Empirical performance bounds for quantum approximate optimization

Phillip C. Lotshaw* and Travis S. Humble†

Quantum Computational Sciences Group
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37830 USA

Rebekah Herrman‡ and James Ostrowski§

Department of Industrial and Systems Engineering,
University of Tennessee at Knoxville
Knoxville, Tennessee 37996-2315 USA

George Siopsis¶

Department of Physics and Astronomy,
University of Tennessee at Knoxville
Knoxville, Tennessee 37996-1200 USA

The quantum approximate optimization algorithm (QAOA) is a variational method for noisy, intermediate-scale quantum computers to solve combinatorial optimization problems. Quantifying performance bounds with respect to specific problem instances provides insight into when QAOA may be viable for solving real-world applications. Here, we solve every instance of MaxCut on non-isomorphic unweighted graphs with nine or fewer vertices by numerically simulating the pure-state dynamics of QAOA. Testing up to three layers of QAOA depth, we find that distributions of the approximation ratio narrow with increasing depth while the probability of recovering the maximum cut generally broadens. We find QAOA exceeds the Goemans-Williamson approximation ratio bound for most graphs. We also identify consistent patterns within the ensemble of optimized variational circuit parameters that offer highly efficient heuristics for solving MaxCut with QAOA. The resulting data set is presented as a benchmark for establishing empirical bounds on QAOA performance that may be used to test on-going experimental realizations.

I. INTRODUCTION

Noisy, intermediate-scale quantum (NISQ) computers may soon solve problems of practical importance with a quantum computational advantage [1, 2]. One promising approach uses the quantum approximate optimization algorithm (QAOA) [3–17] or its variants [18–24] to find approximate solutions to combinatorial optimization problems. By using alternating layers of a mixing operator and a cost operator, QAOA promises to prepare an approximation to the quantum state that maximizes the expectation value of the cost operator [3]. Moreover, the cost operator may represent an instance of unconstrained combinatorial optimization, which opens the application of QAOA to a wide variety of practical but challenging computational problems, including the well known graph problem MaxCut.

Alongside theoretical guarantees, the empirical performance of QAOA is an important open question for evaluating computational utility. The optimal number of alternating circuit layers as well as their tuning has been observed to vary with specific problem instances. Previous efforts examining MaxCut have tested performance in terms of the approximation ratio, which quantifies the average cost function value observed relative to the optimal value. These include studies on families of problem instances represented by 2-regular graphs [3–5], 3-regular graphs [6–10], small samples of random graphs [11, 12], and small samples of graphs with various fixed symmetries [13]. Notably, Crooks has shown that QAOA may exceed the performance bounds set by the Goemans-Williamson algorithm [12], which yields the best conventional lower bound on the approximation ratio for MaxCut. This collection of results has encouraged further study of the more general circumstances under which QAOA may provide a practical improvement in performance.

An additional measure of QAOA performance is the resources required to prepare the optimized circuit layers. By design, the alternating layers of QAOA are tuned to prepare the approximate quantum state using parameterized gates optimized with respect to the observed cost.
value. However, the effort to identify these optimized circuits becomes intractable for large numbers of parameters. Previous studies have used optimization heuristics based on machine learning \[12, 13\] and various numerical optimization algorithms \[6, 8, 11, 22\]. These results have been limited to small sets of graphs, often with simple regular structures, and an open question is whether there are general heuristics that yield good parameters quickly and consistently.

Here we use numerical simulations of pure state dynamics to quantify performance bounds of QAOA solving an exhaustive set of MaxCut instances for \( n \leq 9 \) vertex graphs at depths \( p \leq 3 \). The exhaustive problem instance set gives a thorough and systematic account of QAOA behavior on MaxCut for small graphs. We also evaluate the effectiveness of circuit optimization using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm \[22\] by benchmarking against exact optimization software and brute force solutions for all graphs at \( p = 1 \) and for graphs with \( n \leq 6 \) vertices at \( p = 2 \). We confirm BFGS returns optimal angles for QAOA state preparation in all these cases. Ultimately, our simulation results reveal patterns in the optimized circuits that support heuristics for more efficient parameter selection.

Our characterization of QAOA performance for solving MaxCut address the approximation ratio and the probability of obtaining the optimal result. While the former measure estimates the average cost function value returned by QAOA relative to the actual optimal value, the latter quantifies the probability to recover the optimal cut. We compile the results for each graph instance into a data set \[26\] that serves as a validated benchmark for support heuristics for more efficient parameter selection.

The remainder of the presentation is organized as follows: we review QAOA in Sec. \( \text{II} \) and discuss our approach to variational circuit simulations in Sec. \( \text{III} \). We present results from these simulations in Sec. \( \text{IV} \) including trends in the performance measures and gate parameters, before concluding with our key findings in Sec. \( \text{V} \).

\section{II. QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM}

The quantum approximate optimization algorithm (QAOA) is a variational algorithm designed to find good approximate solutions to combinatorial optimization problems \( \mathbb{R} \). A problem instance is specified by a cost function \( C(z) \) with \( z = (z_{n-1}, z_{n-2}, ..., z_0) \) and \( z_j \in \{0, 1\} \). QAOA recovers a candidate solution \( z^* \) and the quality of this candidate can be quantified by the normalized value of the cost function

\[ \mathcal{R}(z^*) = \frac{C(z^*)}{C_{\text{max}}} \]  \hspace{1cm} (1)

Here

\[ C_{\text{max}} = \max_z C(z) \]  \hspace{1cm} (2)

is the globally optimal value with optimal solution

\[ z_{\text{max}} = \arg \max_z C(z) \]  \hspace{1cm} (3)

and \( \mathcal{R}(z_{\text{max}}) = 1 \) while \( \mathcal{R} < 1 \) otherwise. QAOA encodes the bitstring \( z \) as a quantum state with respect to the computational basis \( |z_j\rangle \in \{|0\}, |1\rangle \) such that \( |z\rangle = |z_{n-1}, ..., z_0\rangle \). The cost function \( C(z) \) is expressed as the operator \( \hat{C} \) that is diagonal in the computational basis with matrix elements

\[ \langle z|\hat{C}|z\rangle = C(z). \]  \hspace{1cm} (4)

The initial state of QAOA is taken as a uniform superposition of the computational basis states

\[ |\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} |z\rangle \]  \hspace{1cm} (5)

that is transformed by a series of \( p \) unitary operations as

\[ |\psi_p(\gamma, \beta)\rangle = \left( \prod_{q=1}^{p} \hat{U}(B_q, \beta_q)\hat{U}(\hat{C}, \gamma_q) \right) |\psi_0\rangle \]  \hspace{1cm} (6)

with variational angle parameters \( \gamma = (\gamma_1, ..., \gamma_p) \) and \( \beta = (\beta_1, ..., \beta_p) \). The first unitary operator

\[ \hat{U}(\hat{C}, \gamma_q) = \exp(-i\gamma_q \hat{C}) \]  \hspace{1cm} (7)

applies a \( C(z) \)-dependent phase to each of the computational basis states. The second unitary

\[ \hat{U}(B_q, \beta_q) = \exp(-i\beta_q B_q) \]  \hspace{1cm} (8)

applies coupling in the computational basis as

\[ \hat{B} = \sum_{j=0}^{n-1} \hat{X}_j, \]  \hspace{1cm} (9)

where \( \hat{X}_j \) is the Pauli X operator on qubit \( j \). The latter transitions between the \( \{|z\rangle\} \) depend on the \( C(z) \)-dependent phases from \( \hat{U}(\hat{C}, \gamma_q) \) and the angle parameters \( (\gamma, \beta) \). QAOA selects these parameters to maximize the value \langle \hat{C} \rangle \), as discussed in more detail in Sec. \( \text{II}A \).

Measurement of the state \( |\psi_p(\gamma, \beta)\rangle \) in the computational basis yields each \( z_j \in \{0, 1\} \) and therefore a candidate solution \( z^* \). The probability to observe a specific \( z \) is given by the Born rule

\[ P(z) = |\langle z|\psi_p(\gamma, \beta)\rangle|^2. \]  \hspace{1cm} (10)

Following measurement, the observed result \( z \) is used to calculate the cost \( C(z) \). Repeating the sequence of preparation and measurement approximates the distribution of
States with the optimized angles are repeatedly prepared algorithms, as in an approach to Max-
perhaps even between varying problems and QAOA-like
observed in small samples of random graphs [11, 12], or
graphs. Other studies suggest that parameter patterns

timization. Zhou gives the best approximation.
brandão
circuits using linear interpolation or discrete cosine and
developed parameter optimization approaches for deep
and small samples of graphs examined so far.

given by Eq. (10). The proposed solution \( z^\ast \) may then
be selected as the argument that yields the largest cost
function as defined by Eq. (11).

As shown by Farhi et al. [8], the probability that a
measurement returns the optimal solution \( z_{\text{max}} \) converges
to unity as the QAOA depth \( p \) goes to infinity. Less is
known about the performance of QAOA at finite \( p \), where
studies suggest a modest \( p \) may suffice for achieving a quantum computational advantage [7, 12].

A. Gate parameter optimization

Finding optimal solutions, or good approximations, to
Eq. (1) requires optimizing the angle parameters that
tune each QAOA layer. A standard approach to
optimizing these angles is to pick an initial set of values
\((\gamma^{(0)}, \beta^{(0)})\) and then prepare and measure many copies
of the state \(|\psi_p(\gamma^{(0)}, \beta^{(0)})\rangle\) to estimate the expectation
value of the cost function operator as

\[
\langle \hat{C}(\gamma, \beta) \rangle = \sum_z P(z) C(z).
\]

Analytically, \( P(z) \) depends on \(|\psi_p(\gamma, \beta)\rangle\) through
Eq. (10).

When using an optimization algorithm, such as gra-
dient descent or BFGS [23], new angles \((\gamma^{(1)}, \beta^{(1)})\) are
tested for increases in \( \langle \hat{C} \rangle \). After \((\gamma^{(1)}, \beta^{(1)})\) have been
selected, the process is repeated with the new angles:
prepare and measure a set of states \(|\psi_p(\gamma^{(1)}, \beta^{(1)})\rangle\), send
the results \( \{z\} \) to a classical computer to calculate an
estimate of \( \langle \hat{C}(\gamma^{(1)}, \beta^{(1)}) \rangle \), then pick new angles to try
with the quantum computer. The optimization process
is repeated until a set of angles is found to maximize \( \langle \hat{C} \rangle \).
States with the optimized angles are repeatedly prepared
and measured to sample \( z \) and identify a solution \( z^\ast \) that
gives the best approximation.

Previous studies have found patterns in the optimized
parameters that can potentially simplify gate-angle
optimization. Zhou et al. [6] examined regular graphs and
developed parameter optimization approaches for deep
circuits using linear interpolation or discrete cosine and
sine transformations of parameters from lower depths.
Brandão et al. [8] have argued that parameters should
be transferable between graphs with sufficiently similar
structure, suggesting that approaches like that of Zhou
et al. may be applicable within any families of structured
graphs. Other studies suggest that parameter patterns
apply more generally between varying types of graphs, as
observed in small samples of random graphs [11, 12], or
perhaps even between varying problems and QAOA-like
algorithms, as in an approach to Max-\( k \) Vertex Cover
using a variant formulation of QAOA [22]. Together these
studies suggest that parameter patterns may be generic
and important in simplifying optimization for QAOA.
However, it is unclear if these patterns will extend over
general graphs or if they are byproducts of the structures
and small samples of graphs examined so far.

In addition to the angle parameters, a depth parameter
\( p \) defines a given state preparation circuit and instance
of QAOA. Higher depths theoretically give better perfor-
mance [8], but they also require more computationally
intensive optimizations. It is therefore efficient to use the
smallest depth needed to obtain a desired quality of solu-
tion. Previous studies have characterized performance
for 2-regular graphs at arbitrary depths [3, 4] and estab-
lished performance bounds for 3-regular graphs at several
low depths [10]. For general graphs, an analytical expres-
sion that characterizes performance has been derived for
the lowest-possible depth \( p = 1 \) [4, 5] and the relationship
between depth and performance under specific parameter
schedules has been examined for graphs with varying
symmetry [19]. However, much less is known about
QAOA performance on general graphs at depths \( p > 1 \)
and which depths suffice for applications.

B. Characterizing QAOA performance

The quality of a solution \( z^\ast \) is characterized by the nor-
malized cost function in Eq. (11). In QAOA, \( z^\ast \) depends
on measurements that probabilistically sample the
computational basis states following Eq. (10). Therefore, we
characterize QAOA performance using the quantum ap-
proximation ratio

\[
r = \langle \mathcal{R}(z) \rangle,
\]

where \( \mathcal{R}(z) \) is the normalized cost and the expectation
value is taken as in Eq. (11). The quantity \( r \) describes the
average \( \mathcal{R}(z) \) with respect to measurements of \(|\psi_p(\gamma, \beta)\rangle\). An argument about concentration in QAOA claims that
solutions \( z^\ast \) with \( \mathcal{R}(z^\ast) \geq r \) can be expected with high
probability given a modest number of quantum state
measurements [8]. The quantum \( r \) is thus also related
to the classical notion of an approximation ratio as a
minimum bound on performance for the optimization al-
gorithm.

A second way to characterize QAOA success quantifies the
probability to obtain the optimal solution \( z_{\text{max}} \) from
Eq. (3). In general, there can be multiple optimal solu-
tions, which we denote as the set of \( k \) optimal solutions
\( \{z^i_{\text{max}}\}_{i \in [k]} \), and the probability of optimizing the cost
function is

\[
P(C_{\text{max}}) = \sum_{i=1}^{k} P(z^i_{\text{max}}),
\]

where \( P(z^i_{\text{max}}) \) is the probability of \( z^i_{\text{max}} \) from Eq. (10).
Letting \( P_s \) be a desired threshold to recover the max cut,
the minimal number of samples \( N_s \) to observe a cut that
maximizes the cost function is given by

\[
N_s = \frac{\log (1 - P_s)}{\log (1 - P(C_{\text{max}}))},
\]

since the probability to observe no maximum cut in \( N_s \)
samples is \( P^s = (1 - P(C_{\text{max}}))^{N_s} \) and \( P^s = 1 - P_s \).
C. MaxCut

MaxCut partitions a graph $G$ such that the number of edges shared between the partitions is maximized. Let $G = (V, E)$ with $V = \{n-1, \ldots, 0\}$ the set of vertex labels and $E = \{(j, k) : j, k \in V\}$ the set of unweighted edges. For each vertex $j \in V$, a cut $z = (z_{n-1}, \ldots, z_0)$ assigns a binary label $z_j \in \{0, 1\}$ that denotes the corresponding set. We express the cost function for MaxCut as a sum of pair-wise clauses

$$C(z) = \sum_{(j,k)} C_{(j,k)}(z). \quad (15)$$

with

$$C_{(j,k)}(z) = z_j + z_k - 2z_jz_k = \begin{cases} 1 & z_j \neq z_k \\ 0 & z_j = z_k \end{cases} \quad (16)$$

An individual clause $C_{(j,k)}(z)$ is maximized when two vertices $j, k$ connected by an edge are assigned to opposing sets, and the purpose of MaxCut is to find a cut $z_{\text{max}}$ that maximizes Eq. (15). There are always at least two maximum cuts since the $C_{(j,k)}$ is invariant under $0 \leftrightarrow 1$ for all the bits $z_j$ in any $z$.

From Eq. (15), the cost function is cast as a sum of operators for the edges

$$\hat{C} = \sum_{(j,k)} \hat{C}_{(j,k)}, \quad (17)$$

with edge operators

$$\hat{C}_{(j,k)} = \frac{1}{2} \left( \mathbb{1} - \hat{Z}_j \hat{Z}_k \right), \quad (18)$$

where $\mathbb{1}$ is the identity operator and $\hat{Z}_j$ is the Pauli $Z$ operator on qubit $j$. The notation $\hat{Z}_j \hat{Z}_k$ uses an implicit tensor product with the identity operator on the Hilbert space of unlisted qubits.

III. SIMULATIONS OF QAOA

We simulate QAOA to solve MaxCut of all connected non-isomorphic graphs with $n \leq 9$ vertices. We only consider connected graphs since solutions for any disconnected graph can be constructed from separate solutions of the connected graph components. The number of graphs $N_n$ grows rapidly with the number of vertices $n$, as shown in Table I.

| $n$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----|---|---|---|---|---|---|---|---|
| $N_n$ | 1 | 2 | 6 | 21 | 112 | 853 | 11,117 | 261,080 |

TABLE I: The number of connected non-isomorphic graphs $N_n$ grows with the number of vertices $n$.

A. Global optimal solutions for $p = 1$

We first optimize instances of QAOA for depth $p = 1$ using the expectation value of the cost function operator

$$\langle \hat{C}(\gamma_1, \beta_1) \rangle = \sum_{(j,k)} \langle \hat{C}_{(j,k)}(\gamma_1, \beta_1) \rangle \quad (19)$$

Previously, Hadfield derived a general expression for $\langle \hat{C}(j,k) \rangle$ at depth $p = 1$ as

$$\langle \hat{C}_{(j,k)}(\gamma_1, \beta_1) \rangle = \frac{\sin(4\beta_1) \sin(\gamma_1) (\cos^d(\gamma_1) + \cos^c(\gamma_1))}{4} - \frac{\sin^2(2\beta_1) \cos^d + e - 2f(\gamma_1)(1 - \cos^f(2\gamma_1))}{4} + \frac{1}{2}, \quad (20)$$

where $d = \text{deg}(j) - 1$ and $e = \text{deg}(k) - 1$ are one less than the degrees of the vertices $j$ and $k$ connected by the edge $(j,k)$ and $f$ is the number of triangles containing $(j,k)$.

We maximize $\langle \hat{C}(\gamma_1, \beta_1) \rangle$ using the numerical optimization software Couenne [29, 30], which optimizes problems of the form

$$\max F(x, y) \quad \text{s.t.} \quad p_i(x, y) \leq 0 \quad \forall i \in [n]$$

$$x \in \mathbb{R}^n$$

$$y \in \mathbb{Z}^n \quad (21)$$

Here, we specify the continuous variable $x = (\gamma_1, \beta_1)$ and the integer variable $y$ is not used. The function $F(x, y)$ then corresponds with Eq. (19) while the polynomial constraints $p_i(x)$ are not included since there are no constraints on the angles. Couenne simplifies this problem by reformulating it with auxiliary variables. Then, a convex relaxation of the reformulated problem is found and solved using branch and bound techniques. The process is repeated until an optimal solution is recovered [29, 30]. The results provide globally optimal instances of QAOA for $p = 1$.

B. Numerical searches for general $p$

For QAOA depth $p > 1$, we use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimization algorithm [25]. BFGS has been used previously in a wide variety of contexts including QAOA [8]. BFGS begins with an initial set of angles and then iteratively finds angles that converge to a local maximum of $\langle C(\gamma, \beta) \rangle$. Each iteration is determined by the numerical gradient of $\langle \hat{C} \rangle$ and an approximate Hessian second-derivative matrix that is constructed from the gradient at successive steps. Using the approximate Hessian when calculating the steps gives faster convergence than first-order methods while also avoiding the computational expense of calculating the exact Hessian.
We test convergence of BFGS using random initial angles and monitoring the local maximum of $\langle C \rangle$. We repeat the initialization procedure a fixed number of times that varies with depth $p$. We use 50 random-angle seeds when $p = 1$, 100 random seeds when $p = 2$, and 500 seeds when $p = 3$. As shown in Fig. 1(a), the average and standard deviation of the quantum approximation ratio at $p = 1$ quickly converges to the solution obtained using Couenne. We compared results from BFGS to results from Couenne for every graph with $n \leq 8$ vertices and found exact agreement out to the 6-digit precision of Couenne. We were unable to verify solutions at $n = 9$ due to numerical issues with Couenne, so instead we checked these results using brute force searches. As shown in Fig. 1(b) and (c), we observe similar behavior in convergence for the cases of $p = 2$ and 3.

We extend our validation of BFGS for $n = 9$ with $p = 1$ and $n \leq 6$ at $p = 2$ using brute force search. We evaluate $\langle C \rangle$ for all $(\gamma_1, \gamma_2, \beta_1, \beta_2)$ on a grid with spacing $\pi/100$ for each angle. We found both BFGS and brute force return values for the optimal $\langle C \rangle$ to within an additive factor of $10^{-2}$. As BFGS consistently recovers larger maxima, we attribute these differences to coarse-graining in the brute force method. We conclude that BFGS is finding globally optimal results at $p = 2$ for these graphs.

To verify our results are consistent for $n > 6$ with $p = 2$ and for all $n$ with $p = 3$, we run BFGS calculations a second time using different sets of 100 or 500 random seeds. We observe an increase in $\langle C \rangle$ after the second round of BFGS for less than 1% of graphs at each $n$ and $p$. We conclude that BFGS typically finds globally optimal solutions at these higher $n$ and $p$ though it is unclear how deviations may grow beyond $p = 3$.

C. Symmetry analysis

The optimized angles recovered by BFGS exhibit a variety of symmetries, in which multiple angle solutions give the same optimized expectation value of the cost function. We simplify these results by systematically reporting a single assignment for the optimized angles to compare the distributions of angles recovered across different graph instances. Symmetry analysis proves essential for revealing patterns in the optimized angles discussed in Secs. III C - IV D. We summarize how we use the symmetries below with detailed descriptions of the symmetries deferred to the Appendix.

Many of these observed symmetries are identified by extending the analysis of Zhou et al. for regular graphs [6], which we apply more broadly to graphs where each vertex has even degree or each vertex has odd degree. Specifically, the angles $\beta_q$ are periodic over intervals of $\pi/2$ and, without loss of generality, may always transform to the interval $-\pi/4 \leq \beta_q \leq \pi/4$ for all $q$. We always set $\beta_1 < 0$ following the “time-reversal” symmetry that follows from the dynamics in Eq. (6). There are no symmetries in $\gamma_q$ for generic graphs, so generally these have $-\pi \leq \gamma_q \leq \pi$ for all $q$. For graphs where every vertex has even degree or every vertex has odd degree, we use symmetries to always transform to angles in the half-intervals $-\pi/2 \leq \gamma_q \leq \pi/2$. A small number of graphs have additional symmetries and for these we report angles with the most component $(\gamma_q, \beta_q)$ pairs inside the intervals

$$\Gamma = [-\pi/2, 0], \ B = [-\pi/4, 0].$$ \ \ \ (22)

IV. QAOA SIMULATION RESULTS

We analyze QAOA performance on the MaxCut problem for graphs with various numbers of vertices $n$ at various depths $p$ using the two performance measures of Section III B. We also discuss patterns in the observed optimized angle distributions and we construct a search heuristic that uses these patterns to reduce the computational expense of parameter optimization.
FIG. 2: Distributions of the approximation ratios \( r \) for graphs with \( n = 7 \) vertices. Panels (a),...,(d) show \( p = 0, \ldots, 3 \) respectively, the red dashed line shows the approximation ratio from the Goemann-Williamson algorithm \( r_{GW} \).

| \( p \) | \( \bar{r} \) | \( \sigma \) | \( F_{0.7} \) | \( F_{0.8} \) | \( F_{0.9} \) |
|---|---|---|---|---|---|
| 0 | 0.633 | 0.063 | 0.116 | 0.004 | 0.000 |
| 1 | 0.809 | 0.053 | 0.986 | 0.551 | 0.039 |
| 2 | 0.884 | 0.037 | 1.000 | 1.000 | 0.321 |
| 3 | 0.930 | 0.029 | 1.000 | 1.000 | 0.825 |

TABLE II: Mean \( \bar{r} \), standard deviation \( \sigma \), and complimentary cumulative distribution functions \( F_R(r) \) of Eq. (23) for the \( r \) distributions in Fig. 2.

A. Approximation ratio

Our first measure of QAOA performance on MaxCut is the approximation ratio \( r \) from Eq. (12). We present the distribution of \( r \) across graphs with various \( n \) and \( p \).

Figure 2 shows an example of the distribution of \( r \) for all 853 graphs with \( n = 7 \) vertices for depths \( p = 0, \ldots, 3 \) in panels (a)-(d), respectively. Note that \( p = 0 \) corresponds to the initial state in Eq. (5).

Trends for other \( n \) are similar and described further below. In Fig. 2 the width of the plotted bars is smaller than the histogram bin width \( w = 0.02 \) to clearly visualize the distributions. Each panel also overlays a Gaussian distribution \( (w/\sqrt{2\pi\sigma^2}) \exp(-(r-\bar{r})^2/2\sigma^2) \) using the mean \( \bar{r} \) and standard deviation \( \sigma \) in Table II as calculated from the observed distributions of \( r \).

Figure 2(a) shows \( r \) for the initial states of Eq. (5). Every \( z \) is equally probable in these states, so \( r \) is equivalent to the approximation ratio obtained by averaging random cuts from the uniform distribution. This cuts half the edges in \( E \) on average, \( \langle \hat{C} \rangle = E/2 \), while the maximum number of cuts is bounded by the total number of edges, \( C_{max} \leq E \), so \( r = \langle \hat{C} \rangle/C_{max} \geq 1/2 \).

Figures 2(b), (c), and (d) show similar distributions of the \( r \) for \( p = 1, 2, 3 \) respectively. These are well described by Gaussian distributions for each \( p \) and \( r \) is approaching one as \( p \) increases. QAOA instances also perform more similarly with increasing depth as indicated by the decrease in standard deviation of \( r \).

We further quantify increases in \( r \) with \( p \) by calculating \( F_R(r) \) as the fraction of graphs with approximation ratio \( r \) that exceeds a threshold \( R \). This is given by the complimentary cumulative distribution function

\[
F_R(r) = \frac{1}{N_n} \sum_{G(n)} \Theta(r - R), \tag{23}
\]

where \( N_n \) is the number of non-isomorphic connected graphs with \( n \) vertices, \( \sum_{G(n)} \) is the sum over these graphs, and \( \Theta \) is the Heaviside step function. Table III presents \( F_R(r) \) for \( n = 7 \) with \( R = 0.7, 0.8, \) and 0.9. Almost all graphs exceed \( r \geq 0.7 \) at \( p = 1 \) and more than half exceed \( r \geq 0.8 \). At \( p = 2 \), all graphs have \( r > 0.8 \) while most graphs have \( r > 0.9 \) at \( p = 3 \). We note that the latter results exceeds the best lower bound for the approximation ratio of \( r_{GW} = 0.87856 \) set by the Goemans-Williamson algorithm [31].

We analyzed the distribution of \( r \) for all \( n \leq 9 \). Similar trends are observed for all \( n \): the mean \( r \) increases as \( p \) increases and the distributions become narrower in terms of the standard deviation. The distributions of \( r \) over different graphs at the same \( n \) are well described by Gaussian distributions when \( n \geq 7 \), while for \( n < 7 \) the small numbers of graphs yield more irregular histogram distributions.

Figure 3 shows the mean of \( r \) as a function of \( n \) for \( p = 0, \ldots, 3 \) in panels (a)-(d), respectively. Averages for \( p = 0 \) are approximately 2/3 for all \( n \) while the standard deviation decreases with increasing \( n \) due to the number of graphs sampled increasing as \( N_n \). For \( p \neq 0 \) in Figs. 3(b)-(d), the smallest graphs reach an optimal \( r = 1 \) while larger graphs decay to steady values of \( r < 1 \) as \( n \) increases. There is not much variation in the mean or standard deviation of \( r \) between nearby \( n \), especially at the largest \( n \), so larger \( n \) might be expected to return similar \( r \) distributions. As \( p \) increases, at each \( n \) the mean \( r \) becomes larger and the standard deviations become smaller, which is consistent with the trends at \( n = 7 \) seen in Fig. 2. The mean \( r \) all exceed the Goemans-Williamson \( r_{GW} \) by at least a standard deviation at \( p = 3 \).

B. Probability of the maximum cut

We next assess the performance of QAOA in terms of the probability for obtaining the maximum cut, \( P(C_{max}) \). Figure 4 shows an example of the normalized distribution of the \( P(C_{max}) \) for the 853 graphs with \( n = 7 \) at varying depths \( p \). Statistics describing these distributions are given in Table III.

The distribution for \( p = 0 \) shown in Fig. 4(a) peaks around \( P(C_{max}) \approx 0 \) with a rapid decrease away from zero, as expected. An exponential fit for this histogram
distribution yields \( \exp(-\kappa_0 P(C_{\text{max}})) \) with \( \kappa_0 = 29.1 \pm 0.9 \). At \( p = 0 \), each possible cut is equally probable in the initial state and there are \( 2^n \) possible cuts with at least two maximum cuts. This implies the bound

\[
P_0(C_{\text{max}}) \geq 1/2^{n-1},
\]

(24)

where the “0” subscript refers to the initial state. The minimum value \( P_0(C_{\text{max}}) = 1/2^{n-1} \) gives the main contribution to the lowest bar in Fig. 4(a) while higher values are obtained for those graphs with additional symmetries, for example, in a complete graph where symmetry gives \( n!/([n/2]![n/2]) \) maximum cuts. The peak around \( P_0(C_{\text{max}}) = 1/2^{n-1} \) indicates that large degeneracies like this are uncommon in the total set of graphs.

Figure 4(b) shows \( P(C_{\text{max}}) \) when \( p = 1 \). We fit this distribution with an exponential starting at the second histogram bar and similar to the \( p = 0 \) distribution. This yields a smaller decay constant \( \kappa_1 = 7.0 \pm 0.3 \). Panels (c) and (d) show the distributions for \( p = 2 \) and 3, respectively. These distributions are better described by Gaussian distributions with means and standard deviations presented in Table III. The means increase with \( p \) and the distributions widen indicating greater differences between graphs at larger \( p \). The graphs pass thresholds \( P(C_{\text{max}}) > R \) gradually, as seen by \( \bar{F}_R(P(C_{\text{max}})) \).

We contrast distributions of \( P(C_{\text{max}}) \) against distributions of the quantum approximation ratio \( r \) in Fig. 2 and Table II. The mean of \( P(C_{\text{max}}) \) increases the least when going from \( p = 0 \) to \( p = 1 \), while by contrast the mean of \( r \) increases the most in going from \( p = 0 \) to \( p = 1 \). The exponential and Gaussian distributions are also significantly different at small \( p \). At higher \( p \), \( P(C_{\text{max}}) \) and \( r \) both follow Gaussian distributions but there are differences in how their widths change with \( p \), with increasing differences between different graphs for \( P(C_{\text{max}}) \) and the opposite for \( r \).

We explain why distributions of \( r \) have high averages and narrow widths when concurrently \( P(C_{\text{max}}) \) presents broad widths and relatively small averages. Consider the normalized eigenspectrum \( C/C_{\text{max}} \) of energies \( \mathcal{R}(z) = C(z)/C_{\text{max}} \). For each \( n \), we made a histogram of \( \mathcal{R}(z) \) for each graph, then average these values to obtain an average spectrum for \( \mathcal{R}(z) \). Figure 4 shows the average spectrum of \( \mathcal{R}(z) \) at \( n = 7 \) vertices with each data point indicating the fraction of basis states in a histogram bin of width 1/12 averaged over all graphs with \( n = 7 \). The results look similar for other \( n \geq 6 \) while the distributions for \( n \leq 5 \) are irregular due to small numbers of graphs.

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
p & P_0(C_{\text{max}}) & \sigma & \bar{F}_{0.25} & \bar{F}_{0.50} & \bar{F}_{0.75} \\
\hline
0 & 0.046 & 0.051 & 0.000 & 0.001 & 0.000 \\
1 & 0.211 & 0.132 & 0.278 & 0.035 & 0.009 \\
2 & 0.426 & 0.172 & 0.835 & 0.318 & 0.046 \\
3 & 0.610 & 0.187 & 0.975 & 0.715 & 0.252 \\
\hline
\end{array}
\]

TABLE III: Mean \( \bar{F}(C_{\text{max}}) \), standard deviation \( \sigma \), and complimentary cumulative distribution functions \( \bar{F}_R(P(C_{\text{max}})) \) similar to Eq. (23) for the \( P(C_{\text{max}}) \) distributions in Fig. 4.
of the way up the list, rounded to the nearest integer.

Most basis states give sub-optimal cuts \( R(z) < 1 \) in the average spectrum of Fig. 5, with the majority of states in the range \( 2/3 \lesssim R(z) \lesssim 0.95 \) and with few optimal states at \( R(z) = 1 \). Thus, optimization of \( r \) in state preparation favors large total probabilities in many states with near-optimal \( R(z) \) over the smaller probability of preparing truly optimal states. This gives relatively high and uniform \( r \) for different graphs but a much more variable \( P(C_{\text{max}}) \).

Figure 6 shows how the average of \( P(C_{\text{max}}) \) varies for the distribution of graphs at various \( n \) and \( p \) using asymmetric error bars expressed as quartiles. The mean of \( P(C_{\text{max}}) \) decreases exponentially with \( n \) as shown by the blue fit to the averages

\[
P_{\text{mean}}^{\text{fit}}(C_{\text{max}}) = N_p e^{-kp_n}
\]

with the fit parameters in Table IV. We fit to a subset of the data at each \( p \) to include only the largest \( n \) where decay is observed, and we show the \( n \) we fit at each \( p \) by the location of the fit curve.

At \( p = 0 \) the first quartiles follow the minimum of Eq. (24), shown by the dashed black line. This indicates many of the graphs saturate the minimum theoretical value of \( P_0(C_{\text{max}}) \). As \( n \) increases, the averages and quartiles of the \( P(C_{\text{max}}) \) distributions approach small values near the minimum of Eq. (24).

Figure 6(b) shows the averages and quartiles of the distributions of \( P(C_{\text{max}}) \) at \( p = 1 \). The average \( P(C_{\text{max}}) \) have increased relative to \( p = 0 \) and are again decreasing exponentially with \( n \), but the rate of exponential decay \( k_p \) is about half of the rate at \( p = 0 \), see Table IV. The error bars denoting quartiles reduce for \( n = 3 \), as these graphs are approaching full optimization. For higher \( n \) the error bars increase to indicate there are larger deviations between different graphs at \( p = 1 \).

The same trends continue at higher \( p \): the average \( P(C_{\text{max}}) \) increase with \( p \), with decreasing exponential decay constants in Table IV. The distributions widen as \( p \) increases at large \( n \), indicating \( P(C_{\text{max}}) \) is increasingly sensitive to graph structure. This is in contrast to the distributions of the \( r \), which narrow as \( p \) increases, with more similar \( r \) for different graphs.

### C. Optimized angle distributions

We next discuss the distribution of optimized angle parameters shown in Figs. 7 for the 853 graphs at \( n = 7 \). These figures present two-dimensional histograms of the angle distributions, where the optimized \((\gamma_j, \beta_j)\) have been organized into bins of size \( \pi/20 \times \pi/20 \) and counted. We use a logarithmic color scale to visualize the distributions and add a base of one counts in each bin so the logarithm does not diverge when there are zero counts.

Patterns have been observed previously in the optimized angles for solving simple families of graphs. Optimized angle patterns have been observed for 3-regular graphs [6], small samples of random graphs [11, 12], and even in a variant formulation of QAOA solving the Max-k Vertex Cover problem [22]. An argument has been given that angle patterns should be expected within families of graphs with systematic structure [9]. However, it is not clear if similar angle patterns hold for over the set of general graphs considered here.

#### 1. Angle patterns at \( p = 1 \)

Figure 7 shows the distribution of optimized angles for all graphs with \( n = 7 \) vertices at depth \( p = 1 \) in Eq. (6). The overwhelming majority of the angles are focused in the bright spot near \( \gamma_1 \approx -\pi/6, \beta_1 \approx -\pi/8 \). This concentration is generic for all \( n \) studied: a very limited range of angles \((\gamma_1, \beta_1)\) are observed to optimize almost all graphs with \( p = 1 \).

We categorize the optimized angles using a partitioning of parameter space. Using \( \Gamma = [-\pi/2, 0] \) and \( B = [-\pi/4, 0] \) from Eq. (22), we say angle-pairs \((\gamma_j, \beta_j)\) inside \((\Gamma, B)\) follow the angle patterns and say angle-pairs outside \((\Gamma, B)\) deviate from the pattern. Categorizing by

| \( p \) | \( N_p \) | \( k_p \) |
|---|---|---|
| 0 | \( 2.1 \pm 0.8 \) | \( 0.65 \pm 0.02 \) |
| 1 | \( 2.3 \pm 0.2 \) | \( 0.34 \pm 0.02 \) |
| 2 | \( 2.6 \pm 0.1 \) | \( 0.260 \pm 0.008 \) |
| 3 | \( 2.6 \pm 0.2 \) | \( 0.210 \pm 0.008 \) |

Table IV: Parameters for the exponential fits Eq. (25) in Fig. 6.
The number of graphs that deviate increases with \( n \). Every graph follows the pattern with \( n \) that extends to categorizing graphs when \( p \) follows the pattern when they are inside the parameter space of (\( \Gamma, B \)). The majority of (\( \gamma_1, \beta_1 \)) are still concentrated near \( \pi/4 \), with the symmetry of Section \( \text{III C} \), and a small cluster of graphs with angles \( \gamma_1 \approx -9\pi/10, \beta_1 \approx -\pi/4 \). There is a new cluster of angles at \( \gamma_1 \approx \pi/3 \) with a spread over all the \( \beta_1 \) (recall \( \beta_1 \leq 0 \) by the symmetry of Section \( \text{III C} \)), and a small cluster of graphs with angles \( \gamma_1 \approx -\pi/4, \beta_1 \approx -\pi/13 \). There are also subsets of graphs with angles that form clusters around seemingly-random parameter values.

We again categorize the optimized angles as agreeing with the pattern when they are inside the parameter space of (\( \Gamma, B \)) from Eq. \( (22) \) and deviating from the

![Graphs](image)

FIG. 6: Probabilities for obtaining the maximum cut \( P(C_{\text{max}}) \), averaged over graphs at various \( n \), with \( p = 0, \ldots, 3 \) in panels (a)–(d) respectively. Asymmetric error bars show the quartiles of the distributions of \( P(C_{\text{max}}) \), the dashed black line in (a) follows \( 1/2^{n-1} \) from Eq. \( (24) \), solid blue lines show the exponential fits of Eq. \( (25) \) to the indicated data, with parameters from Table \( \text{V} \).

![Histogram](image)

FIG. 7: Optimized angle distribution histogram for \( p = 1 \) QAOA on graphs with \( n = 7 \) vertices.

\[
D_{n,p} = \frac{D_{n,p}}{N_n}
\]

be the fraction of \( n \)-vertex graphs that deviate from the patterns at depth \( p \), where the \( N_n \) is the total number of graphs from Table \( \text{IV} \). We calculate the \( D_{n,p} \) using a numerical error tolerance of \( \epsilon = 10^{-5} \), so that \( \gamma_j \) that are in \( \Gamma \) within numerical error are counted as \( \gamma_j \in \Gamma \), and similarly for the \( \beta_j \).

Table \( \text{V} \) lists \( D_{n,p} \) and the fractions \( D_{n,p} \) of graphs that do not follow the angle patterns at \( p = 1 \) for \( n \leq 9 \). Every graph follows the pattern with \( n \leq 6 \), a small number of graphs deviate beginning with \( n = 7 \). The number of graphs that deviate increases with \( n \), but they are a small fraction of the total number of graphs.
FIG. 8: Angle patterns at depth $p = 2$ for graphs with $n = 7$ vertices, with $(\gamma_1, \beta_1)$ from the first layer in (a) and $(\gamma_2, \beta_2)$ in (b).

pattern when they are outside $(\Gamma, B)$ to within numerical error. The total fraction of graphs that deviate from the pattern is $\mathcal{D}_{n, p}$ from Eq. (29). We further separate the $\mathcal{D}_{n, p}$ into components to understand which $(\gamma_j, \beta_j)$ deviate and their correlations at $p = 2, 3$.

Let $\mathcal{D}_{n, p}^{(j)}$ denote the fraction of graphs that deviate only in the $j$th angle pair, $(\gamma_j, \beta_j) \notin (\Gamma, B)$ and $(\gamma_k, \beta_k) \in (\Gamma, B)$ for all $k \neq j$. Let $\mathcal{D}_{n, p}^{(jk)}$ denote the fraction of graphs where two angle pairs deviate, $(\gamma_j, \beta_j) \notin (\Gamma, B)$ and $(\gamma_k, \beta_k) \notin (\Gamma, B)$ but $(\gamma_l, \beta_l) \in (\Gamma, B)$ for all $l \notin \{k, j\}$, and similarly let $\mathcal{D}_{n, p}^{(jkl)}$ denote the fraction of graphs where the $j$th, $k$th, and $l$th angle-pairs deviate from $(\Gamma, B)$. The total fraction of graphs that deviate from the angle patterns is the sum of fractions that deviate in different sets of angle pairs, for example, at $p = 2$ the total fraction of graphs that deviate is the sum of fractions that deviate in one or both angle pairs, $\mathcal{D}_{n, p} = \mathcal{D}_{n, p}^{(1)} + \mathcal{D}_{n, p}^{(2)} + \mathcal{D}_{n, p}^{(12)}$.

Table VI shows the fractions of graphs that deviate from the angle patterns at $p = 2$. The total fraction of graphs that deviate $\mathcal{D}_{n, p}$ typically decreases as $n$ increases, however, the $\mathcal{D}_{n, p}$ is also increasing with $p$ as seen by comparison with Table VI. The angles deviate most often in $(\gamma_2, \beta_2)$ or in both $(\gamma_1, \beta_1)$ and $(\gamma_2, \beta_2)$, with $\mathcal{D}_{n, p}^{(2)} \approx \mathcal{D}_{n, p}^{(12)}$, while deviations in only $(\gamma_1, \beta_1)$ are observed less often, with $\mathcal{D}_{n, p}^{(1)} < \mathcal{D}_{n, p}^{(2)}$, $\mathcal{D}_{n, p}$.

Figure 9 shows the optimized angle distributions at $p = 3$ and $n = 7$; Table VII shows the fractions of graphs that deviate from $(\Gamma, B)$. These continue the pattern: the majority of angles are concentrated in a single cluster in $(\Gamma, B)$ at each layer, with additional small clusters of angles distributed unpredictably over the parameter space. The fractions of graphs in these clusters increases with $p$. Deviations from the angle patterns occur most often in the final layer of angles and in combinations connecting the final layer with earlier layers, with relatively large $\mathcal{D}_{n, p}^{(3)} \approx \mathcal{D}_{n, p}^{(23)} \approx \mathcal{D}_{n, p}^{(123)}$. Typically, smaller fractions of graphs deviate in the earlier layers only, as in $\mathcal{D}_{n, p}^{(1)}$, $\mathcal{D}_{n, p}^{(2)}$, and $\mathcal{D}_{n, p}^{(12)}$, or in disjoint layers, as in $\mathcal{D}_{n, p}^{(13)}$. The total fractions of graphs that deviate from the patterns $\mathcal{D}_{n, p}$ decreases with $n$ and increases with $p$.

The deviations in the clusters of angles away from $(\Gamma, B)$ appears to be due to differences in graph structure. We have confirmed that optimal angles deviate from $(\Gamma, B)$ for some graphs. While parameter optimization is limited by the sampling of random seeds with BFGS, these deviations are not attributed to limits on sampling. We expect the clusters contain graphs with similar structures, with more graphs in the clusters at higher $p$ indicating a greater sensitivity to graph struc-
TABLE VIII: Median optimized angles at $p = 3$, shown to five decimal places.

| $n$ | $p$ | $D_{n,p}^{(1)}$ | $D_{n,p}^{(2)}$ | $D_{n,p}^{(3)}$ | $D_{n,p}^{(4)}$ | $D_{n,p}^{(5)}$ | $D_{n,p}^{(6)}$ | $D_{n,p}^{(7)}$ | $D_{n,p}^{(8)}$ |
|-----|-----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 2   | 3   | 0.000          | 0.000          | 0.000          | 0.000          | 0.000          | 0.000          | 0.000          | 0.000          |
| 3   | 3   | 0.000          | 5.000×10^{-1}  | 0.000          | 0.000          | 0.000          | 0.000          | 0.000          | 5.000×10^{-1}  |
| 4   | 3   | 0.000          | 0.000          | 1.667×10^{-1}  | 0.000          | 0.000          | 0.000          | 0.000          | 5.000×10^{-1}  |
| 5   | 3   | 0.000          | 4.762×10^{-2}  | 9.524×10^{-2}  | 0.000          | 4.762×10^{-2}  | 2.381×10^{-1}  | 4.762×10^{-2}  | 4.762×10^{-2}  |
| 6   | 3   | 8.929×10^{-2}  | 0.000          | 6.250×10^{-2}  | 8.929×10^{-2}  | 1.786×10^{-2}  | 1.250×10^{-1}  | 8.036×10^{-2}  | 3.036×10^{-1}  |
| 7   | 3   | 0.000          | 1.172×10^{-3}  | 3.517×10^{-2}  | 3.517×10^{-2}  | 2.345×10^{-3}  | 5.627×10^{-2}  | 3.048×10^{-2}  | 1.290×10^{-1}  |
| 8   | 3   | 0.000          | 4.498×10^{-3}  | 1.079×10^{-2}  | 5.397×10^{-4}  | 9.895×10^{-4}  | 1.430×10^{-2}  | 1.934×10^{-2}  | 4.642×10^{-2}  |
| 9   | 3   | 0.000          | 6.511×10^{-3}  | 2.731×10^{-2}  | 4.979×10^{-3}  | 1.264×10^{-2}  | 2.214×10^{-3}  | 6.764×10^{-3}  | 1.195×10^{-2}  |

D. Median Angles

We consider how patterns of optimized angles may identify good approximate angles for most graphs and greatly reduce the computational cost of searching for angles with BFGS. Similar uses of angle patterns have been considered for 3-regular graphs [9], families of structured graphs [3], and small samples of random graphs [12].

We define a set of angles that follows the angle patterns by first taking the median $\gamma_1$ over all $\gamma_j$ for graphs with $n = 7$ and $p = 3$, then define similar median $\gamma_j$ and $\beta_j$ for each $j$ to obtain the set of angles shown in Table VII. We consider two approaches to QAOA that use these angles to avoid the computationally expensive random seeding of the standard BFGS approach. The first approach uses the median angles in the evolution of Eq. (10) without any optimization, results from this approach are denoted $\gamma_m$ and $P_m(C_{m})$. The second approach uses the median angles as seeds in a single BFGS optimization, results from this approach are denoted $\gamma_{mb}$ and $P_{mb}(C_{mb})$. Figures 10(a) and 11 assesses the probability of obtaining the maximum cut that are typically identical or close to results from the BFGS search with random seeds and are calculated at a small fraction of the computational expense.

Figure 10(b) compares the standard $\gamma_m$ from single BFGS optimizations using the median angles as seeds. The $\gamma_{mb}$ are significantly improved in comparison with the $\gamma_m$, with 87% of the results satisfying $\gamma_{mb} = \gamma$ up to an additive factor of $10^{-6}$, with significantly reduced mean and standard deviation of the difference $\gamma_{mb} - \gamma$ in Table IX. The approximation ratio from the single runs of BFGS with the median angle seeds are typically identical or close to results from the BFGS search with random seeds.
TABLE IX: Mean and standard deviations of the differences between quantities $r_{m}, r_{mB}, P_{m}(C_{\text{max}})$, and $P_{mB}(C_{\text{max}})$ calculated with the median angles and quantities $r$ and $P(C_{\text{max}})$ calculated from BFGS optimizations with random seeds, see text for details.

|                  | $r - r_{m}$ | $r - r_{mB}$ | $P(C_{\text{max}}) - P_{m}(C_{\text{max}})$ | $P(C_{\text{max}}) - P_{mB}(C_{\text{max}})$ |
|------------------|-------------|--------------|---------------------------------------------|---------------------------------------------|
| **mean**         | $2.2434 \times 10^{-2}$ | $1.4198 \times 10^{-3}$ | $7.2206 \times 10^{-2}$ | $1.2641 \times 10^{-2}$ |
| **standard deviation** | $2.4664 \times 10^{-2}$ | $6.2716 \times 10^{-3}$ | $1.0585 \times 10^{-1}$ | $5.9882 \times 10^{-2}$ |

FIG. 10: Two-dimensional histograms comparing the approximation ratio from the full BFGS search to results using the median angles without optimization (a) and using the the median angles as seeds in BFGS (b), for $n = 7$ and $p = 3$. Dashed white lines indicate the diagonals where $r_{m} = r$ and $r_{mB} = r$.

FIG. 11: Histograms similar to Fig. [10] but showing the probability of obtaining a maximum cut.

V. CONCLUSIONS

We have presented results that identify empirical performance bounds on optimized instances of QAOA for MaxCut. Using numerical simulations, we investigated an exhaustive set of MaxCut instances for depths $p \leq 3$ on graphs with $n \leq 9$ vertices. We calculated optimal solutions using exact numerical search and brute force search at $p = 1$ and 2 and validated results from BFGS at depths $p \leq 3$. The catalog of graph instances, optimized angles, and simulated states are available online [26].

Our analysis used the approximation ratio $r$ and the probability for obtaining a maximum cut $P(C_{\text{max}})$ as measures of QAOA performance. We observed that $r$ becomes more similar across graph structures as $p$ and $n$ increase and the Gaussian distributions of $r$ narrow.
Most graphs at \( n \leq 9 \) exceed the Goemans-Williamson bound by \( p = 3 \) in these narrow distributions, indicating viability of modest-depth QAOA to outperform GW.

In contrast to the narrowing distributions of \( r \), distributions of \( P(C_{max}) \) were found to broaden with increasing \( p \). We attributed the difference to design of the cost function and its corresponding spectrum. A preponderance of nearly-optimal states skews the optimization of \( r \) away from the much smaller set of truly optimal states. While this yields a large, uniform value for \( r \), it leaves \( P(C_{max}) \) less constrained. One alternative is to focus optimization on state preparation to ensure the largest values of \( P(C_{max}) \), for example by choosing gate angles that optimize a non-linear function such as \( \exp(\alpha C) \).

For \( p > 0 \), we observed an exponential decay in \( P(C_{max}) \) with respect to \( n \). The rate constant, \( k_p \), was found to decrease as \( p \) increases, and this raises the question as to whether such exponential behavior can predict performance metrics at larger \( n \) and \( p \).

The patterns observed in the optimized angles across this exhaustive set of instances mirrors previous results for narrower cases \( 6, 9, 11, 12, 22 \). Using these patterns as a search heuristic demonstrated high quality results for most graphs and required a significantly smaller computational cost than BFGS search with random seeding. Identifying the prevalence of these patterns at larger \( n \) and \( p \) as well as the correlation with graph properties can enable new search heuristics, for example, for general QAOA-like algorithms, as suggested by the work of Cook et al. \( 22 \).

ACKNOWLEDGMENTS

P. C. L. thanks Zak Webb and Yan Wang for discussing time-reversal symmetry. This work was supported by DARPA ONISQ program under award W911NF-20-2-0051. J. Ostrowski acknowledges the Air Force Office of Scientific Research award, AF-FA9550-19-1-0147. G. Siopsis acknowledges the Army Research Office award W911NF-19-1-0397. J. Ostrowski and G. Siopsis acknowledge the National Science Foundation award OMA-1937008. This research used resources of the Compute and Data Environment for Science (CADES) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

APPENDIX: ANGLE SYMMETRIES

In this Appendix we discuss details of the angle symmetries from Section III B. Most of the symmetries have been described previously by Zhou et. al. \( 6 \), although we note two of the symmetries they described for regular graphs apply more broadly to graphs where every vertex has even degree or every vertex has odd degree. Each symmetry relates different sets of angles \((\gamma, \beta)\) and \((\gamma', \beta')\) that give the same approximation ratio in Eq. (11).

\[
\langle \tilde{C}(\gamma, \beta) \rangle = \langle \tilde{C}(\gamma', \beta') \rangle, \tag{27}
\]

where

\[
\gamma = (\gamma_1, ..., \gamma_{q-1}, \gamma_q, \gamma_{q+1}, ..., \gamma_p)
\]

\[
\beta = (\beta_1, ..., \beta_{q-1}, \beta_q, \beta_{q+1}, ..., \beta_p), \tag{28}
\]

with \((\gamma', \beta')\) related to \((\gamma, \beta)\) in different ways for the different symmetries. An example is periodic behavior of the \( \beta_q \) angles over intervals of \( \pi/2 \), we use this to restrict every \( \beta_q \) component to the interval \(-\pi/4 \leq \beta_q \leq \pi/4\), as described by Zhou et. al. There are a variety of additional symmetries we describe below, with derivations in subsequent subsections.

The first type of symmetry applies to graphs where every vertex has even degree. In this case the \( \gamma_q \) angles are periodic over intervals of \( \pi \) since all the eigenvalues of \( C \) are even, as shown in detail in the next subsection. Thus the symmetry of Eq. (27) holds for any pair of angles with \( \beta = \beta' \) and

\[
\gamma' = (\gamma_1, ..., \gamma_{q-1}, \gamma_q \pm \pi, \gamma_{q+1}, ..., \gamma_p), \tag{29}
\]

where \( \gamma' \) differs from \( \gamma \) by a shift \( \gamma_q' = \gamma_q \pm \pi \) for any \( q \). We use this symmetry to organize the angle distributions so \( |\gamma_q| \leq \pi/2 \) for all \( q \) for graphs where every vertex degree is even. We search for generic optimized angles in our implementation of the BFGS algorithm from Section II B but if we find a \( |\gamma_q| > \pi/2 \) then we add or subtract \( \pi \) to get a \( |\gamma_q'| \leq \pi/2 \).

The second type of symmetry applies to graphs where every vertex has odd degree. This gives a joint symmetry in both sets of angles in Eq. (27),

\[
\gamma' = (\gamma_1, ..., \gamma_{q-1}, \gamma_q \pm \pi, \gamma_{q+1}, ..., \gamma_p)
\]

\[
\beta' = (\beta_1, ..., \beta_{q-1}, -\beta_q, -\beta_{q+1}, ..., -\beta_p). \tag{30}
\]

In Eq. (30), \( \gamma' \) differs from \( \gamma \) in the \( q \)th component, \( \gamma_q' = \gamma_q \pm \pi \) for any \( q \), and \( \beta' \) differs from \( \beta \) in the sign of all subsequent components, \( \beta'_r = -\beta_r \) for all \( r \geq q \). The proof follows from Pauli operator commutation relations, as shown in detail later. In our calculations, we use this symmetry to organize the angles so \( |\gamma_q| \leq \pi/2 \) for all \( q \) for graphs where every vertex degree is odd, similar to how we organize the angles when all the vertex degrees are even.

The third analytic symmetry we use is the “time-reversal” symmetry \( 4 \)

\[
(\gamma', \beta') = (-\gamma, -\beta). \tag{31}
\]
The symmetry is related to the \( \hat{B} \) and \( \hat{C} \) operators in Eqs. (7)-(8) and the initial state \( |\psi_0\rangle \) of Eq. (4), which have real-valued matrix elements and coefficients in the computational basis. We give a proof at the end of the Appendix. We use this symmetry to always transform to angles with \( \beta_1 \leq 0 \). The \( \beta_q \) for \( q > 1 \) can be positive or negative. We find additional symmetries in the optimized angles for a small subset of graphs. For these, we report the angles with the most component-pairs \( (\gamma_i, \beta_i) \in (\Gamma, B) \) from Eq. (22). Using the angles with the most components in \( (\Gamma, B) \) is designed to emphasize the angle patterns in Section IV C. The number of graphs \( M(p) \) we used the \( (\Gamma, B) \) symmetry on is shown for each \( n \) and \( p \) in Table X. The symmetry was used most often when a graph became fully optimized. For example, we found two sets of angles for an \( n = 3 \) graph that optimized the cost function \( \langle \hat{C} \rangle = C_{\text{max}} \); we saved the angles with the most components in \( (\Gamma, B) \). Asterisks in the table denote each graph which we used the rule on for that \( n \) and \( p \) was fully optimized. Note when graphs are optimized at some \( p \) we do not simulate them at depths \( p' > p \), so they do not carry over to columns for greater depths in the table. Overall, the rule is applied to a very limited subset of the graphs we study.

| \( n \) | \( M(1) \) | \( M(2) \) | \( M(3) \) |
|------|----------|----------|----------|
| 2    | 0        | -        | -        |
| 3    | 0        | 1*       | -        |
| 4    | 0        | 2*       | 0        |
| 5    | 0        | 2*       | 1*       |
| 6    | 0        | 0        | 2*       |
| 7    | 0        | 0        | 2*       |
| 8    | 0        | 1        | 1        |
| 9    | 0        | 0        | 1*       |

TABLE X: Numbers of graphs \( M(p) \) where we found degeneracies and used the rule of saving the angles with the most components in \( (\Gamma, B) \) following Eq. (22), for various depths \( p \). Numbers with asterisks denote all the graphs we used the rule on were fully optimized, dashes denote all the graphs were optimized at smaller depths.

under changes \( \gamma_q \rightarrow \gamma_q \pm \pi \), since
\[
\langle z | \hat{U}(\hat{C}, \gamma_q) | z \rangle = \exp(\mp i 2m_z \pi) \exp(-i 2m_z \gamma_q)
\]
where the factor \( 1 = \exp(\mp i 2m_z \pi) \) connects the two expressions.

To show the eigenvalues \( C(z) \) are all even when all the vertex degrees are even, we begin by showing there exists a single bitstring \( z^{(0)} \) for which \( C(z^{(0)}) \) is even. Then we show that if \( C(z) \) is even for any \( z \) and all the vertex degrees are even, then modifying any bit \( z_k \) in the bitstring \( z \) to get a new bitstring \( z' \) will give a \( C(z') \) that is also even. Together these imply that every bitstring has even \( C(z) \) when all the vertex degrees are even, since any bitstring can be made from a series of modifications to \( z^{(0)} \) and every modification gives an even \( C(z) \).

First consider the zero bitstring \( z^{(0)} = (0, 0, ..., 0) \). From Eqs. (15)-(16) this has \( C(z^{(0)}) = 0 \) which is even. Next consider an arbitrary bitstring \( z = (z_{n-1}, z_{n-2}, ..., z_k, ..., z_0) \) for which \( C(z) \) is even. Suppose we flip a bit \( z_k \) to make a new bitstring \( z' = (z_{n-1}, z_{n-2}, ..., z'_k, ..., z_0) \) where \( z'_k \neq z_k \) but the \( z_{j 
eq k} \) are the same. The change \( z_k \rightarrow z'_k \) will change the value of each \( C_{(j,k)} \), from Eq. (19) that is associated with \( z_k \), so that if \( C_{(j,k)}(z) = 1 \) then \( C_{(j,k)}(z') = 0 \) and vice-versa.

The total number of edge terms \( C_{(j,k)} \) can be separated into \( k \) terms that increase in value and \( \delta \) terms that decrease in value when \( z \rightarrow z' \). The value of the cost function for \( z' \) can then be expressed as \( C(z') = C(z) + k - \delta \). The number of edge components \( C_{(j,k)} \) that depend on \( z_k \) is even when each vertex degree is even, so \( \delta + k = 2m \) for some integer \( m \). This implies that \( k \) and \( \delta \) are either both even or both odd; either way, the difference \( k - \delta \) is even. Since \( C(z) \) is even by assumption, \( C(z') = C(z) + k - \delta \) is also even.

We have shown there is a bitstring \( z^{(0)} \) for which \( C(z^{(0)}) \) is even and we have shown changing any such bitstring gives a new \( C(z') \) which is also even. This implies \( C(z) \) is even for all \( z \) for every graph where each vertex degree is even. By Eq. (22) the angle components \( \gamma_q \) are periodic over intervals of \( \pi \) for these graphs.

1. Symmetry when all vertices have even degree

When all the vertices have even degree, the angle components \( \gamma_q \) are periodic over intervals of \( \pi \), as discussed around Eq. (20). To demonstrate equivalence of \( \langle \hat{C} \rangle \) in Eq. (27) for the angles \( \gamma' \) from Eq. (29) and \( \gamma \) from Eq. (28), in the next paragraph we show \( \hat{C} \) of Eq. (17) has only even eigenvalues when each vertex has even degree. Then \( C(z) = 2m_z \) for all \( z \), where \( m_z \) is an integer. The periodicity over \( \pi \) follows since the matrix elements from the unitary operator \( \hat{U}(\hat{C}, \gamma_q) \) of Eq. (7) are invariant

2. Symmetry when all vertices have odd degree

When all the vertices have odd degree, there is a symmetry Eq. (24) for angle-pairs \( (\gamma, \beta) \) and \( (\gamma', \beta') \) from Eqs. (28) and (30) respectively. Only graphs with an even number of vertices can have this symmetry since it is impossible to have a graph with an odd number of vertices where every vertex has odd degree. We prove the symmetry holds in a simple case with \( p = 1 \), the extension to \( p > 1 \) uses similar reasoning. We begin by considering the relations between the unitary operators with \( (\gamma, \beta) \) and \( (\gamma', \beta') \), then use the analysis to relate...
the time evolution and probabilities $P(z)$ for states with the different angles.

Let $\gamma'_1 = \gamma_1 \pm \pi$. The unitary operator $\hat{U}(\hat{C}, \gamma_1)$ from Eq. (7) is a product of unitary-operator components for each edge, a single component can be expressed as

$$\exp(-i \gamma_1 \hat{C}_{j,k}) = \exp\left(-i \frac{\gamma_1}{2} (\hat{\gamma} - \hat{Z}_j \hat{Z}_k) \right)$$

$$= \exp\left(-i \frac{\gamma'_1 + \pi}{2}\right) \exp\left(i \frac{\gamma'_1 + \pi}{2} \hat{Z}_j \hat{Z}_k \right). \quad (33)$$

Separate out a term $\exp\left(\mp i(\pi/2) \hat{Z}_j \hat{Z}_k \right)$ to obtain

$$\exp(-i \gamma_1 \hat{C}_{j,k}) = e^{i\delta} \hat{Z}_j \hat{Z}_k \exp(-i \gamma'_1 \hat{C}_{j,k}) \quad (34)$$

where $e^{i\delta}$ is an overall phase.

Now consider the unitary operator $\hat{U}(\hat{C}, \gamma_1)$ of Eq. (7). Using Eq. (34) we have

$$\hat{U}(\hat{C}, \gamma_1) = \prod_{(j,k)} \exp(-i \gamma'_1 \hat{C}_{j,k})$$

$$= e^{i\eta} \left( \prod_{j=0}^{n-1} \hat{Z}_j \right)^{n-1} \hat{U}(\hat{C}, \gamma'_1), \quad (35)$$

where $e^{i\eta}$ is a phase factor. The term $\prod_{j=0}^{n-1} \hat{Z}_j$ comes from the product of $\hat{Z}_j \hat{Z}_k$ for all the edges—each $\hat{Z}$ is raised to an odd power in the product since each vertex degree is odd and using $(\hat{Z}_j)^2 = \hat{1}$ reduces this to $\prod_{j=0}^{n-1} \hat{Z}_j$.

We will use Eq. (35) to simplify the time evolution of $|\psi_p(\gamma_1, \beta_1)\rangle$ in Eq. (4). This includes the unitary operator $\hat{U}(\hat{B}, \beta_1)$ from Eq. (8), which shows up in the product $\hat{U}(\hat{B}, \beta_1)\hat{U}(\hat{C}, \gamma_1)$. Our next goal will be to use commutation relations to move the $\prod_{j=0}^{n-1} \hat{Z}_j$ to the left side of the product.

Express the $\hat{U}(\hat{B}, \beta_1)$ from Eq. (8) as

$$\hat{U}(\hat{B}, \beta_1) = \prod_{j=0}^{n-1} \left( \cos(\beta_1) \hat{1} - i \sin(\beta_1) \hat{X}_j \right). \quad (36)$$

Now consider the product of the unitary operators

$$\hat{U}(\hat{B}, \beta_1)\hat{U}(\hat{C}, \gamma_1)$$

$$= e^{i\eta} \prod_{j=0}^{n-1} \left( \cos(\beta_1) \hat{1} - i \sin(\beta_1) \hat{X}_j \right) \left( \prod_{j=0}^{n-1} \hat{Z}_j \right) \hat{U}(\hat{C}, \gamma'_1) \quad (37)$$

The Pauli operators anticommute so

$$\hat{U}(\hat{B}, \beta_1)\hat{U}(\hat{C}, \gamma_1) = e^{i\eta} \left( \prod_{j=0}^{n-1} \hat{Z}_j \right) \hat{U}(\hat{B}, -\beta_1)\hat{U}(\hat{C}, \gamma'_1). \quad (38)$$

We are now ready to show the angles $(\gamma_1, \beta_1)$ and $(\gamma'_1, \beta'_1)$ give equivalent $(\hat{C})$ in Eq. (11), where $\gamma'_1 = \gamma_1 \pm \pi$ and $\beta'_1 = -\beta_1$. To demonstrate the equivalence we show the computational basis state probabilities $P(z)$ are the same for both $(\gamma_1, \beta_1)$ and $(\gamma'_1, \beta'_1)$.

The probability amplitude for a basis state $|z\rangle$ is

$$\langle z|\psi(\gamma_1, \beta_1)\rangle = \langle z|\hat{U}(\hat{B}, \beta_1)\hat{U}(\hat{C}, \gamma_1)|\psi_0\rangle \quad (39)$$

Using Eq. (38) this can be expressed as

$$\langle z|\psi(\gamma_1, \beta_1)\rangle = \langle z| e^{i\eta} \left( \prod_{j=0}^{n-1} \hat{Z}_j \right) \hat{U}(\hat{B}, \beta'_1)\hat{U}(\hat{C}, \gamma'_1)|\psi_0\rangle$$

$$= e^{i\eta} \langle z| \left( \prod_{j=0}^{n-1} \hat{Z}_j \right) |\psi(\gamma'_1, \beta'_1)\rangle \quad (40)$$

Squaring the amplitudes we obtain

$$|\langle z|\psi(\gamma_1, \beta_1)\rangle|^2 =$$

$$\langle \psi(\gamma_1, \beta'_1) | \left( \prod_{j=0}^{n-1} \hat{Z}_j \right) \langle z| \left( \prod_{j=0}^{n-1} \hat{Z}_j \right) |\psi(\gamma'_1, \beta'_1)\rangle | \quad (41)$$

The $\left( \prod_{j=0}^{n-1} \hat{Z}_j \right)$ can be moved to the left since it commutes with $|z\rangle\langle z|$, then using $\left( \prod_{j=0}^{n-1} \hat{Z}_j \right)^2 = \hat{1}$ gives

$$P(z) = |\langle z|\psi(\gamma_1, \beta_1)\rangle|^2 = |\langle z|\psi(\gamma'_1, \beta'_1)\rangle|^2, \quad (42)$$

where $P(z)$ is the probability of $|z\rangle$ from Eq. (10). The $P(z)$ are the same for $(\gamma_1, \beta_1)$ and $(\gamma'_1, \beta'_1)$ so $\langle \hat{C}(\gamma_1, \beta_1) \rangle = \langle \hat{C}(\gamma'_1, \beta'_1) \rangle$ in Eq. (11), which is the desired symmetry.

3. Time-reversal symmetry

We finally consider the “time-reversal” symmetry with the angles of Eqs. (28) and (31) in the symmetry relation Eq. (27) [4]. The symmetry is related to structures of the $\hat{B}$ and $\hat{C}$ operators, which have real-valued matrix elements in the computational basis in Eqs. (9) and (10), and the structure of the initial state $|\psi_0\rangle$, which has real-valued coefficients in the computational basis in Eq. (5). To demonstrate the symmetry we calculate generic computational basis probabilities $P(z)$ for states with both
sets of angles and show they are equal, thus the $\langle \hat{C} \rangle$ are equal following Eq. (11).

Consider the probability amplitude $\langle z | \psi_p (\gamma, \beta) \rangle$ for a single basis state $| z \rangle$. From Eqs. (11)-(13) this is

$$
\langle z | \psi_p (\gamma, \beta) \rangle = \frac{1}{\sqrt{2^n}} \sum_{z'} (\prod_{q=1}^{p} e^{-i \beta_q B e^{-i \gamma_q C}}) | z' \rangle.
$$

(43)

Taking the complex conjugate of Eq. (13) only changes the sign of the angle terms since the $\hat{B}$ and $\hat{C}$ matrices have real-valued matrix elements in the computational basis, thus

$$
\langle z | \psi_p (\gamma, \beta) \rangle^* = \langle z | \psi_p (-\gamma, -\beta) \rangle. \tag{44}
$$

The Born probabilities $P(z)$ from Eq. (10) are the same for both states since

$$
P(z) = |\langle z | \psi_p (\gamma, \beta) \rangle|^2 = |\langle z | \psi_p (-\gamma, -\beta) \rangle|^2,
$$

(45)

so the $\langle \hat{C} \rangle$ are the same in Eq. (14).

[1] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. Nature, 574(7779):505–510, 2019.

[2] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, Peng Hu, Xiao-Yang Yang, Wei-Jun Zhang, Hao Li, Yuxuan Li, Xiao Jiang, Lin Gan, Guang-wen Yang, Lixing You, Zhen Wang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. Quantum computing and approximate optimization. Scientific Reports, 10:021067, 2020.

[3] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028, 2014.

[4] Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G Rieffel. Quantum approximate optimization algorithm for MaxCut: A fermionic view. Physical Review A, 97(2):022304, 2018.

[5] Stuart Hadfield. Quantum algorithms for scientific computing and approximate optimization. arXiv preprint arXiv:1805.03265, 2018. Eq. 5.10, p. 114.

[6] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. Phys. Rev. X, 10:021067, 2020.

[7] G G Guerreschi and A Y. Matsuura. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. Scientific Reports, 9, 2019.

[8] Matija Medvidović and Giuseppe Carleo. Classical variational simulation of the quantum approximate optimization algorithm. arXiv preprint arXiv:2009:01760v1, 2020.

[9] Fernando G. S. L. Brandão, Michael Broughton, Edward Farhi, Sam Gutmann, and Hartmut Neven. For fixed control parameters the quantum approximate optimization algorithm’s objective function value concentrates for typical instances. arXiv preprint arXiv:1812.04170, 2018.

[10] Jonathan Wurtz and Peter Love. Bounds on MAX-CUT QAOA performance for $p > 1$. arXiv preprint arXiv:2010.11920, 2020.

[11] R. Shaydulin and Y. Alexeev. Evaluating quantum approximate optimization algorithm: A case study. In 2019 Tenth International Green and Sustainable Computing Conference (IGSC), pages 1–6, 2019.

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

[13] Ruslan Shaydulin, Stuart Hadfield, Tad Hogg, and Ilya Safro. Classical symmetries and QAOA. arXiv preprint arXiv:2012.04713, 2020.

[14] James Ostrowski, Rebekah Herrman, Travis S. Humble, and George Siopsis. Lower bounds on circuit depth of the quantum approximate optimization algorithm. arXiv preprint arXiv:2008.01820v2, 2020.

[15] V. Akshay, H. Philathong, M. E. S. Morales, and J. D. Biamonte. Reachability deficits in quantum approximate optimization. Phys. Rev. Lett., 124:090504, 2020.

[16] Mario Szegedy. What do QAOA energies reveal about graphs? arXiv preprint arXiv:1912.12272v2, 2020.

[17] Guido Pagano, Aniruddha Bapat, Patrick Becker, Katherine S. Collins, Arinjoy De, Paul W. Hess, Harvey B. Kaplan, Antonis Kyprianidis, Wen Lin Tan, Christopher Baldwin, Lucas T. Brady, Abhinav Deshpande, Fangli Liu, Stephen Jordan, Alexey V. Gorshkov, and Christopher Monroe. Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator. Proceedings of the National Academy of Sciences, 117(41):25396–25401, 2020.

[18] Zhihui Wang, Nicholas C Rubin, Jason M Dominy, and Eleanor G. Rieffel. XY-mixers: analytical and numerical results for the quantum alternating operator ansatz. Physical Review A, 101:012320.

[19] Linghua Zhu, Ho Lun Tang, George S. Barron, Nicholas J. Mayhall, Edwin Barnes, and Sophia E. Economou. An adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. arXiv preprint arXiv:2005.10258, 2020.

[20] Zhang Jiang, Eleanor G. Rieffel, and Zhihui Wang. Near-optimal quantum circuit for Grover’s unstructured search using a transverse field. Physical Review A, 95:062317, 2017.

[21] Andreas Bärtscihi and Stephan Eidenbenz. Grover mixers for QAOA: Shifting complexity from mixer design to state preparation. arXiv preprint arXiv:2006.00354v2, 2020.

[22] Jeremy Cook, Stephan Eidenbenz, and Andreas Bärtscihi. The quantum alternating operator ansatz on maximum $k$-vertex cover. arXiv preprint arXiv:1910.13483v2, 2020.
[23] Li Li, Minjie Fan, Marc Coram, Patrick Riley, and Stefan Leichenauer. Quantum optimization with a novel Gibbs objective function and ansatz architecture search. *Phys. Rev. Research*, 2, 2020.

[24] Reuben Tate, Majid Farhadi, Creston Herold, Greg Mohler, and Swati Gupta. Bridging classical and quantum with SDP initialized warm-starts for QAOA. *arXiv preprint arXiv:2010.14021*, 2020.

[25] William H. Press, Brian P. Flannery, and Saul A. Teukolsky. *Numerical Recipes in Fortran 77: The Art of Scientific Computing*. Cambridge University Press, second edition, 1993. https://people.sc.fsu.edu/~inavon/5420a/DFP.pdf.

[26] Phillip C. Lotshaw and Travis S. Humble. QAOA dataset. Found at https://code.ornl.gov/qci/qaoa-dataset-version1.

[27] Rebekah Herrman, Lorna Treffert, James Ostrowski, Phillip C. Lotshaw, Travis S. Humble, and George Siopsis. Impact of graph structures for QAOA on MaxCut. *arXiv preprint arXiv:2102.05997*, 2021.

[28] Brendan McKay. Graphs. https://users.cecs.anu.edu.au/~bdm/data/graphs.html. Accessed July 8, 2020.

[29] Pietro Belotti. Couenne: A user’s manual. Technical report, Technical report, Lehigh University, 2009.

[30] Pietro Belotti, Jon Lee, Leo Liberti, Francois Margot, and Andreas Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods & Software*, 24(4-5):597–634, 2009.

[31] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the Association for Computing Machinery*, 42(6):1115–1145, 1995.