA ansatzë [3, 22] or other parameterized quantum circuits. Our main technical result (Theorem 1) connects the classical symmetries of the problem objective (cost) function to be optimized to symmetries of the QAOA dynamics, operationalized in terms of QAOA outcome probabilities. We show that the binary strings connected by a symmetry of the objective function will have the same output probabilities. For the special case of variable (qubit) permutation symmetries, the theorem implies the following simple corollary.

Corollary 1. *Suppose a group of variable permutations leaves the objective function invariant. Then QAOA output probabilities are the same across all bit strings connected by such permutations, for all fixed choices of QAOA parameters and depth.*

More generally, Sec. 3 shows that symmetries of the classical objective function induce a group of symmetries of the QAOA ansatz, and this leads to conserved probability amplitudes over the subspace of states connected by these symmetry transformations. Our results remain applicable even if we know only some convenient subgroup of the group of all possible symmetries of the classical objective function. An important and well-known fact is that such symmetries can imply quantum dynamics restrict to a reduced Hilbert space. For QAOA we quantify the relationship between the group of symmetries and the resulting dimension of the effective Hilbert space in terms of group properties (Theorem 2), which in some cases can be exponentially reduced, and we discuss implications to classical simulation. We illuminate our results with several examples.

In Sec. 4 we show an illustrative application of our results by extending our framework to the practical task of predicting the QAOA depth required to achieve a desired approximation ratio, in the setting of linear QAOA parameter schedules. We demonstrate a machine learning approach to the instance-wise analysis of QAOA that constructs symmetry groups and a number of related symmetry measures of the classical optimization problem in order to extract features used to predict the required QAOA depth. We focus specifically on MaxCut, which is a standard problem considered for QAOA [2, 6, 7, 12, 11, 10, 23, 21, 24], with symmetries related to those of the underlying graph, though our approach may be extended to other problems of interest. We note that for many practically important but classically hard classes of problem instances, such as MaxCut on bounded degree graphs, the full symmetry group of the graph can always be constructed in polynomial time [25].

The paper is organized as follows. Section 2 presents an overview of the relevant background. Section 3 presents a series of analytical results exploring the connections between classical symmetry and QAOA. Section 4 provides an example application of using symmetry considerations to predict the QAOA depth required achieve a target approximation ratio for MaxCut in the case of linear QAOA schedules. The methods and numerical results in Sec. 4 are mostly self-contained, and the interested reader may proceed there directly, independently of the details and proofs of Sec. 3. Section 5 concludes our study with an outlook toward future applications.
Reproducibility: The Python code for the numerical experiments of Sec. 4 and the generated dataset are available online [26, 27].

2 Background

In this section we briefly review our notions of binary optimization, QAOA, and classical symmetries. Though for simplicity we consider maximization problems our results apply similarly to minimization.

Binary optimization Consider a class of problem instances, that is, nonnegative objective functions $f(x)$ on the Boolean cube $\{0,1\}^n$, with the corresponding optimization problem

$$\max_{x\in\{0,1\}^n} f(x). \tag{1}$$

The objective function f may be encoded as the n-qubit Hamiltonian C, which acts as $C\left|x\right\rangle = f(x)\left|x\right\rangle$ for each $x \in \{0,1\}^n$, namely, the diagonal matrix in the computational basis

$$C = diag(f(x)).$$

Compact representations of such Hamiltonians (in terms of Ising spin, i.e., Pauli-Z matrices) can be constructed efficiently for many combinatorial optimization problems [28]. A candidate solution $x^* \in \{0,1\}^n$ is an r-approximation for a given instance, $0 \le r \le 1$ if

$$f(x^*) \;\geq\; r \cdot \max_x f(x), \tag{2}$$

and an algorithm is said to achieve approximation ratio r for (1) if it returns an r-approximation or better for every problem instance in the class (i.e., in the worst case). For many applications (e.g., NP-hard problems) the best *efficient* algorithms known can achieve only some $r < 1$, where r may or may not depend on the problem size [29]. Similar definitions apply for minimization problems.

MaxCut An objective function can be defined on a graph $G = (V, E)$ by choosing an appropriate encoding. For graph problems such as MaxCut, the commonly used encoding we consider assigns a binary indicator variable to each vertex such that variable assignments correspond to subsets of vertices. With this choice of encoding, the set of binary strings $x \in \{0,1\}^{|V|}$ encodes the possible configurations of the graph using $|V|$ qubits; note that alternative choices of encoding may give different configuration spaces with potentially different resource requirements such as number of bits. For MaxCut the objective is find a subset of vertices (i.e., partition the vertices V into two disjoint parts) such that the number of cut graph edges (those with an endpoint in each part) is maximized. The MaxCut problem on general graphs is APX-complete [30], which means it has no polynomial-time approximation scheme unless P=NP (i.e., no efficient deterministic classical algorithm can find a cut achieving an approximation ratio better than some constant), although efficient algorithms exist for particular classes of graphs [31]. In light of this computational difficulty, MaxCut has been considered extensively for QAOA; see, for example, [2, 6, 21]. In the standard problem mapping the objective function is encoded by the Hamiltonian

$$C_{\text{MaxCut}} = \frac{1}{2} \sum_{(u,v)\in E} (I - Z_u Z_v),$$

where Z_u indicates the single-qubit Pauli-Z operator applied to the qubit corresponding to vertex u. Although in this paper we consider MaxCut, we emphasize that our approach and results are general and may be applied to other problems of interest.

QAOA The Quantum Approximate Optimization Algorithm is a hybrid quantum-classical algorithm that combines a parameterized quantum evolution with a classical outer-loop optimizer to (approximately) solve binary optimization problems [2, 3]. At each call to the quantum computer, a trial state is prepared by applying a sequence of quantum operators in alternation

$$|\vec{\beta}, \vec{\gamma}\rangle_p := U_B(\beta_p) U_C(\gamma_p) \dots U_B(\beta_1) U_C(\gamma_1) |s\rangle, \tag{3}$$

where C is the problem Hamiltonian, $U_C(\gamma) = e^{-i\gamma C}$ is the phase operator, $U_B(\beta)$ is the mixing operator, and $|s\rangle$ is some easy-to-prepare initial state. The parameterized quantum circuit (3) is called the *QAOA ansatz*. We refer to the number of alternating operator pairs p as the *QAOA depth*. The selected parameters $\vec{\beta}, \vec{\gamma}$ are said to define a *schedule*, analogous to a similar choice in quantum annealing. For unconstrained optimization problems as we consider in this paper,[1] the standard mixing operator is $U_B(\beta) = e^{-i\beta \sum_j X_j}$ which corresponds to time evolution under the transverse field Hamiltonian $B := \sum_j X_j$, and the initial state is the uniform superposition product state $|s\rangle := \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle = |+\rangle^{\otimes n}$. Here X_j similarly denotes the single-qubit Pauli-X operator applied to qubit j.

Preparation of the state (3) is followed by a measurement in the computational basis; we let $\mathsf{Pr}_p(x)$ denote the probability of obtaining bitstring $x \in \{0,1\}^n$ with

$$\mathsf{Pr}_p(x) = |\langle x|\vec{\beta}, \vec{\gamma}\rangle_p|^2.$$

The output of repeated state preparation and measurement may be used by a classical outer-loop algorithm to select the schedule $\vec{\beta}, \vec{\gamma}$. We consider optimizing the expectation value $\langle C\rangle_p = \langle\vec{\beta}, \vec{\gamma}|_p C |\vec{\beta}, \vec{\gamma}\rangle_p$ as originally proposed in [2], although other outer-loop objectives have been suggested in the literature; see, for example, [32]. The output of the overall procedure is the best solution found for the given problem. We emphasize that the task of finding good QAOA parameters appears challenging in general. In Sec. 4 we explore a strategy for reducing the number of free parameters towards mitigating this issue. Hence we say that depth p QAOA

[1]Alternative problem encodings, mixing operators, and initial states for QAOA have been proposed; see e.g. [3].

achieves approximation ratio r for a problem instance $f(x)$ if there exist parameters $\vec{\beta}, \vec{\gamma}$ such that

$$\langle C \rangle_p := \langle \vec{\beta}, \vec{\gamma}|_p \, C \, |\vec{\beta}, \vec{\gamma}\rangle_p \; \geq \; r \cdot \max_x f(x), \tag{4}$$

and similarly in the case of *QAOA with parameter schedules* where $\vec{\beta}, \vec{\gamma}$ satisfying (4) are restricted to a specified set of allowed values as considered in Sec. 4. Note that under this definition the approximation ratio can be evaluated exactly in simulation, but only approximately on a quantum computer due to finite sampling error.

Classical symmetries We define a *classical (objective function) symmetry* to be a permutation of the Boolean cube mapping each $x \to x'$ such that the objective function $f(x) = f(x')$ is unchanged, for each $x \in \{0,1\}^n$. More formally, we consider the symmetric group S_{2^n} of permutations on n-bit strings [33] (i.e., permutations of 2^n objects) and its subgroups (called permutation groups), in particular the natural subgroup $S_n \subset S_{2^n}$ induced by the group of permutations of the bit indices. (We use S_n to denote this important subgroup throughout.) Then a *symmetry* of the objective function $f(x)$ is a transformation $a \in S_{2^n} : \{0,1\}^n \to \{0,1\}^n$ such that for each $x \in \{0,1\}^n$ we have $f(x) = f(a(x))$. One can easily see that a set of such symmetries gives a group under composition; indeed, we leverage several ideas from group theory in our results below.

On qubits, each symmetry $a(x)$ may be faithfully represented as the permutation matrix $A = \sum_{x \in \{0,1\}^n} |a(x)_1 \dots a(x)_n\rangle \langle x_1 \dots x_n|$, which acts as $A\,|x\rangle = |a(x)\rangle$ for each computational basis state $|x\rangle$ and is unitary $A^\dagger A = I$. With this representation in mind, we will sometimes write $A \in S_{2^n}$, in place of the permutation a. (In particular, we will also use S_n to denote permutations on n qubits.) Then an objective function having symmetry $A \in S_{2^n}$ is equivalent to the condition $A^\dagger C A = C$ for its Hamiltonian representation C. For example, this condition is always satisfied for the MaxCut Hamiltonian C_{MaxCut} above for the symmetry $A = X_1 X_2 \dots X_n \in S_{2^n}$, which corresponds to flipping all bits simultaneously. Similarly, we say a general operator Q acting on n qubits is invariant under a symmetry A if it satisfies $A^\dagger Q A = Q$.

Any set of objective function symmetries $\{A_1, \dots, A_\ell\} \subset S_{2^n}$ generates a group under composition. Two bit strings $x, y \in \{0,1\}^n$ are then said to be in the same orbit if there exists a symmetry group element A such that $|y\rangle = A\,|x\rangle$, which gives an equivalence relation on $\{0,1\}^n$ (or, equivalently, on $\{|x\rangle : x \in \{0,1\}^n\}$). We remark that our consideration of permutation groups is general since from Cayley's theorem [33] every finite group is isomorphic to a permutation group on some underlying set. Two important observations for our results to follow are that the standard QAOA initial state $|s\rangle = |+\rangle^{\otimes n}$ satisfies $A\,|s\rangle = |s\rangle$ for any $A \in S_{2^n}$ and that for any qubit permutation $A \in S_n$ the transverse-field mixing Hamiltonian is invariant $A^\dagger B A = B$.

Graph symmetries A symmetry of a graph is a transformation that leaves the graph invariant. For optimization problems on a graph such as MaxCut with a given encoding, graph symmetries induce classical symmetries, typically corresponding to a subgroup of the qubit permutations S_n. Indeed, we will primarily consider the well-studied class of graph (vertex) automorphisms, that is, permutations of the vertices that map a graph onto itself. Objective functions on graphs that do not depend on the vertex or edge labels are naturally invariant under such transformations. More formally, for a graph $G = (V, E)$, an automorphism is a permutation $\sigma : V \to V$ such that $(\sigma(v), \sigma(u)) \in E$ if and only if $(u, v) \in E$. One can easily see that the automorphisms of a given graph form a group under composition [34]. While the size (order) of the group of automorphisms can be as large as $n!$ for structured graphs, random graphs generally have much smaller automorphism groups. For example, random three-regular graphs or Erdos–Renyi model graphs almost surely have no nontrivial automorphisms [35, 36], though practical applications may involve classes of graphs with useful structure; see Sec. 3.3

for additional discussion. Two vertices u and v are said to belong to the same orbit if there exists an automorphism σ such that $\sigma(v) = u$, and orbits induce equivalence classes on the vertex set V. For an in-depth presentation of graph automorphisms and their properties the reader is referred to [34, Part Three].

Our results in the next section consider graph symmetries of objective functions, as well as classical symmetries more generally. Our application to MaxCut in Sec. 4 considers graph automorphisms as well as several related notions of graph symmetry described therein.

3 Classical symmetries in QAOA

Various general notions of symmetry for quantum algorithms are possible, and different notions may be appropriate in different applications. When an underlying classical function is concerned, for example an objective function over a set of configurations on a graph, a natural direction to explore is what its symmetries from the classical perspective can tell us about derived quantum algorithms such as QAOA. We consider classical symmetries of the objective function (i.e., problem Hamiltonian) and show how they relate to symmetries of the QAOA ansatz and its resulting measurement outcome probabilities. In particular we consider the special case of graph symmetries. While in general it may not be possible to efficiently obtain the full symmetry group of the objective function, in the case of graph symmetries for many (but not all) practically interesting classes of problems there exist polynomial-time algorithms or fast heuristic solvers that often give the full graph automorphism symmetry group; we elaborate on these considerations in Sec. 3.3. An appealing aspect of our results is that they apply regardless, i.e., when only some subgroup of classical symmetries is known.

For QAOA, symmetries relate to both the phase and mixing operators (and hence to the problem and mixing Hamiltonians). For classical symmetries shared by both operators, Theorem 1 (and Lemma 1 more generally) shows how such symmetries restrict the QAOA dynamics: *QAOA amplitudes in the computational basis are always the same for states connected by the the group of such symmetry transformations, and hence likewise for measurement probabilities.* Our results are general and are applicable to a wide class of optimization problems. Note that useful symmetries may arise from the problem encoding, the particular problem class or instance, or the quantum states and operators themselves. Although we focus in this paper on QAOA with the standard transverse-field mixer and initial state, our results may be similarly extended to more general mixing operators with potentially different symmetries, such as those described in [3, 22], or to different choices of QAOA initial state.

We now state our main theorem, followed by several useful corollaries, as well as an explicit example of QAOA symmetry for an instance of MaxCut. The proof of the theorem is given using a more general lemma presented in Sec. 3.1.

Theorem 1. *Consider a depth-p QAOA state $|\vec{\beta}, \vec{\gamma}\rangle_p = U_B(\beta_p)U_C(\gamma_p)\dots U_B(\beta_1)U_C(\gamma_1)|s\rangle$ with objective Hamiltonian C, transverse-field mixing Hamiltonian B, and initial state $|s\rangle = |+\rangle^{\otimes n}$.*

- *Given a classical symmetry matrix $A \in S_{2^n}$ implementing the permutation $A|x\rangle = |a(x)\rangle$ for each $x \in \{0,1\}^n$ such that (i) $[A, C] = 0$ and (ii) $[A, B] = 0$, then for all $p, \vec{\beta}, \vec{\gamma}$, the measurement probability of all bit strings connected by A is the same:*

$$\forall x \in \{0,1\}^n \qquad \mathsf{Pr}_p(x) = \mathsf{Pr}_p(a(x)). \tag{5}$$

- *Suppose we know a number of classical symmetries $A_1, \dots A_\ell \in S_{2^n}$ such that (i) $[A_j, C] = 0$ and (ii) $[A_j, B] = 0$ for each $j = 1, \dots, \ell$. Then taking matrix products of the A_j generates the QAOA symmetry group*

$$\mathcal{A} = \{A_0 = I, A_1, \dots A_\ell, A_{\ell+1} \dots A_{\ell'}\} \subset S_{2^n}, \tag{6}$$

which acts as $A_j |x\rangle = |a_j(x)\rangle$ for each $x \in \{0,1\}^n$, $A_j \in \mathcal{A}$, and for all $p, \vec{\beta}, \vec{\gamma}$,

$$\forall x \in \{0,1\}^n, \ \ \forall A_j \in \mathcal{A} \qquad \mathsf{Pr}_p(x) = \mathsf{Pr}_p(a_j(x)), \tag{7}$$

i.e., probabilities are the same across all strings in the orbit $\mathcal{A} \cdot x$.

- *In either case a symmetry $A \in S_{2^n}$ satisfies $[A, C] = 0 = [A, B]$ if and only if (i') $c(a(x)) = c(x)$ and (ii') $\{a(x^{(j)}) : j = 1, \ldots, n\} = \{a(x)^{(j)} : j = 1, \ldots n\}$ as subsets of $\{0,1\}^n$, where $y^{(j)}$ denotes the bitstring y with its jth bit flipped.*

We remark that for the transverse field mixer $B = \sum_j X_j$, qubit permutations $A \in S_n \subset S_{2^n}$ always satisfy condition (ii) of the theorem, although $A \in S_n$ is not necessary. Corollary 1 of the Introduction then follows trivially observing that the variable permutation symmetry of the objective function implies qubit permutation symmetry of the objective Hamiltonian. An example satisfying $[A', B] = 0$ but where $A' \notin S_n$ is $A' = X_1 X_2 \ldots X_n \in S_{2^n}$ which flips all bits simultaneously. We emphasize that the theorem may be applied instance-wise or in some cases over a class of problem instances. For example, A' applies to all instances of MaxCut, corresponding to the well-known $\mathbb{Z}_2$-symmetry of that problem, with additional symmetries arising from the automorphisms of the particular graph instance (see Cor. 2 below).

Theorem 1 and Corollary 1 consider general objective function symmetries. A practically important subset of such symmetries are ones that can be understood as the symmetries of some graph derived from the structure of the objective function and corresponding problem Hamiltonian. Following for instance [16], we refer to this graph as the *interaction graph* of a given problem instance. For example, for MaxCut and problems similarly defined over a set of configurations on a graph, the interaction graph is typically the underlying graph on which the problem is defined.[2] For an arbitrary polynomial objective function defined on the Boolean cube, we can define an interaction graph by fixing variable labeling, letting each variable be a vertex, and connecting two vertices by an edge if there is a term in the polynomial including the corresponding variables. For an arbitrary pseudo-Boolean function (including constraint satisfaction problems, commonly considered in QAOA) such polynomial representations exist and can be constructed by considering the function's Fourier expansion [28]. Hence, Cor. 1 implies that symmetries $A \in S_n$ of the interaction graph always yield QAOA symmetries, although as explained not all symmetries of the objective function (problem Hamiltonian) are graph symmetries in general. We formalize this for the case of problems on graphs.

Corollary 2. *Consider an n-vertex graph G with some (label-independent) objective function f defined on its vertex configurations $\{0,1\}^n$. Suppose we know a subgroup of graph automorphisms $\mathcal{A}_G \subseteq Aut(G) \subseteq S_n \subset S_{2^n}$. Then for all $p, \vec{\beta}, \vec{\gamma}$ the corresponding QAOA states satisfy*

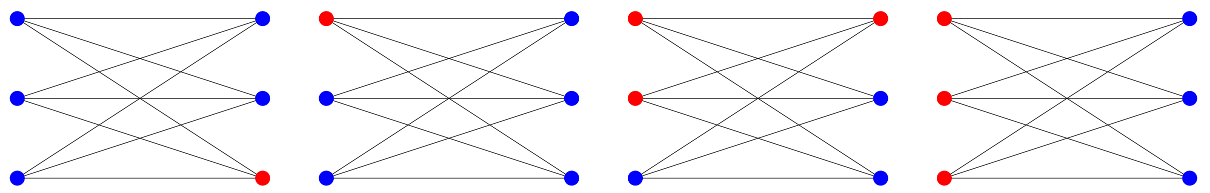
$$\forall x \in \{0,1\}^n \ \ \forall a_j \in \mathcal{A}_G \qquad \mathsf{Pr}_p(x) = \mathsf{Pr}_p(a_j(x)). \tag{8}$$

Generally, $Aut(G)$ may be a proper subgroup of the full QAOA symmetry group $\mathcal{A}$.

Proof. Each permutation matrix $A \in S_n$ representing a graph automorphism satisfies the conditions of the theorem as in Corollary 1. As mentioned the MaxCut symmetry $A' = X_1 X_2 \ldots X_n$ of flipping all bits shows not all symmetries correspond to graph automorphisms. $\square$

Example: QAOA for MaxCut on $K_{3,3}$ To illustrate our results consider MaxCut on the 6-node complete bipartite graph $K_{3,3}$. This graph has a single vertex orbit (see Fig. 1). Recall that

[2]Note that in cases where the terms of the objective polynomial do not all have equal coefficients (as they do in the case of, e.g., MaxCut), the relevant symmetries of the interaction graph are *weighted* automorphisms (i.e., automorphisms preserving edge weights [37]). Obtaining the group of weighted automorphisms is in general more difficult than in the nonweighted case, and the introduction of weights makes the problem less likely to have nontrivial automorphisms; hence we do not consider them in further detail.



(a) 100000 (b) 000001 (c) 000111 (d) 010101

Figure 1: MaxCut on $K_{3,3}$ graph: All nodes of the graph are on the same vertex orbit in V (i.e., there exists an automorphism that takes one into another; for example, Fig. 1a can be taken into Fig. 1b by a vertical mirror symmetry applied after horizontal mirror symmetry). Corollary 2 implies all bitstrings with Hamming weight 1 have the same QAOA measurement probabilities. On the other hand, no graph automorphism takes the red nodes in Fig. 1c into those of Fig. 1d.

vertex orbits induce orbits on bitstrings. Hence from Corollary 2 all bitstrings with Hamming weight one will belong to the same orbit in $\{0,1\}^n$ and therefore have the same probabilities in QAOA state $|\vec{\beta},\vec{\gamma}\rangle_p$, regardless of depth p or the choice of parameters $\vec{\beta},\vec{\gamma}$. At the same time, there does not exist a graph automorphism taking the assignment 000111 to 010101, as may be anticipated by observing that Fig. 1c corresponds to fewer cut edges than does Fig. 1d, and so QAOA may yield different measurement probabilities for such pairs of bitstrings.

3.1 Symmetry preserves probability amplitudes

More generally than the measurement probability viewpoint of Theorem 1, we show that QAOA symmetries also lead to identical probability amplitudes of computational basis states connected by suitable classical symmetries.

Lemma 1. *Consider depth-p QAOA $|\vec{\beta},\vec{\gamma}\rangle_p = U_B(\beta_p)U_C(\gamma_p)\ldots U_B(\beta_1)U_C(\gamma_1)\,|s\rangle$ with classical problem Hamiltonian C, mixing Hamiltonian B, and initial state $|s\rangle = |+\rangle^{\otimes n}$.*

Suppose we know a symmetry $A \in S_{2^n}$ acting as $A\,|x\rangle = |a(x)\rangle$, $x \in \{0,1\}^n$, such that

i) $[A, U_C(\gamma)] = 0,\ \forall\gamma \in R_\gamma \subset \mathbb{R}$ *and*

ii) $[A, U_B(\beta)] = 0,\ \forall\beta \in R_\beta \subset \mathbb{R}$.

Here R_γ and R_β are some (possibly discrete) sets of parameter values. Then solution probability amplitudes are invariant under A; that is, for any $x \in \{0,1\}^n$, for all p we have

$$\langle x|\vec{\beta},\vec{\gamma}\rangle_p = \langle a(x)|\vec{\beta},\vec{\gamma}\rangle_p\,, \qquad \forall\vec{\gamma} \in R_\gamma^p, \vec{\beta} \in R_\beta^p, \tag{9}$$

and hence, moreover, $\mathsf{Pr}_p(x) = \mathsf{Pr}_p(a(x))$.

Proof. Observe that since A is unitary, conjugating *i)* by A gives $A^\dagger[A, U_C(\gamma)]A = [A^\dagger, U_C(\gamma)] = 0$ for all $\gamma \in R_\gamma$, and similarly *ii)* implies $[A^\dagger, U_B(\beta)] = 0$ for all $\beta \in R_\beta$. Computing QAOA amplitudes by using the assumptions and $A^\dagger\,|s\rangle = |s\rangle$ gives

$$\begin{aligned}\langle a(x)|\vec{\beta},\vec{\gamma}\rangle &= \langle x|\, A^\dagger U_B(\beta_p)U_C(\gamma_p)\ldots U_B(\beta_1)U_C(\gamma_1)\,|s\rangle \\ &= \langle x|\, U_B(\beta_p)U_C(\gamma_p)\ldots U_B(\beta_1)U_C(\gamma_1)A^\dagger\,|s\rangle\ = \langle x|\vec{\beta},\vec{\gamma}\rangle\end{aligned}$$

for any $\gamma_i \in R_\gamma$, $\beta_i \in R_\beta$, $i = 1,\ldots,p$, as claimed. Taking the absolute value squared of each side gives the probability result. □

The conditions of Lemma 1 are slightly more general than those of Theorem 1; in particular, the conditions $[A, C] = 0$ and $[A, B] = 0$ of the theorem imply $[A, U_C(\gamma)] = 0$ and $[A, U_B(\beta)] = 0$ for all $\gamma, \beta \in \mathbb{R}$. Using the lemma we now give the proof of Theorem 1.

Proof of Theorem 1. We prove the more general second case, which suffices to prove the first. First observe that the conditions *i)* and *ii)* of the theorem are sufficient to satisfy those of Lemma 1; this is easily seen by expanding the matrix exponential as a power series and using the linearity of the commutator. Next observe, trivially, that composing any two permutations gives a permutation, and that since S_{2^n} is a group of finite order, iterating any fixed permutation eventually results in the identity. It follows that given any subset of S_{2^n}, composing elements yields a subgroup (if not the whole group). Moreover, if two symmetry operators A_k, A_l satisfy the conditions of Lemma 1, then clearly so does their product A_kA_l. Therefore, given any such set of symmetry operators $A_1, A_2, \dots,$ the conditions of Lemma 1 are satisfied for each element of the group generated by these operators, which implies the statements of the first two cases of the theorem.

For the final point of the theorem, using the unitarity of A, we have $[A, C] = 0$ if and only if $A^\dagger CA\,|x\rangle = C\,|x\rangle$ for all $x \in \{0,1\}^n$, or equivalently $c(a(x))\,|x\rangle = c(x)\,|x\rangle$, and hence true if and only if we have (i') $c(x) = c(a(x))$ for each $x \in \{0,1\}^n$. Similarly, $[A, B] = 0$ if and only if $A^\dagger BA\,|x\rangle = B\,|x\rangle$ for each $x \in \{0,1\}^n$. Using that B acts as on computational basis states as $B\,|y\rangle = \sum_{j=1}^n |y^{(j)}\rangle$ gives

$$A^\dagger BA\,|x\rangle = A^\dagger \sum_{j=1}^n |a(x)^{(j)}\rangle = \sum_{j=1}^n |a^{-1}(a(x)^{(j)})\rangle\,.$$

Therefore, for $A^\dagger BA\,|x\rangle = B\,|x\rangle$ to hold, we must have $\{x^{(j)} : j \in [n]\} = \{a^{-1}(a(x)^{(j)}) : j \in [n]\}$ as sets of size n. Applying $a(\cdot)$ to both sets gives (ii'), which proves the claim. □

Remark. *Since p is arbitrary in both Lemma 1 and Theorem 1, the results therein also apply to all intermediate states between QAOA layers.*

3.2 Reduced dynamics from symmetry and applications to classical simulation

In principle, each known QAOA symmetry allows us to reduce the size of the Hilbert space necessary to exactly simulate QAOA. If a large enough symmetry group is known, the dimension of the reduced subspace may become only polynomially large in the problem input size, in which case efficient classical simulation may be possible, assuming one can efficiently compute the necessary matrix elements. In this section we formalize this observation.

We first recall some additional basic group theory beyond that of Sec. 2; see [33] for a detailed overview. Let $\mathbb{B} := \{0,1\}^n$ denote the Boolean cube that is identified with the set of computational basis states $\{|x\rangle : x \in \mathbb{B}\}$, and consider a QAOA symmetry group $\mathcal{A} \subset S_{2^n}$ as in Theorem 1 (i.e., for some fixed problem and objective Hamiltonian C). Recall that the action of $\mathcal{A}$ induces equivalence classes on $\mathbb{B}$, where each class includes all elements $x \in \mathbb{B}$ belonging to the same orbit, denoted $\mathcal{A} \cdot x$ for a particular x. The quotient $\mathbb{B}/\mathcal{A}$ gives the set of equivalence classes, called coinvariants of $\mathcal{A}$, with $|\mathbb{B}/\mathcal{A}|$ the number of disjoint orbits. Invariants, on the other hand, are fixed points, defined for each $A \in \mathcal{A}$ as $\mathbb{B}^A := \{x \in \mathbb{B} : A \cdot x = x\}$. Similarly, for each $x \in \mathbb{B}$ its stabilizer subgroup is $\mathcal{A}_x := \{A \in \mathcal{A} : A \cdot x = x\}$.

We now state our result that with symmetry QAOA dynamics can be reduced to a Hilbert space of dimension $|\mathbb{B}/\mathcal{A}|$, which can be expressed as the average number of $x \in \mathbb{B}$ fixed by each $A \in \mathcal{A}$, or equivalently the average size of the stabilizer subgroups.

Theorem 2. *Consider a QAOA symmetry group $\mathcal{A} \subset S_{2^n}$ satisfying the conditions of Theorem 1. Then QAOA dynamics reduce to a subspace of dimension $|\mathbb{B}/\mathcal{A}|$ satisfying*

$$|\mathbb{B}/\mathcal{A}| = \frac{1}{|\mathcal{A}|}\sum_{A\in\mathcal{A}} |\mathbb{B}^A| = \frac{1}{|\mathcal{A}|}\sum_{x\in\mathbb{B}} |\mathcal{A}_x| = \sum_{x\in\mathbb{B}} \frac{1}{|\mathcal{A}\cdot x|}. \tag{10}$$

Here the three equalities of (10) are stated for convenience, since a particular form may be easier to apply to a particular symmetry group. We consider several examples below.

Proof. We again make use of the isomorphism between the Boolean cube and computational basis states. From Lemma 1, all basis states within the same orbit $\mathcal{A}\cdot|x\rangle := \{A|x\rangle : A \in \mathcal{A}\}$ have the same probability amplitude at all steps of the QAOA algorithm for all possible algorithm parameters, and hence it suffices to track a single amplitude for each orbit. For the cardinality, since $|\mathcal{A}|, |\mathbb{B}|$ are finite, the first equality of (10) is Burnside's lemma [33], the second equality follows observing $\sum_{A\in\mathcal{A}} |\mathbb{B}^A| = |\{(A, x) \in \mathcal{A} \times \mathbb{B} : A \cdot x = x\}| = \sum_{x\in\mathbb{B}} |\mathcal{A}_x|$, and the final equality follows observing $|\mathcal{A}_x| = |\mathcal{A}|/|\mathcal{A} \cdot x|$ for each $x \in \mathbb{B}$ from Lagrange's theorem and the orbit-stabilizer theorem. □

For example, consider a fixed $x \in \mathbb{B}$. If $\mathcal{A} \cdot x = \mathbb{B}$, that is, the orbit of x gives all bitstrings, then the third equality of (10) implies $|\mathbb{B}/\mathcal{A}| = |\mathbb{B}|/|\mathbb{B}| = 1$, and so QAOA reduces to one-dimensional (i.e., trivial) dynamics. Indeed, in this case $\mathcal{A} = S_{2^n}$, which implies that the objective function (objective Hamiltonian) is constant. If instead the objective function depends only on the Hamming weight so that $\mathcal{A} = S_n$, the group of qubit permutations, then clearly the orbit of each x contains all strings of the same Hamming weight, and so $|\mathbb{B}/\mathcal{A}| = n + 1$. Clearly, "too much" symmetry may correspond to easier instances for classical algorithms or even make the problem trivial to solve, as these examples illustrate. Conversely, for problems without any (known) classical symmetries, we have the trivial symmetry group $\mathcal{A} = \{I\}$, and hence $|\mathbb{B}/\mathcal{A}| = 2^n$; in other words, there is no implied reduction in the required Hilbert space size. We consider several less-trivial examples related to MaxCut below.

As implied by the first two examples above, for problems with sufficient classical symmetry the dimension of the effective Hilbert space may be reduced from 2^n to merely a polynomial in n. We emphasize, however, that such a reduction does not necessarily entail classical simulability of the corresponding QAOA circuit.[3] In order for such symmetry reductions to imply efficient classical simulaton, one must know or be able to compute various specific details, such as the relevant matrix elements.[4] While this is often the case for toy problems of interest (for example, the Hamming weight ramp problem [39]), we leave as a topic of future work a detailed application of Theorem 2 towards deriving more rigorous conditions and settings allowing for efficient classical simulation of QAOA circuits.

Indeed, such investigations may benefit from more technical generalizations of Theorems 1 and 2. We emphasize that the statements and proofs of both theorems depend in an essential way on the particular choice of QAOA initial state as the uniform superposition $|s\rangle = |+\rangle^{\otimes n}$, which guarantees that all of the computational basis states in each symmetry group orbit always have the same probability amplitudes.[5] Both theorems may be generalized to arbitrary QAOA initial states, but at the expense of a more complicated presentation in terms of group representation theory beyond the scope of this study. Similar considerations apply for extending our results to different mixing operators beyond the transverse-field QAOA mixing Hamiltonian.

We conclude the section with three illustrative examples.

Example: QAOA for problems with $\mathbb{Z}_2$ symmetry Consider an objective function that is invariant under the $\mathbb{Z}_2$ symmetry $A = X_1X_2 \ldots X_n$ of flipping all bits simultaneously. We recall,

[3]General efficient classical simulation of even depth 1 QAOA circuits is impossible under standard computational complexity assumptions [38] (including, of course, worst-case problem instances with little or no symmetry).

[4]A useful example for comparison is Grover's quantum algorithm for unstructured search; while it is well known that similar considerations reduce the quantum dynamics to a 2-dimensional Hilbert space, this insight does not allow one to obtain the solution (without the quantum computer). The algorithm's quadratic speedup over classical computers is typically proven by using this reduction. A similar observation is made for a generalization of QAOA with mixing operator derived from Grover's algorithm in [22].

[5]More precisely, this particular state projects entirely to the one-dimension representative (symmetric superposition) corresponding to each element of the quotient space $\mathbb{B}/\mathcal{A}$.

for example, that MaxCut always has this symmetry; other problems may have this symmetry by construction or only for particular instances. For the symmetry group $\{I, A\}$ clearly we have $\mathbb{B}^I = \mathbb{B}$ and $\mathbb{B}^A = \emptyset$, and so the first equation of Theorem 2 gives $|\mathbb{B}/\mathcal{A}| = \frac{1}{2}(2^n + 0) = 2^{n-1}$; that is, as one may expect, the Hilbert space dimension may be reduced in half. In this case it suffices to consider the set of basis states $\{\frac{1}{\sqrt{2}}(|x\rangle + A\,|x\rangle)\}$ given by the symmetric linear combinations of the two computational basis states in each orbit.

Example: QAOA for MaxCut on general graphs For an arbitrary MaxCut instance on a graph G with automorphism group $Aut(G)$, Theorem 2 implies that the QAOA dynamics reduce to a subspace of dimension $|\mathbb{B}/Aut(G)|$, with basis states given by symmetric sums of the states in each orbit $\mathbb{B}^{Aut(G)}$, or similarly of dimension $|\mathbb{B}/\mathcal{A}|$ if only a subgroup $\mathcal{A} \subset Aut(G)$ is known.

Further consider that MaxCut also satisfies the $\mathbb{Z}_2$-symmetry $A = X_1X_2\dots X_n$ of the preceding example. For each $A_j \in \mathcal{A} \subset Aut(G)$ one can easily see that $AA_j = A_jA$, and hence $\mathcal{A}\cup\{A\}$ generates the subgroup $\mathcal{A}' = \mathcal{A}\cup A\mathcal{A} = \{A_1 = I, A_2, \dots, A_{|\mathcal{A}|}, A, AA_2, \dots, AA_{|\mathcal{A}|}\} \subset S_{2^n}$ of order $2|\mathcal{A}|$, which reduces the effective Hilbert space dimension by a further factor of 2.

Example: QAOA for MaxCut on complete graphs Consider MaxCut on the complete graph $G = K_n$ with n vertices and $\binom{n}{2}$ edges. Clearly, for complete graphs the MaxCut objective function $f(x)$ is invariant under permutations of the vertices, corresponding to S_n permutation symmetry of the qubit encoding, and the dynamics can again be reduced to an $(n+1)$-dimensional subspace. Indeed, a simple calculation shows that the objective function may be written as $f(x) = f(|x|) = n|x| - |x|^2$ where $|\cdot|$ is the Hamming weight. Let D denote the Hamiltonian representation of the Hamming weight function, that is, $D|x\rangle = |x||x\rangle$ for all $x \in \mathbb{B}$, which is given by the 1-local Hamiltonian $D = \frac{n}{2}I - \sum_{j=1}^n Z_j$. Then the objective Hamiltonian may be written $C = nD - D^2$, which acts diagonally on the Hamming basis of states $|d\rangle = \frac{1}{\sqrt{\binom{n}{d}}}\sum_{x:|x|=d}|x\rangle$, $d = 0, 1, \dots, n$. The initial state $|s\rangle$ is easily expressed in this basis as

$$|s\rangle = \sum_{d=0}^{n}\sqrt{\frac{\binom{n}{d}}{2^n}}|d\rangle,$$

and a straightforward calculation shows the action of the mixing Hamiltonian to be

$$B|d\rangle = \sqrt{d(n-d+1)}|d-1\rangle + \sqrt{(d+1)(n-d)}|d+1\rangle,$$

with the understanding that $|n+1\rangle = |0-1\rangle = 0$.

Hence, the classical simulation of QAOA for MaxCut on complete graphs requires dealing with matrices of size $(n+1)\times(n+1)$, which can be accomplished in poly(n) time for QAOA depth up to p =poly(n), modulo accuracy considerations. Note that we may again apply the $\mathbb{Z}_2$-symmetry above to further reduce the subspace dimension by an additional factor of 2 by the identification of pairs of states $|d\rangle$ and $|\bar{d}\rangle := |n-d\rangle$ (more precisely, to dimension $(n+1)/2$ for n odd or $n/2+1$ for n even).

3.3 On the difficulty of obtaining the symmetry group

For some problems efficiently determining all the classical symmetries in general may not be possible. Nevertheless, our results above apply for any known group of classical symmetries, which may be obtained through a number of possible approaches. For determining graph symmetries, efficient algorithms or fast heuristic methods exist for many practically important classes of problem graphs. Here we briefly review some such approaches as well as the complexity of obtaining the full graph automorphism group. Similar considerations apply more generally.

Given an arbitrary graph (e.g., an instance of MaxCut), an important open problem is whether there exists an efficient classical algorithm for determining its automorphism group (in general), with the best algorithm known resulting from Babai's recent breakthrough discovery of a quasi-polynomial time algorithm for the graph isomorphism problem [40]. Since determining a generating set of the graph automorphism group is polynomial time equivalent to graph isomorphism [41], this leads to a quasi-polynomial upper bound to the time complexity of determining the full group of automorphisms. Nevertheless, polynomial-time algorithms exist for many important classes of graphs, including those with bounded degree [25], bounded genus [42, 43], or bounded tree-width [44] or characterized by particular forbidden minors [45, 46], among others [47, 48].

Note that while the full graph automorphism group $Aut(G)$ can contain up to $n!$ elements, for the applications presented in Sec. 4 it suffices to construct its generating set. Since for an arbitrary group with m elements there always exists a generating set of size $\log m$ [41], the automorphism group can always be represented efficiently (though, as indicated, finding this representation may be challenging). These observations imply that the developed approach can be efficiently applied to a variety of important problems. For example, consider MaxCut on three-regular graphs, which is NP-hard to approximate beyond 0.997 [49]; in this case, the generating set of automorphisms can be constructed efficiently, and hence the techniques of this section and the following may be applied.

We emphasize that in practice one often can determine the automorphism group heuristically for problems of substantial size, although worst-case guarantees may be lacking. Indeed, state-of-the-art heuristics are able to compute the automorphism group generators for random graphs with up to tens of thousands of nodes in less then a second [48]. Many such heuristics have been proposed to accelerate computation of automorphisms, including `nauty` [50, 47] (which we employ for the application of Sec. 4), `Traces` [48], `saucy` [51], `Bliss` [52], and `conauto` [53], with accelerated running time resulting from the application of the Weisfeiler–Lehman test and search tree pruning heuristics to generate canonical labeling of nodes, among other techniques.

4 Using symmetry to predict QAOA performance

We now present an example application that extends and applies the ideas introduced in Section 3 to study the empirical relationship between QAOA performance and symmetries of the objective function in a particular setting. We use a machine learning approach employing features computed from the group of automorphisms of a graph, as well as other symmetry considerations, to predict the minimum depth required for QAOA to achieve a fixed approximation ratio for the MaxCut problem on it. We focus on studying the connection between symmetry properties of the particular problem instance and QAOA performance. To this end, we train two support vector models that use a set of problem symmetry measures. We restrict QAOA parameters to a specific class of schedules, defined and motivated in Sec. 4.1, to circumvent any computational difficulties related to parameter optimization. We use symmetry measures corresponding both to exact symmetries as defined in Theorem 1, and to approximate symmetries we describe in Sec. 4.2 below.

Here we use $p_{\min} = p_{\min}(C, B, \epsilon)$ defined as "the smallest p with which QAOA restricted to linear schedules achieves an approximation ratio of $r = (1-\epsilon)$ for a given instance" as the metric for quantifying QAOA performance, with the approximation ratio r defined in Eq. (4). Thus defined, $p_{\min}$ exists for any problem instance, since from the adiabatic limit QAOA with $p \to \infty$ can solve any optimization problem exactly [2], and this remains true in our chosen setting of linear schedules. As the running time of QAOA grows linearly with its depth p, and since in the NISQ era the achievable depth p is limited by the error rates and coherence time of the quantum hardware, $p_{\min}$ gives a meaningful measure of QAOA performance for our purposes. Note that here we approach QAOA as a *heuristic* applied to a particular problem *instance* or set of instances and not necessarily as an approximation algorithm with performance guarantees on a given problem class.

4.1 Smooth schedules: a practical approach to QAOA

One challenge in both the analysis and the implementation of QAOA is the need to select sufficiently good algorithm parameters $\vec{\beta}, \vec{\gamma}$. Optimizing parameters for QAOA (e.g., variationally) appear to be difficult in practice [54, 10, 55], presenting two challenges in particular for numerical experiments. First, performing such numerical studies requires solving the parameter optimization problem for a large number of instances, which is especially difficult in practice as the number of parameters becomes significant. Second, even if such data were collected, it would likely not be representative of real-world QAOA performance, since finding optimal parameters for every instance may not be a practical way of using QAOA as the depth p becomes large.

Instead, more practical approaches to employing QAOA have been proposed. One approach, explored in [55, 56, 57], among others, involves choosing a class of instances and training a machine learning model to quickly produce high-quality QAOA parameters for that class. While powerful, however, this approach resolves only the second challenge, because in order to train the model, one still has to find near-optimal parameters for a sufficiently large dataset representative of the chosen class of instances. Another approach, which is used in this section, is to specify a class of *parameter schedules* and then fit a schedule of this class to a particular problem instance, such that the number of free parameters is significantly decreased, and the difficulty of the parameter optimization problem significantly reduced. There are various reasonable choices for the class of schedules; in this work, we focus on *smooth* schedules, that is, schedules where the change from one component to the next in $\vec{\beta} \in \mathbb{R}^p$ and $\vec{\gamma} \in \mathbb{R}^p$ is small relative to the difference between minimum and maximum values of β and γ. Similar approaches to parameter setting for QAOA have been considered in [11, 12, 58], among others. Our particular method is inspired by the reparameterization developed in [11]. Specifically, we focus on *linear schedules*, that is, schedules where for a given depth p the component parameters β_j, γ_j change linearly with each step j:

$$\beta_j = a_\beta j + b_\beta, \qquad \gamma_j = a_\gamma j + b_\gamma, \tag{11}$$

for $j = 1, 2, \ldots, p$, where $a_\beta, b_\beta, a_\gamma, b_\gamma$ are the remaining free parameters to be optimized. An example of such a parameter schedule is presented in Fig. 2.

Restricting QAOA parameters to a particular class of schedules clearly yields a tradeoff between the difficulty of parameter optimization and the algorithm's performance, relative to unrestricted parameters. Fitting a particular class of schedules is much easier than finding an optimal general schedule. For example, in general for $p = 10$ finding optimal parameters requires optimizing $2 \times p = 20$ parameters, whereas fitting a linear schedule requires optimizing only over 4 parameters (slope and intercept for both β and γ). We also observe this in practice, with multistart methods [54] capable of finding optimal linear schedules quickly and reliably. At the same time, this restriction may increase the minimum depth needed to achieve the desired approximation ratio (i.e., $p_{\min}(C, B, \epsilon) > p_{\min}(C, B, \epsilon, \text{linear schedule})$). Note that in adiabatic limit $p \to \infty$ linear schedules are sufficient, so such minimum depth $p_{\min}$ always exists. For the remainder of the section we will consider QAOA with linear schedules only, i.e. $p_{\min} := p_{\min}(C, B, \epsilon, \text{linear schedule})$.

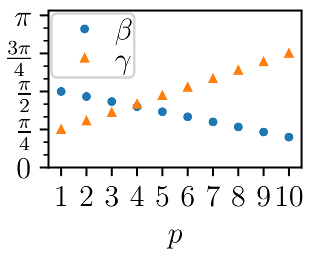


Figure 2: Example of a linear schedule for $p = 10$. Here β_j changes linearly from $\pi/2$ at the first QAOA step to $\pi/5$ at the last QAOA step, and γ_j changes from $\pi/4$ to $3\pi/4$.

4.2 Exact and approximate symmetry measures

Quantitative analysis of graph symmetry is a challenging problem. Over the years, multiple symmetry indices have been introduced to attempt to tackle this problem in different contexts. Some notable examples are graph entropy [59, 60], index of symmetry [61], network redundancy [62], and normalized network redundancy [63]. However, we observe that all of these metrics fail to capture the finer differences between instances required for achieving high predictive power. Therefore we propose constructing a data-driven symmetry index that is tailored to our particular application. We construct this index by noting that all of the metrics in [59, 60, 61, 62, 63] are a combination of the number of vertex orbits, the size of orbits, the size of group of automorphisms, the number of nodes of a graph, and the graph entropy. Instead of combining them in an index constructed from some prior intuition, we use them as features in a machine learning model. The resulting trained model serves as a symmetry measure of a graph.

First, we use the following *exact* symmetry measures:

1. Logarithm of the size of the group of automorphisms of the graph: $\log |Aut(G)|$
2. Number of vertex orbits of the graph: $|O(G)|$
3. Entropy of the graph: $I(G) = \frac{1}{n}\sum_i |A_i| \log |A_i|$, where $|A_i|$ is the size of ith vertex orbit and n is the number of nodes in G

Second, we consider the following *approximate* symmetry measures to further improve the performance of our model. Let G'_e denote the graph constructed from G by removing edge e. Intuitively, if G'_e has an automorphism σ, then σ is said to be an *approximate* automorphism of G. Therefore for each measure of exact symmetry we can introduce a corresponding measure of *approximate symmetry* of G defined as the average value of the exact measure on G'_e when averaged over all edges e. Similarly, we consider G''_{e_1,e_2}, constructed by removing the two edges e_1, e_2 from G, and use the symmetry measure of G''_{e_1,e_2} averaged across all pairs of edges (e_1, e_2). For example, the first metric, the logarithm of order of the graph automorphism group, induces the following two approximate metrics: the average size of a group of automorphisms for graphs constructed from G by removing one edge ($\frac{1}{h'}\sum_{i=1}^{h'} \log |Aut(G'_i)|$, $h' = \binom{|E|}{1} = |E|$), or two edges ($\frac{1}{h''}\sum_{i,j} \log |Aut(G''_{i,j})|$, $h'' = \binom{|E|}{2}$). Hence, approximate symmetry measures add another six metrics to the list of features, and in addition we include the number of vertices explicitly, for a total of ten features.

Note that if a generating set of the group of graph automorphisms is given, the number of vertex orbits of the graph and their sizes can be computed in polynomial time [41]. This means that the proposed features can be computed in polynomial time from the generating set of the group of automorphisms of the graph and, for approximate features, the induced graphs. Recalling the discussion of the complexity of obtaining such generating sets in Sec. 3.3, here we simply reiterate that for many practically interesting problem classes (such as graphs with bounded degree) the full graph symmetry groups and complete set of proposed features can be computed efficiently. Additionally, useful bounds on the magnitudes of these symmetry measures are often available; in particular, upper bounds for the number of graph automorphisms are shown in terms of the vertex degrees (e.g., maximum and average degree of the graph and its subgraphs) in [64], and in terms of the the graph diameter and number of nodes in [65].

4.3 Learning the relationship between symmetry and performance

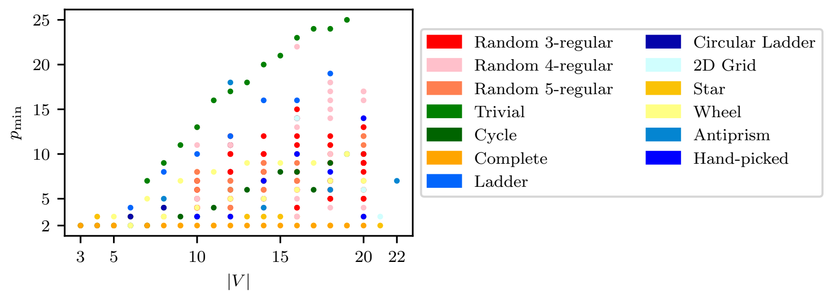
To numerically study the relationship between the symmetry of the objective function and QAOA performance, we use the MaxCut problem on a dataset of 294 graphs, selected as described in Table 1. We set the target approximation ratio $r = 0.95$, with r defined by Eq. 4.

Name	# graphs	$\min \lvert V\rvert$	$\max \lvert V\rvert$	$\min \lvert E\rvert$	$\max \lvert E\rvert$	$\min\log \lvert Aut(G)\rvert$	$\max\log \lvert Aut(G)\rvert$	$\min \frac{1}{h'}\sum_{i=1}^{h'}\log \lvert Aut(G'_i)\rvert$	$\max \frac{1}{h'}\sum_{i=1}^{h'}\log \lvert Aut(G'_i)\rvert$	$\min \frac{1}{h''}\sum_{i,j}\log \lvert Aut(G''_{i,j})\rvert$	$\max \frac{1}{h''}\sum_{i,j}\log \lvert Aut(G''_{i,j})\rvert$	$\min \lvert O(G)\rvert$	$\max \lvert O(G)\rvert$	$\min \frac{1}{h'}\sum_{i=1}^{h'} \lvert O(G'_i)\rvert$	$\max \frac{1}{h'}\sum_{i=1}^{h'} \lvert O(G'_i)\rvert$	$\min \frac{1}{h''}\sum_{i,j} \lvert O(G''_{i,j})\rvert$	$\max \frac{1}{h''}\sum_{i,j} \lvert O(G''_{i,j})\rvert$	$\min \lvert I(G)\rvert$	$\max \lvert I(G)\rvert$	$\min \frac{1}{h'}\sum_{i=1}^{h'} I(G'_i)$	$\max \frac{1}{h'}\sum_{i=1}^{h'} I(G'_i)$	$\frac{1}{h''}\sum_{i,j} I(G''_{i,j})$	$\frac{1}{h''}\sum_{i,j} I(G''_{i,j})$
Antiprism	7	6	22	12	44	2.8	3.9	0.7	1.4	0.4	1.0	1.0	1.0	3.0	11.5	3.5	16.7	1.8	3.1	0.6	0.7	0.3	0.6
Circular Ladder	7	6	18	9	27	2.5	3.9	0.9	1.4	0.5	1.0	1.0	1.0	3.0	8.3	4.2	12.4	1.8	2.9	0.7	1.0	0.5	0.8
Complete	18	3	20	3	190	1.8	42.3	0.7	37.1	0.7	33.0	1.0	1.0	2.0	2.0	2.0	2.7	1.1	3.0	0.5	2.7	0.5	2.5
Cycle	18	3	20	3	20	1.8	3.7	0.7	0.7	0.7	1.3	1.0	1.0	2.0	10.0	2.0	10.3	1.1	3.0	0.5	0.7	0.5	0.7
2D Grid	4	16	21	24	42	1.4	4.4	0.2	1.4	0.1	0.6	1.0	6.0	7.5	16.0	13.5	17.0	1.2	3.0	0.2	1.1	0.1	0.5
Hand-picked	7	10	20	15	30	4.6	5.8	1.4	2.8	0.7	1.5	1.0	1.0	3.0	7.0	5.2	10.9	2.3	3.0	0.9	1.7	0.6	1.1
Ladder	7	6	18	7	25	1.4	1.4	0.4	0.9	0.4	1.0	2.0	5.0	3.9	14.4	3.5	15.8	1.2	1.4	0.3	0.6	0.2	0.6
Random 3-reg	60	10	20	15	30	0.0	3.5	0.0	1.8	0.0	1.4	3.0	20.0	7.3	20.0	7.9	20.0	0.0	1.4	0.0	0.5	0.0	0.4
Random 4-reg	59	10	20	20	40	0.0	2.5	0.0	1.3	0.0	0.8	4.0	20.0	7.1	20.0	8.3	20.0	0.0	1.0	0.0	0.4	0.0	0.2
Random 5-reg	60	10	20	25	50	0.0	1.4	0.0	0.7	0.0	0.4	4.0	20.0	8.6	20.0	9.2	20.0	0.0	1.0	0.0	0.2	0.0	0.1
Star	18	4	21	3	20	1.8	42.3	0.7	39.3	1.4	37.1	2.0	2.0	3.0	3.0	2.0	3.0	0.8	2.9	0.3	2.7	0.6	2.5
Trivial	13	7	19	6	18	0.0	0.0	0.8	1.1	1.6	1.7	7.0	19.0	4.7	14.6	3.9	12.7	0.0	0.0	0.3	0.5	0.5	0.7
Wheel	16	5	20	8	38	2.1	3.6	0.7	1.0	0.5	0.9	2.0	2.0	3.0	11.0	3.3	15.4	1.1	2.8	0.5	0.6	0.3	0.5
Total	294	3	22	3	190	0.0	42.3	0.0	39.3	0.0	37.1	1.0	20.0	2.0	20.0	2.0	20.0	0.0	3.1	0.0	2.7	0.0	2.5

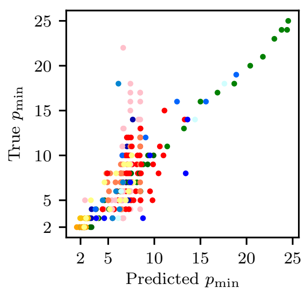
Table 1: Description of the dataset. ”Trivial” is graph with a trivial group of automorphisms. ”Hand-picked” includes various textbook graphs with large groups of automorphisms, e.g. Peterson and Heawood graphs. We make the full dataset available online [27].

Therefore the metric to be predicted is given by $p_{\min}$ = "the smallest p with which QAOA restricted to linear schedules achieves an approximation ratio of 0.95 for MaxCut on a given instance." While this choice of target approximation ratio is somewhat arbitrary, it is inspired by classical hardness results for MaxCut, which is NP-hard to approximate better than $\frac{16}{17} \approx 0.941$ in the worst-case [66]. Under the Unique Games conjecture, a weaker assumption than P≠NP, the 0.87856 approximation ratio achieved by the Goemans–Williamson algorithm is optimal [67, 68]. Indeed, for similar complexity reasons we do not expect quantum computers to efficiently solve NP-hard problems such as MaxCut. In any case, these complexity results become meaningful only as problem sizes becomes large, which makes comparison with results based on fixed size training and prediction sets difficult. Hence, we select an approximation ratio of 0.95 as a reasonable target that reflects desirable QAOA performance and the possibility of quantum advantage beyond the finite dataset of this study.

We use `nauty` [48] to compute the features described in Sec. 4.2. We compute $p_{\min}$ for each problem in the dataset by considering iteratively larger depths p, starting with $p = 2$ and optimizing the linear parameter schedule as defined in Sec. 4.1. until the target approximation ratio is achieved. We use COBYLA [69, 70] implemented in the SciPy [71] package as a local optimizer in the libEnsemble [72] implementation of APOSMM [73, 74]. We use NetworkX [75] for graph operations and GNU-Parallel for large-scale experiments [76]. The code is available online at [26].



(a) $p_{\min}$ as a function of the number of nodes. $p_{\min}$ grows the fastest for the least symmetric graphs ("Trivial") and the slowest for the most symmetric (complete and star graphs).



(b) True $p_{\min}$ and $p_{\min}$ predicted by support vector regression for both training and testing data. Median absolute error on the test set is 1.37.

Figure 3: Using problem symmetry to predict $p_{\min}$.

4.3.1 Initial observations from the training set

We begin by discussing some notable properties of the training set. First, we observe that all the chosen features correlate with $p_{\min}$, with Pearson correlation coefficients between 0.3 and 0.5 in absolute value, where the Pearson correlation coefficient is defined as $\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{x})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$, and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean. The empirically observed correlation coefficients are presented in Table 2. The large value of the correlation coefficients indicates that the chosen features are meaningful and have predictive power. We note that measures of approximate symmetry have correlation coefficients similar to the corresponding exact symmetries, as expected.

Second, we observe that the hardest problem instances for QAOA (with respect to $p_{\min}$) in our dataset are found to be the ones corresponding to the least symmetric graphs, and the easiest are the ones corresponding to the most symmetric graphs, as one may expect from our results of Sec. 3. Fig. 3a shows the scaling of $p_{\min}$ with problem size, with the hardest instances seen to be

Feature	Pearson r	p-value
$\log Aut(G)$	-0.36397	0.00000
$\frac{1}{h'}\sum_{i=1}^{h'} \log Aut(G'_i)$	-0.32851	0.00000
$\frac{1}{h''}\sum_{i,j} \log Aut(G''_{i,j})$	-0.30854	0.00000
$\lvert V\rvert$	+0.37075	0.00000
$\lvert O(G)\rvert$	+0.39314	0.00000
$\frac{1}{h'}\sum_{i=1}^{h'} \lvert O(G_i)\rvert$	+0.44326	0.00000
$\frac{1}{h''}\sum_{i,j} \lvert O(G''_{i,j})\rvert$	+0.43035	0.00000
$I(G)$	-0.32295	0.00000
$\frac{1}{h'}\sum_{i=1}^{h'} I(G_i)$	-0.36913	0.00000
$\frac{1}{h''}\sum_{i,j} I(G''_{i,j})$	-0.34254	0.00000

Table 2: Pearson correlation coefficient r and p-value for the test of non-correlation of features with $p_{\min}$. See Section 4.2 for the definitions of the features.

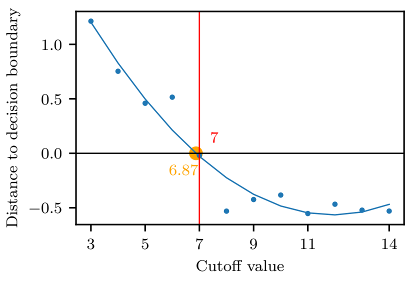
those for which $p_{\min}$ grows relatively quickly, and the easiest those for which $p_{\min}$ grows slowly or does not grow at all. For problem instances with almost no symmetry (graphs with trivial automorphism group), $p_{\min}$ is observed to grow the fastest, and for instances with the most symmetry (e.g., complete graphs) $p_{\min}$ grows the slowest. These results provide further evidence of the connection between symmetry and QAOA performance, though we emphasize that more work is needed to better understand to what degree these observations hold asymptotically, or, more generally, in the setting of real-world quantum devices.

4.3.2 Machine learning approaches to predicting QAOA performance

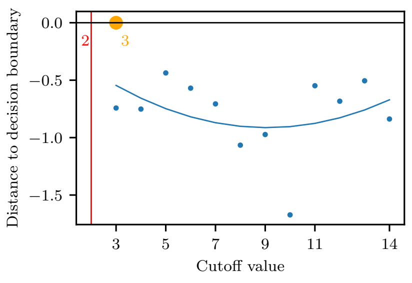
To quantify the intuition arising from Fig. 3a, we use the metrics discussed in Section 4.2 as features for a machine learning model that we train to predict $p_{\min}$. We reserve 30% on the dataset as the testing set and use the rest for training the model, with the training set further partitioned for cross-validation as specified below. We approach the problem in two ways. In the first approach, we treat the task of predicting $p_{\min}$ as a regression problem. In the second approach, we use an ensemble of classifiers to predict $p_{\min}$. Below we discuss both approaches and the tradeoffs they introduce. We choose median absolute error as our target metric. This choice is motivated by the physical meaning of $p_{\min}$, namely, the minimum depth required to achieve 0.95 approximation ratio by using QAOA with linear schedules. Since $p_{\min}$ is an integer, absolute error is an easily interpretable metric. Moreover, an absolute error of one or close to one is tolerable for most applications (e.g., if we want to use the trained model to predict the value of depth p to use in QAOA or if we want to establish whether a given problem requires depth beyond the capabilities of target quantum hardware).

First, we approach the problem of predicting QAOA performance (i.e., predicting the number $p_{\min}$ for a given problem instance) directly as a regression problem. We use support vector regression (SVR) [77, 78, 79] with the radial basis function kernel. We use 5-fold cross-validation stratified by graph class for hyperparameter optimization. We achieve 0.73 median absolute error on the training set and 1.37 on the testing set (i.e., on the instances previously unseen by the model). We observe a correlation between the predicted $p_{\min}$ and true $p_{\min}$ with a Pearson correlation coefficient of 0.71 on the test set. The results are visualized in Fig. 3b.

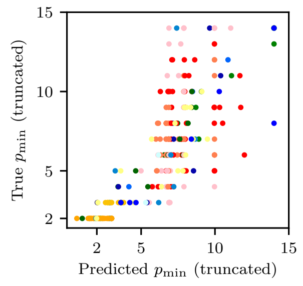
The second approach we use for predicting $p_{\min}$ is training an ensemble of classifiers. We simulate a realistic scenario of limited circuit depth by grouping all instances with $p_{\min} \geq 15$ into one class corresponding to "depth beyond the capabilities of the target hardware." Since $p_{\min}$ is a discrete value (an integer), classification is a natural choice. An issue with using a classifier directly is that assigning each value of $p_{\min}$ to be a class discards the information that two consecutive integers are more similar than two integers that are far apart. To address this issue, we instead train an ensemble of binary classifiers, where each classifier answers the question "Is the value of $p_{\min}$ smaller than the specified cutoff?" Using an ensemble of "cutoff" classifiers is a standard approach for ordinal regression [80, 81]. Each classifier is a support vector machine classifier [78, 79]. We use the radial basis function kernel and the optimal hyperparameters found by cross-validation for the support vector regression.



(a) The blue dots are distance to the decision boundary for each classifier. The blue line is the least squares quadratic fit. The yellow dot is the predicted $p_{\min}$, and the red vertical line is the true $p_{\min}$.



(b) If the least squares quadratic fit does not intersect the x-axis within the bounds, we choose either the minimum or maximum $p_{\min}$ based on the majority vote of classifiers. Here all classifiers put the point in the first class ("$p_{\min}$ is smaller than the cutoff").



(c) True $p_{\min}$ and $p_{\min}$ predicted by an ensemble of support vector machines for both training and testing data. Median absolute error on the test set is 1.35. See Fig. 3 for the color legend.

Figure 4: Using an ensemble of SVM classifiers to predict $p_{\min}$.

The ensemble of "cutoff" binary classifiers is used to predict $p_{\min}$ in the following way. For a new point, we compute the distance to the decision boundary for each classifier. By convention, if the distance is positive, then the point belongs to the first class ($p_{\min}$ is smaller than the cutoff); otherwise it belongs to the second class ($p_{\min}$ is greater than or equal to the cutoff). Therefore the minimum cutoff value at which the binary classifiers are assigning the point to the second class (i.e., at which the distance to the decision boundary changes sign) is the $p_{\min}$ for this point. To make the distances to the decision boundary directly comparable for all classifiers, we standardize them in the following way. For each classifier we compute distances to the decision boundary on the entire training set. If the standard deviation of these distances is σ, we let the standardized distance to the decision boundary be $d_z = d/\sigma$, where d is the distance to the decision boundary for the new point. To predict $p_{\min}$ for the point, we fit a least squares quadratic function to the standardized distances to the decision boundary as a function of the cutoff value for all classifiers, and we use its intersection with the x-axis as the predicted value. This process is visualized in Figs. 4a and 4b. We achieve a median absolute error of 1.35 on the test set.

The two approaches considered for predicting $p_{\min}$ demonstrate similarly good performance, as observed by comparing Figures 3b and 4c. As predicted by the high correlation between an exact symmetry feature and the corresponding approximate symmetry features, we observe only a small decrease in performance ($\approx 10\%$ increase in test error) if we discard the approximate symmetry features and use only the three exact ones. From the observed high ratio of support vectors, we expect that performance can be significantly improved by introducing more datapoints (problem instances). Whereas in this study the size of the dataset was limited by the cost of classical simulation, recent advances in QAOA simulation [82, 83] and the increasing availability of quantum hardware are widely believed to enable large-scale studies of QAOA that we expect to significantly boost the performance of the proposed methods.

5 Discussion and future work

In this paper we introduced a formal connection between the dynamical symmetries of QAOA and the classical symmetries of the underlying objective function, which we employed to derive several operational results. We further demonstrated the power of this connection by training machine learning models to predict QAOA performance solely from the information about exact and approximate symmetries of the problem. As emphasized, our approach is general and applies to a wide variety of optimization problems, and may be easily extended to generalizations of QAOA beyond the transverse-field mixer and uniform superposition initial state.

An especially attractive future research direction is to apply our results towards deriving improved bounds concerning QAOA performance and conditions for (in)efficient classical simulation (i.e., to better characterize regimes of potential quantum advantage), in the spirit of the approaches of [21] and [38]. Particularly desirable is the further identification of classes of problems where symmetry leads to provable performance advantages, or limitations. In any case, we are optimistic that symmetry insights may be an important tool in the design of better quantum ansatz and algorithms for optimization, and beyond.

Another interesting future direction is to consider further generalizations of our notion of symmetry. Since in some cases QAOA may not "see the whole graph" [14], symmetries involving particular subsets of variables (subgraphs) as dictated by the structure of the cost function may be even more precise predictors of QAOA performance. Research in these directions is complicated by the fact that such symmetries may come into play only for much larger graphs, as well as other combinatoric considerations limiting our ability to perform numerical studies. Indeed, a limitation to applying our machine learning approach in practice is that for some classes of problems the metrics defined in Section 4.2 may not be computable in polynomial time in the worst case. While for problem instances with hundreds or thousands of variables off-the-shelf tools like `nauty` [48] can compute them in seconds, this approach is not scalable to graphs with hundreds of thousands or millions of nodes. Therefore an important next step is exploring methods to efficiently approximate these quantities with techniques such as network alignment [84]. Interpretability of the machine learning approach may also evolve with both the size of instances and consequently the training set size. Although the SVM/SVR approach is one of the most interpretable learning models that helps estimate the importance of features and their combinations, advanced scalable nonlinear methods such as those presented in [85] will likely be required in order to avoid overfitting of the model.

Acknowledgments

We thank Salvatore Mandrà, Gianni Mossi, Stefan Wild, Ehsan Sadrfaridpour, Justin Sybrandt, Jeffrey Larson, and Mikhail Klin for valuable discussions and assistance in the preparation of this manuscript. R.S. is grateful to KBR, the QuAIL group, and the NASA Ames Research Center for the internship opportunity that enabled this research. R.S. and I.S. were supported by funding from the Defense Advanced Research Projects Agency (DARPA). S.H., T.H, and R.S appreciate support from the NASA Ames Research Center and support from DARPA under IAA 8839, Annex 114. S.H. and T.H. were supported in part by the NASA Academic Mission Services under Contract NNA16BD14C. R.S. was supported by Laboratory Directed Research and Development (LDRD) funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357 and by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research for Quantum Computing program. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] F. Arute *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, pp. 505–510, Oct. 2019.

[2] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv:1411.4028*, 2014.

[3] S. Hadfield, Z. Wang, B. O'Gorman, E. Rieffel, D. Venturelli, and R. Biswas, "From the Quantum Approximate Optimization Algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, p. 34, Feb. 2019.

[4] F. Arute *et al.*, "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor," *arXiv:2004.04197*, 2020.

[5] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem," *arXiv:1412.6062*, 2014.

[6] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, "Quantum approximate optimization algorithm for MaxCut: A fermionic view," *Physical Review A*, vol. 97, Feb. 2018.

[7] S. Hadfield, "Quantum algorithms for scientific computing and approximate optimization," *Columbia university PhD dissertation, arXiv:1805.03265*, 2018.

[8] J. Wurtz and P. J. Love, "Bounds on MaxCut QAOA performance for p>1," *arXiv:2010.11209*, 2020.

[9] Z. Jiang, E. G. Rieffel, and Z. Wang, "Near-optimal quantum circuit for Grover's unstructured search using a transverse field," *Physical Review A*, vol. 95, June 2017.

[10] R. Shaydulin and Y. Alexeev, "Evaluating quantum approximate optimization algorithm: A case study," in *2019 Tenth International Green and Sustainable Computing Conference (IGSC)*, IEEE, Oct. 2019.

[11] 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," *arXiv:1812.01041*, 2018.

[12] G. E. Crooks, "Performance of the quantum approximate optimization algorithm on the maximum cut problem," *arXiv:1811.08419*, 2018.

[13] M. B. Hastings, "Classical and quantum bounded depth approximation algorithms," *arXiv:1905.07047*, 2019.

[14] E. Farhi, D. Gamarnik, and S. Gutmann, "The Quantum Approximate Optimization Algorithm needs to see the whole graph: A typical case," *arXiv:2004.09002*, 2020.

[15] C.-K. Chiu, J. C. Teo, A. P. Schnyder, and S. Ryu, "Classification of topological quantum matter with symmetries," *Reviews of Modern Physics*, vol. 88, no. 3, p. 035005, 2016.

[16] M. B. Hastings, "Trivial low energy states for commuting Hamiltonians, and the quantum PCP conjecture," *Quantum Info. Comput.*, vol. 13, no. 5–6, p. 393–429, 2013.

[17] L. Eldar and A. W. Harrow, "Local Hamiltonians whose ground states are hard to approximate," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, IEEE, Oct. 2017.

[18] J. Ostrowski, J. Linderoth, F. Rossi, and S. Smriglio, “Orbital branching,” *Mathematical Programming*, vol. 126, no. 1, pp. 147–178, 2011.

[19] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, “Branch-and-price: Column generation for solving huge integer programs,” *Operations research*, vol. 46, no. 3, pp. 316–329, 1998.

[20] S. Ben-David, A. M. Childs, A. Gilyén, W. Kretschmer, S. Podder, and D. Wang, “Symmetries, graph properties, and quantum speedups,” *arXiv:2006.12760*, 2020.

[21] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, “Obstacles to state preparation and variational optimization from symmetry protection,” *arXiv:1910.08980*, 2019.

[22] A. Bärtschi and S. Eidenbenz, “Grover mixers for QAOA: Shifting complexity from mixer design to state preparation,” *arXiv:2006.00354*, 2020.

[23] G. G. Guerreschi and A. Y. Matsuura, “QAOA for max-cut requires hundreds of qubits for quantum speed-up,” *Scientific Reports*, vol. 9, May 2019.

[24] M. Szegedy, “What do QAOA energies reveal about graphs?,” *arXiv:1912.12277*, 2019.

[25] E. M. Luks, “Isomorphism of graphs of bounded valence can be tested in polynomial time,” *Journal of computer and system sciences*, vol. 25, no. 1, pp. 42–65, 1982.

[26] [Online.] `https://github.com/rsln-s/Classical-symmetries-and-QAOA`.

[27] [Online.] `https://www.dropbox.com/s/cftspvdoovnzi4l/allresults.p.zip?dl=0`.

[28] S. Hadfield, “On the representation of Boolean and real functions as Hamiltonians for quantum computing,” *arXiv:1804.09130*, 2018.

[29] V. V. Vazirani, *Approximation algorithms.* Springer Science & Business Media, 2013.

[30] C. H. Papadimitriou and M. Yannakakis, “Optimization, approximation, and complexity classes,” *Journal of Computer and System Sciences*, vol. 43, pp. 425–440, Dec. 1991.

[31] S. Arora, D. Karger, and M. Karpinski, “Polynomial time approximation schemes for dense instances of NP-hard problems,” *Journal of Computer and System Sciences*, vol. 58, pp. 193–210, Feb. 1999.

[32] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, “Improving variational quantum optimization using CVaR,” *Quantum*, vol. 4, p. 256, Apr. 2020.

[33] J. J. Rotman, *Advanced modern algebra*, vol. 165. American Mathematical Soc., 2015.

[34] N. Biggs, *Algebraic Graph Theory.* Cambridge University Press, May 1974.

[35] J. H. Kim, B. Sudakov, and V. H. Vu, “On the asymmetry of random regular graphs and random graphs,” *Random Structures & Algorithms*, vol. 21, no. 3-4, pp. 216–224, 2002.

[36] P. Erdős and A. Rényi, “Asymmetric graphs,” *Acta Mathematica Academiae Scientiarum Hungarica*, vol. 14, no. 3-4, pp. 295–315, 1963.

[37] K. Balasubramanian, “Computer generation of automorphism groups of weighted graphs,” *Journal of Chemical Information and Modeling*, vol. 34, pp. 1146–1150, Sept. 1994.

[38] E. Farhi and A. W. Harrow, “Quantum supremacy through the quantum approximate optimization algorithm,” *arXiv:1602.07674*, 2016.

[39] A. Bapat and S. Jordan, "Bang-bang control as a design principle for classical and quantum optimization algorithms," *Quantum Info. Comput.*, vol. 19, p. 424–446, May 2019.

[40] L. Babai, "Graph isomorphism in quasipolynomial time," *arXiv:1512.03547*, 2015.

[41] V. Arving, "Lecture notes," 2007. [Online.] `https://www.cmi.ac.in/~ramprasad/lecturenotes/algcomp/tillnow.pdf`.

[42] I. S. Filotti and J. N. Mayer, "A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus," in *Proceedings of the twelfth annual ACM symposium on Theory of computing*, pp. 236–243, 1980.

[43] G. Miller, "Isomorphism testing for graphs of bounded genus," in *Proceedings of the twelfth annual ACM symposium on Theory of computing*, pp. 225–235, 1980.

[44] H. L. Bodlaender, "Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees," *Journal of Algorithms*, vol. 11, no. 4, pp. 631–643, 1990.

[45] M. Grohe, "Fixed-point definability and polynomial time on graphs with excluded minors," *Journal of the ACM (JACM)*, vol. 59, no. 5, pp. 1–64, 2012.

[46] M. Grohe, "Structural and logical approaches to the graph isomorphism problem.," in *SODA*, p. 188, 2012.

[47] B. D. McKay *et al.*, "Practical graph isomorphism," *Congressus Numerantium*, vol. 30, pp. 45–87, 1981.

[48] B. D. McKay and A. Piperno, "Practical graph isomorphism, II," *Journal of Symbolic Computation*, vol. 60, no. 0, pp. 94 – 112, 2014.

[49] P. Berman and M. Karpinski, "On some tighter inapproximability results (extended abstract)," in *Automata, Languages and Programming*, pp. 200–209, Springer Berlin Heidelberg, 1999.

[50] B. D. McKay, "Computing automorphisms and canonical labellings of graphs," in *Combinatorial mathematics*, pp. 223–232, Springer, 1978.

[51] P. T. Darga, M. H. Liffiton, K. A. Sakallah, and I. L. Markov, "Exploiting structure in symmetry detection for CNF," in *Proceedings of the 41st annual Design Automation Conference*, pp. 530–534, 2004.

[52] T. Junttila and P. Kaski, "Engineering an efficient canonical labeling tool for large and sparse graphs," in *2007 Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pp. 135–149, SIAM, 2007.

[53] J. L. López-Presa, L. F. Chiroque, and A. Fernández Anta, "Novel techniques to speed up the computation of the automorphism group of a graph," *Journal of Applied Mathematics*, vol. 2014, 2014.

[54] R. Shaydulin, I. Safro, and J. Larson, "Multistart methods for quantum approximate optimization," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, IEEE, Sept. 2019.

[55] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash, "Learning to optimize variational quantum circuits to solve combinatorial problems," *Proceedings of the Thirty-Forth AAAI Conference on Artificial Intelligence (AAAI-20)*, 2019.

[56] M. Wilson, S. Stromswold, F. Wudarski, S. Hadfield, N. M. Tubman, and E. Rieffel, "Optimizing quantum heuristics with meta-learning," *arXiv preprint arXiv:1908.03185*, 2019.

[57] G. Verdon, M. Broughton, J. R. McClean, K. J. Sung, R. Babbush, Z. Jiang, H. Neven, and M. Mohseni, "Learning to learn with quantum neural networks via classical neural networks," *arXiv:1907.05415*, 2019.

[58] G. B. Mbeng, R. Fazio, and G. Santoro, "Quantum annealing: a journey through digitalization, control, and hybrid quantum variational schemes," *arXiv:1906.08948*, 2019.

[59] A. Mowshowitz, "Entropy and the complexity of graphs: I. an index of the relative complexity of a graph," *The Bulletin of Mathematical Biophysics*, vol. 30, pp. 175–204, Mar. 1968.

[60] G. Simonyi, "Graph entropy: a survey," *Combinatorial Optimization*, vol. 20, pp. 399–441, 1995.

[61] A. Mowshowitz and M. Dehmer, "A symmetry index for graphs," *J. Math. Biophys*, vol. 30, pp. 533–546, 2010.

[62] B. D. MacArthur, R. J. Sánchez-García, and J. W. Anderson, "Symmetry in complex networks," *Discrete Applied Mathematics*, vol. 156, pp. 3525–3531, Nov. 2008.

[63] F. Ball and A. Geyer-Schulz, "How symmetric are real-world graphs? A large-scale study," *Symmetry*, vol. 10, p. 29, Jan 2018.

[64] I. Krasikova, A. Levb, and B. D. Thattec, "Upper bounds on the automorphism group of a graph," *Discrete Mathematics*, vol. 256, pp. 489–493, 2002.

[65] P. Dankelmann, D. Erwin, S. Mukwembi, B. G. Rodrigues, E. Mwambene, and G. Sabidussi, "Automorphism group and diameter of a graph," *Journal of Graph Theory*, vol. 70, no. 1, pp. 80–91, 2012.

[66] J. Håstad, "Some optimal inapproximability results," *Journal of the ACM*, vol. 48, pp. 798–859, July 2001.

[67] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM (JACM)*, vol. 42, pp. 1115–1145, Nov. 1995.

[68] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, "Optimal inapproximability results for MAX-CUT and other 2-variable CSPs?," *SIAM Journal on Computing*, vol. 37, no. 1, pp. 319–357, 2007.

[69] M. J. D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in Optimization and Numerical Analysis*, pp. 51–67, Springer Netherlands, 1994.

[70] M. J. D. Powell, "Direct search algorithms for optimization calculations," *Acta Numerica*, vol. 7, pp. 287–336, Jan. 1998.

[71] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online.] `http://www.scipy.org/`.

[72] S. Hudson, J. Larson, S. M. Wild, and D. Bindel, "libEnsemble users manual," 2019.

[73] J. Larson and S. M. Wild, "A batch, derivative-free algorithm for finding multiple local minima," *Optimization and Engineering*, vol. 17, no. 1, pp. 205–228, 2016.

[74] J. Larson and S. M. Wild, "Asynchronously parallel optimization solver for finding multiple minima," *Mathematical Programming Computation*, vol. 10, no. 3, pp. 303–332, 2018.

[75] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy 2008)* (G. Varoquaux, T. Vaught, and J. Millman, eds.), (Pasadena, CA USA), pp. 11–15, 2008.

[76] O. Tange, "Gnu parallel 2018," 2018. [Online] `https://doi.org/10.5281/zenodo.1146014`.

[77] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural information processing systems*, pp. 155–161, 1997.

[78] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[79] C.-C. Chang and C.-J. Lin, "LIBSVM," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, Apr. 2011.

[80] W. Waegeman and L. Boullart, "An ensemble of weighted support vector machines for ordinal regression," *International Journal of Computer Systems Science and Engineering*, vol. 3, no. 1, pp. 47–51, 2009.

[81] M. Perez-Ortiz, P. A. Gutierrez, and C. Hervas-Martinez, "Projection-based ensemble learning for ordinal regression," *IEEE Transactions on Cybernetics*, vol. 44, pp. 681–694, May 2014.

[82] C. Huang, M. Szegedy, F. Zhang, X. Gao, J. Chen, and Y. Shi, "Alibaba cloud quantum development platform: Applications to quantum algorithm design," 2019.

[83] D. Lykov, R. Schutski, A. Galda, V. Vinokur, and Y. Alexeev, "Tensor network quantum simulator with step-dependent parallelization," 2020.

[84] Z. Qiu, R. Shaydulin, X. Liu, Y. Alexeev, C. S. Henry, and I. Safro, "ELRUNA: Elimination rule-based network alignment," *preprint at arXiv:1911.05486*, 2019.

[85] E. Sadrfaridpour, T. Razzaghi, and I. Safro, "Engineering fast multilevel support vector machines," *Machine Learning*, vol. 108, pp. 1879–1917, May 2019.