

## Parameter-setting heuristic for the quantum alternating operator ansatz

James Sud<sup>1,2,3,\*</sup>, Stuart Hadfield<sup>1,2</sup>, Eleanor Rieffel<sup>1</sup>, Norm Tubman<sup>1</sup>, and Tad Hogg<sup>1</sup>

<sup>1</sup>Quantum Artificial Intelligence Laboratory (QuAIL), NASA Ames Research Center, Moffett Field, California 94035, USA

<sup>2</sup>USRA Research Institute for Advanced Computer Science (RIACS), Mountain View, California 94043, USA

<sup>3</sup>Department of Computer Science, University of Chicago, 5730 S Ellis Ave, Chicago, Illinois 60637, USA



(Received 6 February 2023; accepted 1 December 2023; published 15 May 2024)

The quantum alternating operator ansatz (QAOA) is a generalized approach for solving challenging optimization problems that builds on the alternating structure of the quantum approximate optimization algorithm. Finding high-quality parameters efficiently for QAOA remains a major challenge in practice. In this work, we introduce a classical strategy for parameter setting, suitable for cases in which the number of distinct cost values grows only polynomially with the problem size, such as is common for constraint-satisfaction problems. The crux of our strategy is that we replace the cost function expectation value step of QAOA with a classical model that can be efficiently evaluated classically and has parameters which play an analogous role to the QAOA parameters. This model is based on empirical observations that, in some QAOA states, variable configurations with the same cost have the same amplitudes from step to step. We define this class of states as homogeneous states. For problems with particular symmetries, QAOA states are guaranteed to be homogeneous. More generally, high overlaps between QAOA states and homogeneous states have been empirically observed in a number of settings. Building on this idea, we define a classical homogeneous proxy for QAOA in which this property holds exactly and which yields information describing both states and expectation values. We then classically determine high-quality parameters for this proxy and then use these parameters in QAOA, an approach we label the homogeneous heuristic for parameter setting. We numerically examine this heuristic for MaxCut on unweighted Erdős-Rényi random graphs. For up to three QAOA levels we demonstrate that the heuristic is easily able to find parameters that match approximation ratios corresponding to previously found globally optimized approaches. For levels up to 20 we obtain parameters using our heuristic with approximation ratios monotonically increasing with depth, while a strategy that uses parameter transfer instead fails to converge with comparable classical resources. These results suggest that our heuristic may find good parameters in regimes that are intractable with noisy intermediate-scale quantum devices. Finally, we outline how our heuristic may be applied to wider classes of problems.

DOI: [10.1103/PhysRevResearch.6.023171](https://doi.org/10.1103/PhysRevResearch.6.023171)

### I. INTRODUCTION

The quantum alternating operator ansatz (QAOA) [1] is a widely studied parametrized quantum algorithm for tackling combinatorial optimization problems. It uses the alternating structure of Farhi *et al.*'s quantum approximate optimization algorithm [2], a structure that was also studied in special cases in much earlier work [3,4]. Recent work has explored regimes for which suitable parameters for QAOA are difficult to precompute. This includes extensive research viewing QAOA as a variational quantum-classical algorithm (VQA), in which results from runs on quantum hardware are fed into a classical outer loop algorithm for updating the parameters. Optimizing parameters for VQAs can be quite challenging, as one

typically needs to search over a large parameter space with a complex cost landscape. While, for a successful algorithm, one does not necessarily need to find optimal parameters, but rather good enough parameters for a given target outcome, finding good parameters can be difficult due to the number of local optima [5,6] and in some cases barren plateaus [7,8]. Moreover, the high levels of noise present on current quantum devices can exacerbate these challenges [9].

We propose an approach to QAOA parameter optimization that maps the QAOA circuit to a simpler classical model. The crux of our approach is that in the parameter objective function (as introduced below), we replace the cost function expectation value, which is typically evaluated either on quantum hardware or using expensive classical evaluation, with a simpler model that has parameters which play an analogous role to the QAOA parameters, but can be efficiently evaluated classically. This approach is based on the observation, originally due to Hogg [3,4], that one may leverage a classical “mean-field” or “homogeneous” model where variable configurations with the same cost have the same amplitudes from step to step. Extending this idea, we define a classical homogeneous proxy for QAOA (“proxy” for short) in which

\*jsud@uchicago.edu

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

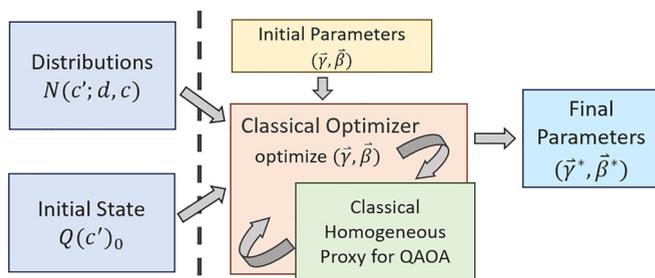


FIG. 1. Flowchart of parameter setting procedure using the classical homogeneous proxy for QAOA. The boxes to the left of the dashed line denote inputs that can be viewed as classical precomputations, as described in Sec. II. Given these inputs, along with an initial guess of parameters, one can perform parameter setting, in which a classical optimizer and the proxy are used in a loop to search for parameters for the algorithm. Here the proxy is used to estimate values of the cost function in order to reduce the time and space complexity for cost function evaluation, while the choice of classical optimizer and initial parameters is left open. In the end, the procedure outputs good parameters for the proxy that are then used as parameters for QAOA.

this property holds exactly and which yields both information describing states and expectation values. We note that this proxy is not a rigorous approximation to QAOA in general and does not yield energies guaranteed to be within some fixed error with respect to QAOA. Rather the proxy is theoretically motivated and numerically explored and we demonstrate cases where the proxy is empirically seen to provide energies close to that of QAOA. We then introduce the strategy of classically determined high-quality parameters for this proxy and, using these parameters in QAOA, an approach we label the homogeneous heuristic for parameter setting (“heuristic” for short). This heuristic is visualized in Fig. 1.

The heuristic is efficient for classes of constraint satisfaction problems (CSPs) for which the number of distinct cost function values is polynomial and we describe the proxy for four such classes of CSPs: random kSAT, MaxCut on unweighted Erdős-Rényi model graphs, random Max-EkLin2, and random Max-k-XOR. For these examples, the proxy leverages information only about the class, rather than problem instance, so that the proxy describes states and expectation values for the entire class. We then explore the heuristic for parameter setting on classes of MaxCut problems. This heuristic returns a set of parameters for the entire *class*, which can then be tested on instances from that class. Our results indicate that, for MaxCut, the heuristic on 20-node graphs is able to—for  $p = 1, 2,$  and  $3$ —identify parameters yielding approximation ratios comparable to those returned by the commonly used strategy of transferring globally optimized parameters [10, 11]. We then demonstrate out to depth  $p = 20$  that the heuristic identifies parameters yielding monotonically increasing approximation ratios as the depth of the algorithm increases. A parameter setting strategy that uses an identical outer loop parameter update scheme but does not leverage the proxy fails to converge given identical classical resources.

The practical implication of this work is that for classes of optimization problems, such as CSPs with a fixed number of randomly selected clauses on a fixed number of variables,

the classical homogeneous proxy for QAOA can be computed with solely classical resources in time polynomially scaling with respect to  $p$  as well as  $n$ . Thus the proxy avoids the issue of exponentially growing resources (with respect to  $p$  or  $n$ ) when using exact classical simulators and avoids the noise present on NISQ devices. The homogeneous heuristic for parameter setting leverages this proxy, so the subroutine evaluating cost expectation values may be much quicker. However, the parameter landscape may still be challenging to optimize over, especially in cases in which the number of free parameters grows with the problem size. Nevertheless, we demonstrate for MaxCut that our heuristic is able to outperform previous results of parameter transfer [10], indicating the potential of our heuristic approach (which may also be combined with other advancements in parameter setting, as discussed briefly later in the paper) to achieve further improvements in practice.

*Quantum alternating operator ansatz.* Here we briefly describe the structure of QAOA [1–4]. Given a cost function  $c(x)$  to optimize, a QAOA circuit consists of  $p$  repeated applications of a mixing and cost Hamiltonian  $B$  and  $C$  in alternation, parametrized by  $2p$  parameters  $(\vec{\gamma}, \vec{\beta})$ :

$$|\Psi(\vec{\gamma}, \vec{\beta})\rangle = e^{-i\beta_p B} e^{-i\gamma_p C} \dots e^{-i\beta_1 B} e^{-i\gamma_1 C} |\psi_0\rangle, \quad (1)$$

where  $|\psi_0\rangle$  is an easily prepared initial state. There is freedom in choosing the Hamiltonians  $B$  and  $C$ , although typically (as is followed in this work),  $C$  is chosen such that  $C|x\rangle = c(x)|x\rangle$  for all  $x$  and  $B$  is a mixer that is simple to implement on hardware. More general operators and initial states for QAOA are proposed in [1]. Throughout this work we analyze QAOA with the  $X$  mixer  $B = \sum_{i=1}^n X_i$ , where  $n$  represents the total number of qubits in the circuit, and the starting state  $|\psi_0\rangle$  is chosen to be the uniform superposition over all  $2^n$  bitstrings, as originally considered in [2]. The goal of QAOA is then to produce a state  $\Psi(\vec{\gamma}, \vec{\beta})$  such that repeated sampling in the computational basis yields either an optimal or high-quality approximate solution to the problem. Finding such good parameters is the parameter setting problem and may be approached in a number of ways with different tradeoffs, ranging from black-box optimization techniques to application specific approaches. We refer to fixed QAOA parameters  $p, \vec{\gamma}, \vec{\beta}$  as a *parameter schedule*.

We now describe QAOA as a VQA: given some classical cost function  $c(x)$  on  $n$  variable bitstrings  $\{0, 1\}^n$ , and a quantum circuit ansatz with parameters  $\vec{\Theta}$ , one defines a parameter objective function  $f(\vec{\Theta})$  and optimizes over  $\vec{\Theta}$  with respect to  $f$ . In order to optimize over  $\vec{\Theta}$ , a two-part cycle is typically employed. First, a classical outer loop chooses one or more parameters  $\vec{\Theta}$  for which evaluations of  $f(\vec{\Theta})$  or its partial derivatives are requested, based on initial or prior cycle information. Then, a subroutine evaluates  $f$  and its derivatives for all  $\vec{\Theta}$  requested by the outer loop. Given these evaluations, the parameter update scheme can then update the current best  $\vec{\Theta}$  and choose a new set of parameters to test. Throughout this work we refer to the outer loop as the parameter update scheme and the inner subroutine as parameter objective function evaluation. Typically in QAOA,  $f(\vec{\Theta}) = f(\vec{\gamma}, \vec{\beta})$  is taken to be  $\langle \vec{\gamma}, \vec{\beta} | C | \vec{\gamma}, \vec{\beta} \rangle$ . This choice of  $f$  measures the expected cost function value obtained from sampling the QAOA state,

which we refer to as the typical parameter objective function. In this work, we will replace it with what we define as the homogeneous parameter objective function, which utilizes the classical homogeneous proxy for QAOA.

*Related work.* There have been numerous parameter setting strategies proposed for QAOA. Most of these strategies focus on improving the parameter update scheme, while keeping the typical parameter objective function. These strategies range from the simplest approach of directly applying classical optimizers to approaches incorporating more sophisticated machine learning techniques [12–20]. In practice, however, efficiently finding high-quality parameters remains a challenging task. Global optimization strategies often get stuck in navigating the parameter optimization landscape due to barren plateaus [7–9] or multitudes of local optima [5,6] and the problem becomes especially challenging as the number of parameters grows due to the curse of dimensionality. In some circumstances parameter optimization can even become NP-hard [21]. There have been multiple methods proposed that attempt to alleviate these issues. The first of these include initializing parameters to be close to parameters informed by previous information or intuition [22–24]. One such example is to use parameters that represent a discretized linear quantum annealing schedule [25,26]. Another example is based on the principle that globally optimal parameters for random QAOA instances drawn from the same distribution tend to concentrate [2,13,27,28]. Thus, if globally optimal parameters are known for one or many instances of a specific class of problems, these exact parameters (or median parameters for more than one instance) empirically perform well on a new, unseen instance from the same class [11,29]. Another approach for improving parameter setting is reparametrizing the circuit, which places constraints on the allowed parameter schedules, thereby reducing the number of parameters so that they are optimized more easily. In some cases, this reparametrization can be informed by optimized parameters for typical instances or by connections to quantum annealing [13,23,24,26,30].

An alternative paradigm for parameter setting are methods that modify the parameter objective function itself. Indeed, one can find closed form expressions for expected cost as a function of parameters in some cases—for example, MaxCut at  $p = 1$  [31] or  $p = 2$  for high-girth regular graphs [32]. Moreover, when applicable one can take advantage of problem locality considerations (such as graphs of bounded vertex degree for MaxCut) to compute the necessary expectation values without requiring the full quantum state vector [2,31]. In these cases, then, one could optimize parameters with respect to these expressions rather than by evaluating the entire dynamics. Other examples of parameter objective function modification include using conditional value at risk [33] and Gibbs-like functions [34], which are closely related to the usual parameter objective function. Similar in spirit to our approach, recent work [35–37] has proposed the use of surrogate models, which use quantum circuit measurement outcomes to construct an approximate parameter objective function. In contrast, our approach is entirely classical and the parameters it outputs may be used directly, or possibly improved further, given access to a quantum device. Additionally, a related perspective was recently proposed in [38], wherein the authors study the connection between single-layer

QAOA states and pseudo-Boltzmann states where computational basis amplitudes are also expressed as functions of the corresponding costs (which we define as homogeneous in this work). While [38] provides additional motivation for our approach, the authors there do not consider cases beyond  $p = 1$  and so their results do not immediately apply to parameter setting in the same way.

*Outline of paper.* In Sec. II we define the classical homogeneous proxy for QAOA. In Sec. III, we demonstrate how to implement the proxy for classes of CSPs with a fixed, polynomial number of randomly drawn clauses. In Sec. IV we introduce the homogeneous heuristic for parameter setting. In Sec. V we provide numerical evidence for the efficacy of the proxy and the heuristic applied to MaxCut on unweighted Erdős-Rényi model graphs. Finally in Sec. VI we present the results of the heuristic for the MaxCut on the same class of graphs. We conclude in Sec. VII with a discussion of our results and several future research directions.

## II. CLASSICAL HOMOGENEOUS PROXY FOR QAOA

This section generalizes the mean-field approach of [3], updating the notation to match current notation for QAOA and allowing for easier extension to other CSPs. We describe the so-called classical homogeneous proxy for QAOA from a sum-of-paths perspective [3,39] using a set of classical equations that ensure that our intermediate states are *homogeneous*, as we define below.

*Definition 1. Homogeneous state.* Let  $|\Psi\rangle$  be a quantum state  $|\Psi\rangle = \sum_{x \in \{0,1\}^n} q(x) |x\rangle$ , where  $q(x)$  is the amplitude of bitstring  $x \in \{0,1\}^n$  when written in the computational basis, and  $c$  be a function from  $x \in \{0,1\}^n \rightarrow \mathcal{R}$ . Then  $|\Psi\rangle$  is homogeneous if and only if the amplitude  $q(x)$  of all  $x$  can be written solely as a function of  $c(x)$ , i.e.,  $Q(c(x))$ . Then  $|\Psi\rangle = \sum_{x \in \{0,1\}^n} Q(c(x)) |x\rangle$ .

Simply put, given a cost function  $c$ , homogeneous states are those such that bitstrings of the same cost have the same amplitudes. For example, QAOA states are homogeneous if the cost function is nondegenerate [where each bitstring  $x$  has a unique cost  $c(x)$ ], as well as the maximally degenerate case where the cost function is constant. Importantly, the standard QAOA initial state  $|\psi\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_y |y\rangle$  is homogeneous for all cost functions.

A less trivial example is the class of cost functions that depend only on the Hamming weight of each bitstring,  $c(x) = c(|x|)$ , such as the Hamming ramp  $c(x) = |x|$ . For this class, symmetry arguments [23] imply that the QAOA state is homogeneous for any choice of QAOA circuit depth and parameters. Indeed, useful intuition for homogeneity can be gained from the symmetry perspective; in [23, Lemma 1] it is shown that, for a classical cost function with symmetries compatible with the QAOA mixer and initial state, bitstrings that are connected by these symmetries will have identical QAOA amplitudes. A state being homogeneous is an even stronger condition: all bitstrings with the *same cost* have identical amplitudes, not just those connected by mixer-compatible symmetries. While homogeneity does not hold in general, it can be seen empirically from [3] and similar numerical experiments on CSPs that, under certain parameter assumptions, QAOA produces close to homogeneous states,

in that the amplitudes of bitstrings with identical cost have *similar* amplitudes. This observation motivates our classical proxy. Given this intuition, we begin from the sum-of-paths perspective [3,39] for QAOA to describe our proxy.

Additional intuition comes from the case of strictly  $k$ -local problem Hamiltonians (such as, for instance, MaxCut), where the QAOA state can easily be shown to be homogeneous to leading order in  $|\gamma|$ , independent of  $\beta$ , with amplitude of bitstring  $x$  given by  $\frac{1}{\sqrt{2^n}}[1 - i\gamma c(x)e^{i2k}]$  in the  $p = 1$  case. Similar analysis for the higher- $p$  case also yields homogeneous states to first order in the  $\gamma_j$  parameters. This connection is considered further in Sec. VB.

### A. Classical homogeneous proxy for QAOA from the sum-of-paths perspective for QAOA

In this section, we first consider the evolution of a single step of QAOA exactly. Exactly computing this step classically can require exponentially scaling resources to track all of the quantum amplitudes, so we introduce a series of approximations to relax the calculation into one that is of polynomial size in particular cases of interest. We name the altered evolution the classical homogeneous proxy to QAOA and defer analysis of the validity of these approximations to later sections.

We begin by exactly analyzing the  $\ell$ th step of QAOA using the sum-of-paths perspective described in [39, Appendix A]. Let  $q_{\ell-1}(x)$  denote the amplitude of a bitstring  $x$  after step  $\ell - 1$  steps of QAOA with parameters  $\gamma_1, \beta_1, \dots, \gamma_{\ell-1}, \beta_{\ell-1}$ . Applying an additional layer of QAOA with parameters  $\gamma_\ell, \beta_\ell$  [i.e., application of the operators  $e^{-\beta_\ell B} e^{-i\gamma_\ell C}$  from Eq. (1)] the amplitudes become

$$q_\ell(x) = \langle x | \vec{\gamma}, \vec{\beta} \rangle = \sum_y q_{\ell-1}(y) \cos^{n-d_{xy}} \beta (-i \sin \beta)^{d_{xy}} e^{-i\gamma c_y}, \quad (2)$$

where  $d_{xy}$  is the Hamming (bit-flip) distance between bitstrings  $x$  and  $y$ , the factors  $\langle x | e^{-i\beta B} | y \rangle = \cos \beta^{n-d_{xy}} (-i \sin \beta)^{d_{xy}}$  are the mixing operator matrix elements,  $c_y = c(y)$  is the cost of bitstring  $y$ , and the sum is over all possible bitstrings  $y \in \{0, 1\}^n$ .

Here, we see that the QAOA amplitude  $q_\ell(x)$  at step  $\ell$  results as a sum of contributions from the amplitudes at the previous step of all possible bit strings, weighted by factors that depend on the respective costs and Hamming distances.

Grouping the terms in Eq. (2), we can rewrite the sum as

$$q_\ell(x) = \sum_{d,c} \cos^{n-d} \beta (-i \sin \beta)^d e^{-i\gamma c} \sum_{y|d_{xy}=d, c_y=c} q_{\ell-1}(y), \quad (3)$$

where the sum over  $d$  runs from  $[0, n]$  and the sum over  $c$  runs over the set of unique costs allowed by the cost function, which we describe in Sec. IIB. If we assume that the QAOA state at step  $\ell$  is homogeneous, then it is the case that the amplitudes  $q_{\ell-1}(x)$  of all bitstrings with cost  $c'$  are identical, so we can substitute  $q(x)$  with  $Q(c(x))$  for all  $x$ . For the rest of this text we use this upper case  $Q(c(x))$  to denote that  $Q$  depends on  $c = c(x)$  rather than the bitstring  $x$  itself. This

substitution yields

$$q_\ell(x) \simeq \sum_{d,c} \cos^{n-d} \beta (-i \sin \beta)^d e^{-i\gamma c_y} Q_{\ell-1}(c) n(x; d, c), \quad (4)$$

where  $n(x; d, c)$  represents the number of bitstrings with cost  $c$  that are of Hamming distance  $d$  from  $x$ . Importantly, in this step we note that this replacement is motivated by empirical observations in QAOA and we do not claim its validity in general or the extent to which the replacement approximates QAOA; rather we provide empirical evidence of the utility of the replacement in Sec. V. We also note that for this equation, and all following equations, this evolution is no longer restricted to unitary evolution. As such, the values  $q$  and  $Q$  no longer represent strictly quantum amplitudes, but rather track an object that is an analog to the quantum amplitude under the proxy.

We now apply our final simplification to the sum-of-paths equation. Observe that  $n(x; d, c)$  are the only terms in the sum of Eq. (4) that depend explicitly on  $x$ . Hence if we can suitably approximate  $n(x; d, c)$  by a quantity that depends only  $c, d$ , this will yield a sum that depends on significantly fewer terms. To this end, we replace  $n(x; d, c)$  in the sum above with a distribution  $N(c'; d, c)$  with  $c' = c_x$ , where we again adopt the upper case  $N$  to indicate dependence only on bitstring costs. While this replacement may not be valid in general, in Sec. III we exhibit cases where  $N(c'; d, c)$  is efficiently computable and adequately replaces  $n(x; d, c)$  for the purpose of algorithm parameter setting. We emphasize that we do not make specific claims regarding the accuracy of this substitution for particular problem instances, but rather that it is motivated by empirical observations [3]; its effectiveness is further explored numerically in Secs. V and VI.

Hence, using  $N(c'; d, c)$  to replace  $n(x; d, c)$ , then, we can further rewrite the sum as

$$q_\ell(x) \simeq Q_\ell(c') = \sum_{d,c} \cos^{n-d} \beta (-i \sin \beta)^d e^{-i\gamma c} Q_{\ell-1}(c) \times N(c'; d, c), \quad (5)$$

where  $Q_\ell(c')$  is used to make explicit that the substitutions yield an expression for  $q_\ell(x)$  that depends solely on the cost  $c'$  of bitstring  $x$ . This leads to a crucial point for our analysis: if we start in a homogeneous state, and if the distributions  $n(x; d, c)$  can be replaced by distributions  $N(c'; d, c)$ , which solely depend on the cost  $c'$  of  $x$ , then subsequent layers ensure the state remains homogeneous, yielding analogs of amplitudes  $Q(c')$ . Thus, applying the same substitutions to  $Q_{\ell-1}, Q_{\ell-2}$ , and so on, Eq. (5) gives a recursive equation yielding homogeneous analogs of quantum states, which we label the classical homogeneous proxy for QAOA. Here we note that, importantly, the proxy will not return the analog of amplitude of some bitstring  $x$ , but rather the analog of amplitude of bitstrings with cost  $c'$ , as explicit knowledge of which bitstrings  $x$  have cost  $c'$  would solve the underlying objective function. In Sec. V we argue that there are cases where these analog of amplitudes  $Q(c')$  are close to all  $q(x)$  with  $c_x = c'$ .

At a high level, this section introduced a classical proxy for approximating QAOA dynamics such that the states are

homogeneous at each step. We next explain how to use the proxy to estimate important QAOA quantities such as expectation values of quantum observables.

**B. Cost distributions**

In order to evaluate Eq. (5), we first observe that the set of costs to which indices  $c$  and  $c'$  belong must be defined. Ideally, this set represents exactly the unique objective function values allowed by the underlying cost function. For optimization problems, however, the range of this set is not known exactly and moreover is precisely what we would like to solve. Instead, we can form a reasonable estimate by determining upper and lower bounds on the cost function values. We denote this estimated set of unique costs as  $\mathcal{C}$ . As an example, for CSPs with binary-valued clauses, the cost function counts the number of satisfied clauses, trivially yielding  $\mathcal{C} = \{0, 1, \dots, m\}$ , where  $m$  is the number of clauses. In practice smaller sets may be obtained using problem instance-specific information. The set  $\mathcal{C}$  also allows us to define the initial state  $Q_0(c')$  for the algorithm. QAOA typically begins with a uniform superposition over all bitstrings  $x$ , such that  $q_0(x) = \frac{1}{\sqrt{2^n}}$  for all  $x$ . Thus we can set  $Q_0(c') = \frac{1}{\sqrt{2^n}}$  for all  $c'$  in  $\mathcal{C}$ .

While Eq. (5) yields analogs of amplitudes  $Q_\ell(c')$ , one may also wish to use the classical homogeneous proxy for QAOA to return an estimate of expected value of the cost function. This requires computing a distribution  $P(c')$  for all  $c'$  in  $\mathcal{C}$ , representing the probability a randomly chosen bitstring has cost  $c'$ , averaged over the entire class. Much like in the case of computing  $\mathcal{C}$ , this computation is approximate, as a perfect computation of this distribution would solve the underlying objective function. Examples of estimations of  $P(c')$  are shown in Sec. III. In order to compute our estimate of expected value of the cost, then, we simply sum over costs  $c'$ , weighting by the expected number of bitstrings with cost  $c'$  [ $2^n P(c')$ ] and the squared analog to the amplitude  $|Q_\ell(c')|^2$ ,

$$\widetilde{\langle C \rangle} = \sum_{c'} 2^n P(c') |Q_\ell(c')|^2 c', \tag{6}$$

with the tilde indicating the use of the proxy and that this is no longer a strictly normalized quantum expectation value. This is exactly the equation we use for the homogeneous parameter objective function as described in the Introduction. Note that  $P(c')$  does not give a *bona fide* probability distribution unless it is renormalized by dividing by  $\sum_{c'} 2^n P(c') |Q_\ell(c')|^2$  at each step; nevertheless, we have found these factors remain close to unity for the parameter-setting experiments considered below and so we neglect them going forward, which yields further computational savings. It is straightforward to introduce these factors into Eq. (6) if desired in other applications.

The set  $\mathcal{C}$  and estimate to cost  $\widetilde{\langle C \rangle}$  are critically determined by the class of problems one is working with. Examples of these values for a sample of classes are given in Sec. III.

**C. Algorithm for computing the classical homogeneous proxy for QAOA**

Formalizing the results of the section, we now present Algorithm 1, which describes how, given QAOA parameters

**Algorithm 1.** Classical Homogeneous Proxy for QAOA.

---

**Input:**  $p, \vec{\gamma}, \vec{\beta}, n, m, \mathcal{C}, N(c'; d, c) \forall c' \in \mathcal{C}, P(c') \forall c' \in \mathcal{C}$ .  
**Output:**  $Q_p(c') \forall c' \in \mathcal{C}$ , Optionally  $\langle C(\vec{\gamma}, \vec{\beta}) \rangle_h$   
 $Q_0(c') \leftarrow \frac{1}{\sqrt{2^n}} \forall c' \in \mathcal{C}$   
**for**  $\ell$  in  $[1, p]$  **do**  
     $Q_\ell(c') \leftarrow \sum_{d,c} \cos \beta^{n-d} (-i \sin \beta)^d e^{-i\gamma c} Q_{\ell-1}(c) N(c'; d, c) \forall c' \in \mathcal{C}$   
**end for**  
If desired,  $\widetilde{\langle C \rangle} \leftarrow \sum_{c'} 2^n P(c') |Q_\ell(c')|^2 c'$

---

$p, \vec{\gamma}, \vec{\beta}$ , a set of possible costs  $\mathcal{C}$ , and assumed cost distribution  $N(c'; d, c)$ , we can compute the classical homogeneous proxy for QAOA using Eq. (5) and Eq. (6).

At a high level, this algorithm says, for each step  $\ell$  from 1 to  $p$ , to apply a layer of the classical homogeneous proxy to QAOA from Eq. (5), starting from the initial state described in Sec. II B.

*Time complexity.* Given  $N(c'; d, c)$  as well as  $Q_{\ell-1}(c')$  for all  $c'$  in  $\mathcal{C}$ , the calculation of each amplitude  $Q_\ell(c')$  using Eq. (5) is  $\mathcal{O}(n|\mathcal{C}|)$ . Thus the calculation of all  $|\mathcal{C}|$  amplitudes  $Q_\ell(c')$  is  $\mathcal{O}(n|\mathcal{C}|^2)$ . Computing all  $Q_p(c')$  elements then is  $\mathcal{O}(np|\mathcal{C}|^2)$ . If desired, the evaluation of the approximate cost expectation value, given in Eq. (6), involves a sum over  $\mathcal{C}$  elements and thus is  $\mathcal{O}(|\mathcal{C}|)$ .

Thus, if  $|\mathcal{C}|$  is  $\mathcal{O}(\text{poly}(n))$ , then Algorithm 1 is efficient. Simply, if there are a polynomial number of possible costs allowed by the cost function, the proxy is efficient. Indeed, we show such a case in the following section, demonstrating how to calculate the necessary precomputations of  $\mathcal{C}$ , as well as  $N(c'; d, c)$  and  $P(c')$  for all  $c'$  in  $\mathcal{C}$ .

**III. COST DISTRIBUTIONS FOR RANDOMLY DRAWN CSPS**

In this section we demonstrate several classes of CSPs where we can efficiently derive and implement the classical homogeneous proxy to QAOA as described in Sec. II A. In particular, we look at classes of CSPs where clauses are drawn uniformly at random from some fixed distribution. For these classes, we show that the distributions  $N(c'; d, c)$  necessary in the proxy can be computed in polynomial time and these distributions can reasonably replace the exact distributions for each instance. The question of how reasonable the replacements are is left to Sec. V.

Particularly, we focus on CSPs with a fixed number  $m$  of Boolean clauses, each acting on  $k$  variables selected at random from the set of  $n$  variables. The types of allowed clauses is determined by the problem, for example, SAT problems consider disjunctive clauses. We note that the analysis here generalizes a similar method in [3] applied to 3-SAT, allowing for easy extension to any CSP matching the above criteria. For these problems, the procedure is as follows, noting that *all calculations done here are averaged over the entire class*.

First, we observe that, assuming clauses are drawn uniformly at random, we can rewrite the expected number of bitstrings with cost  $c$  at distance  $d$  from a random bitstring

**Algorithm 2.** Classical Homogeneous Proxy for QAOA Pre-computation for Randomly Drawn Cost Function.

**Input:**  $n, m$ , Description of problem class

**Output:**  $C, N(c'; d, c), P(c') \forall c' \in C$

Determine  $C$  via Sec. II B.

Compute  $P_{\text{both}}, P_{\text{one}}, P_{\text{neither}}$ , and  $P(c') \forall c' \in C$  given the problem class

Use these values to compute  $N(c')$  and  $N(c'; d, c) \forall c' \in C$  via  $2^n P(c')$  and Eq. (11)

of cost  $c'$  as

$$N(c'; d, c) = \binom{n}{d} P(c|d, c'), \quad (7)$$

where  $P(c|d, c')$  represents the probability that a bitstring at distance  $d$  from a bitstring with cost  $c'$  has cost  $c$ . This probability can then be rewritten as

$$P(c|d, c') = \frac{P(c', c|d)}{P(c')}, \quad (8)$$

where  $P(c', c|d)$  represents the probability that two bitstrings separated with Hamming distance  $d$  have costs  $c$  and  $c'$  and  $P(c')$  represents the probability that a randomly chosen bitstring has cost  $c'$ . The numerator can be calculated as follows:

$$P(c', c|d) = \sum_{b=\max(0, c'+c-m)}^{\min(c', c)} P(b, c' - b, c - b|d), \quad (9)$$

where  $P(b, c' - b, c - b|d)$  represents the probability that two randomly chosen bitstrings with Hamming distance  $d$  have  $b$  satisfied clauses in common, along with cost  $c'$  and  $c$ . This expression can be evaluated via the multinomial distribution

$$P(b, c' - b, c - b|d) = \frac{m!}{b!(c' - b)!(c - b)![m + b - (c' + c)]!} \times P_{\text{both}}^b P_{\text{one}}^{c'+c-2b} P_{\text{neither}}^{m+b-(c'+c)}, \quad (10)$$

where  $P_{\text{both}}, P_{\text{one}},$  and  $P_{\text{neither}}$  represent the probability that a randomly selected clause is satisfied by both, one, or neither of the bitstrings separated by Hamming distance  $d$ , respectively. Since  $P_{\text{both}} + 2P_{\text{one}} + P_{\text{neither}} = 1$ , one only needs to calculate two of these three variables. The previous equations then allow computing  $N(c'; d, c)$  for any value  $c'$  as

$$N(c'; d, c) = \binom{n}{d} \frac{1}{P(c')} \sum_{b=\max(0, c'+c-m)}^{\min(c', c)} \frac{m!}{b!(c' - b)!(c - b)![m + b - (c' + c)]!} \times P_{\text{both}}^b P_{\text{one}}^{c'+c-2b} P_{\text{neither}}^{m+b-(c'+c)}. \quad (11)$$

We summarize this approach in Algorithm 2.

*Time complexity.* There are  $|\mathcal{C}|$  distributions  $N(c'; d, c)$  with  $(n + 1)|\mathcal{C}|$  elements each. For fixed  $c', d,$  and  $c$ , we must sum over  $|\mathcal{C}|$  terms that can be evaluated in  $\mathcal{O}(|\mathcal{C}|)$  steps, given by the time complexity of evaluating factorials of  $\mathcal{O}(|\mathcal{C}|)$ . Thus the evaluation of all distributions is  $\mathcal{O}(n|\mathcal{C}|^4)$ .

We once again note that if  $|\mathcal{C}|$  is  $\mathcal{O}(\text{poly}(n))$ , Algorithm 2 runs in polynomial time.

We now demonstrate Algorithm 2 for several common problem classes.

### A. MaxCut

We first analyze MaxCut on Erdős-Rényi random graphs  $\mathcal{G}(n, p_e)$ . In this model, a graph  $G$  is generated by scanning over all possible  $\binom{n}{2}$  edges in an  $n$  node graph and including each edge with independent probability  $p_e$ . The MaxCut problem on  $G$  is to partition the  $n$  nodes into two sets such that the number of cut edges crossing the partition is maximized. With respect to the class  $\mathcal{G}(n, p_e)$  of graphs the cost function to be maximized over configurations  $x$  is

$$c(x) = \sum_{(i,j) \in \binom{n}{2}} e_{ij} x_i \oplus x_j, \quad (12)$$

where  $e_{ij}$  are independent binary random variables that take on value 0 with probability  $1 - p_e$  and 1 with probability  $p_e$ . We use this cost function to evaluate the relevant distributions in Eq. (11). We start by noting that for a fixed graph  $G$  with  $m$  edges, we have  $0 \leq c(x) \leq m$ , and the expected number of edges  $\mathbb{E}[m]$  in graphs drawn from  $\mathcal{G}(n, p_e)$  is  $p_e \binom{n}{2}$ , with a standard deviation proportional to the square root of this value as determined by the binomial distribution. Hence, as  $n$  becomes large, the expected number of edges concentrates to  $p_e \binom{n}{2}$  and so for simplicity in the remainder of this work we set  $\mathcal{C} = \{0, 1, \dots, \lceil \mathbb{E}[m] \rceil\}$ . Note that, in practice, to accommodate the possibility of a given instance with maximum cut greater than this quantity, one may increase  $\mathcal{C}$  by several standard deviations as desired.

We can then easily calculate  $P(c')$ , which is the probability a random bitstring has cost  $c'$  for Eq. (12). For a bitstring drawn uniformly at random, the probability of satisfying any given clause  $x_i \oplus x_j$  is  $1/2$ , as there are two satisfying assignments (01 and 10) and two nonsatisfying assignments (00 and 11). Thus the probability  $P(c')$  also follows a binomial distribution and is simply

$$P(c') = \binom{1}{2}^m \binom{m}{c'}. \quad (13)$$

We can then calculate  $P_{\text{both}}, P_{\text{one}},$  and  $P_{\text{neither}}$ , where we show all three for didactic purposes (since  $P_{\text{both}} + 2P_{\text{one}} + P_{\text{neither}} = 1$ ). To see this, consider two randomly chosen bitstrings  $y$  and  $z$ , separated by Hamming distance  $d$ . We note that there are  $(n - d)$  bits in common between  $y$  and  $z$  and  $d$  bits different. Thus  $y_i \oplus y_j = z_i \oplus z_j$  requires that either both  $i$  and  $j$  are from the set of  $(n - d)$  same bits or the set of  $d$  different bits, which has probability

$$P_{\text{same}} = \frac{\binom{n-d}{2}}{\binom{n}{2}} + \frac{\binom{d}{2}}{\binom{n}{2}}. \quad (14)$$

From this expression, we can easily see that  $P_{\text{both}} = P_{\text{neither}} = \frac{1}{2} P_{\text{same}}$ . Since  $P_{\text{same}}$  represents the probability that

$y_i \oplus y_j = z_i \oplus z_j$ , for a random clause and a random bitstring, the probability that  $y_i \oplus y_j = 1$  is  $1/2$  and the same is true for  $y_i \oplus y_j = 0$  by symmetry. Thus we have

$$P_{\text{both}} = P_{\text{neither}} = \frac{1}{2} \left( \binom{n-d}{2} + \binom{d}{2} \right). \quad (15)$$

For completion, we can calculate  $P_{\text{one}}$ , which is  $(1 - P_{\text{same}} - P_{\text{both}})/2$  as shown in Sec. II A. For this term we need  $y_i \oplus y_j \neq z_i \oplus z_j$ , which can be accomplished if one of  $i$  or  $j$  is chosen from the  $(n-d)$  bits in common and the other is chosen from the  $d$  differing bits. This probability is

$$P_{\text{dif}} = \frac{\binom{n-d}{1} \binom{d}{1}}{\binom{n}{2}}. \quad (16)$$

Thus  $P_{\text{one}}$ , which is the probability of specifically  $y$  satisfying and  $z$  not satisfying (or vice versa), is half of  $P_{\text{dif}}$ , so

$$P_{\text{one}} = \frac{1}{2} \left( \frac{\binom{n-d}{1} \binom{d}{1}}{\binom{n}{2}} \right). \quad (17)$$

Using these quantities  $N(c'; d, c)$  is then obtained from Eq. (11).

### B. MaxE3Lin2/Max-3-XOR

We next consider the MaxE3Lin2 problem which generalizes MaxCut. QAOA for MaxE3Lin2 was analyzed by Farhi *et al.* [2], who showed an advantage over classical approximation algorithms, only to inspire better classical approaches [40,41]. The analogous random class of MaxE3Lin2 problems has cost function

$$c(x) = \sum_{a < b < c} d_{abc} (x_a \oplus x_b \oplus x_c \oplus z_{abc}), \quad (18)$$

where the  $z_{abc}$ ,  $d_{abc}$  are independent random variables with equal probability of being 0 or 1.

Using Eq. (18) we can again calculate the relevant probability distributions. First note that a random bitstring  $x$  will satisfy an individual clause (i.e., term in the sum) with probability  $1/2$ , as  $(x_a + x_b + x_c) \bmod 2$  has an equal probability to be 0 or 1. Thus

$$P(c') = \left( \frac{1}{2} \right)^m \binom{m}{c'}, \quad (19)$$

using the binomial distribution. Then we note that, as in Sec. III A, the probability of  $(y_a + y_b + y_c) \bmod 2 = (z_a + z_b + z_c) \bmod 2$  for two random bitstrings with  $d(x, y) = d$  is given by

$$P_{\text{same}} = \frac{\binom{n-d}{3} + \binom{d}{2} \binom{n-d}{1}}{\binom{n}{3}}, \quad (20)$$

since the value of  $(x_a + x_b + x_c) \bmod 2$  is preserved if none or two of  $x_a, x_b, x_c$  are flipped, which is satisfied if  $a, b, c$  are

all from the  $n-d$  identical bits, or two of  $a, b, c$  are chosen from the  $d$  differing bits. Thus we can easily compute  $P_{\text{both}} = P_{\text{neither}} = P_{\text{same}}/2$ , since there is an equal chance  $(y_a + y_b + y_c) \bmod 2 = (z_a + z_b + z_c) \bmod 2 = 0/1$ .  $P_{\text{one}}$  can then be calculated by a similar argument or by taking  $P_{\text{one}} = (1 - P_{\text{both}} - P_{\text{neither}})/2$ . We note that the probability distributions calculated here are exactly equivalent to those for the Max-3-XOR problem, where all  $z_{abc}$  in Eq. (18) are taken to be 1.

### C. MaxEkLin2/Max-k-XOR

The MaxEkLin2 problem is a further generalization of MaxE3Lin2 for arbitrary fixed  $k$ , where we replace the  $a, b, c$  with  $a_1, \dots, a_k$  in the cost function and the sum is taken over hyperedges of size  $k$ . This class of problems was previously studied for QAOA in [42,43]. For each  $k$ ,  $P(c')$  is the same as for MaxE3Lin2 above. However,  $P_{\text{same}}$  is given by

$$P_{\text{same}} = \frac{\sum_{h=0}^{\lfloor k/2 \rfloor} \binom{d}{2h} \binom{n-d}{k-2h}}{\binom{n}{k}}, \quad (21)$$

where the terms in the sum represent all possible ways to choose an even number of bits to flip from the  $k$  bits in the clause out of  $d$  total bits. Then, again, we have  $P_{\text{both}} = P_{\text{neither}} = P_{\text{same}}/2$  and  $P_{\text{one}} = (1 - P_{\text{both}} - P_{\text{neither}})/2$ . A similar calculation for the Max-k-XOR problem again yields identical probability distributions.

### D. Rand-k-SAT

The case of random k-SAT is analyzed by Hogg in [3]. This cost function is defined as the sum over  $m$  clauses of a logical OR of  $k$  variables randomly drawn from a set of  $n$  variables, each of which may be negated. In the notation used in this paper, the distributions of interest are

$$P(c') = 2^{-km} (2^k - 1)^{m-c'} \binom{m}{c'}, \quad P_{\text{both}} = \frac{2^{-k} \binom{n-d}{k}}{\binom{n}{k}},$$

$$P_{\text{one}} = 2^{-k} - P_{\text{both}}, \quad P_{\text{neither}} = 1 - 2P_{\text{one}} - P_{\text{both}}. \quad (22)$$

## IV. HOMOGENEOUS HEURISTIC FOR PARAMETER SETTING

Leveraging the classical homogeneous proxy for QAOA, here we propose a strategy for finding good algorithm parameters, which we call the homogeneous heuristic for parameter setting, as pictured in Fig. 1 and formalized in Algorithm 3. The main idea is that instead of using tight estimates of the expected cost values, which can be expensive to obtain, one may instead search parameter space using the approximate expectation values obtained from the classical proxy. Only after this classical parameter setting stage would the quantum computer be used to sample from the true quantum state and obtain candidate problem solutions.

**Algorithm 3.** Homogeneous Heuristic for Parameter Setting.

---

**Input:**  $p, \vec{\gamma}_{\text{in}}, \vec{\beta}_{\text{in}}, n, m, C, N(c'; d, c) \forall c' \in C, P(c') \forall c' \in C,$   
constraints on  $(\vec{\gamma}, \vec{\beta})$ .

**Output:**  $\vec{\gamma}_{\text{out}}, \vec{\beta}_{\text{out}}$

$\vec{\gamma}, \vec{\beta} \leftarrow \vec{\gamma}_{\text{in}}, \vec{\beta}_{\text{in}}$

**while** Desired **do**

(1) Using any suitable parameter update scheme, perform one update of  $(\vec{\gamma}, \vec{\beta})$

(2) Evaluate the homogeneous parameter objective function ([Eq. (6)] using the classical homogeneous proxy for QAOA for all  $(\vec{\gamma}, \vec{\beta})$  required to perform next parameter update in 1)

**end while**

$\vec{\gamma}_{\text{out}}, \vec{\beta}_{\text{out}} \leftarrow \vec{\gamma}, \vec{\beta}$

---

Here, a “suitable parameter update scheme” is intended to encapsulate a wide variety of general or specific approaches proposed for this in the literature, “Desired” denotes that the while-loop can be iterated until the update scheme terminates or some desired convergence criteria is met, and “Constraints on  $(\vec{\gamma}, \vec{\beta})$ ” denotes any restrictions on the domain of values allowed for  $\vec{\gamma}, \vec{\beta}$ , including restrictions to schedules of a prescribed form such as linear ramps introduced in Sec. VB.

With the heuristic, we replace the typical cost expectation value with the homogeneous parameter objective function, where each function evaluation takes time as determined by Algorithms 1 and 2. This heuristic is purposefully defined in broad terms, in order to maintain complete freedom in the choice of parameter update schemes. Thus one can still apply a myriad of approaches explored in the parameter-setting literature, such as parameter initialization, reparametrization, and the use of different global or local optimization algorithms.

On the other hand, we emphasize that, while our approach can significantly speed up the parameter setting task, it is by no means a panacea. Indeed, in cases where the number of parameters to optimize grows with the problem size (e.g., when  $p = n$ ), this problem suffers generically from the curse of dimensionality, as well as other potential difficulties such as barren plateaus or plentiful local optima. Hence the incorporation of a variety of parameter setting strategies or approximations that seek to ameliorate these difficulties within our approach is well motivated.

## V. NUMERICALLY INVESTIGATING THE CLASSICAL HOMOGENEOUS PROXY FOR QAOA FOR MAXCUT

In this section, we explore the application of the classical homogeneous proxy for QAOA to MaxCut on Erdős-Rényi  $\mathcal{G}(n, p_e)$  model graphs as considered in Sec. III A. We present numerical evidence justifying the methods presented in Secs. II, III, and IV.

### A. Viability of replacement distance and cost distributions

Here we numerically study the accuracy of replacing  $n(x; d, c)$  distributions with  $N(c'; d, c)$  distributions as calculated via the methods presented in Sec. III. For these experiments, we first choose 10  $\mathcal{G}(10, 1/3)$  Erdős-Rényi model graphs and calculate the  $n(x; d, c)$  for all bitstrings

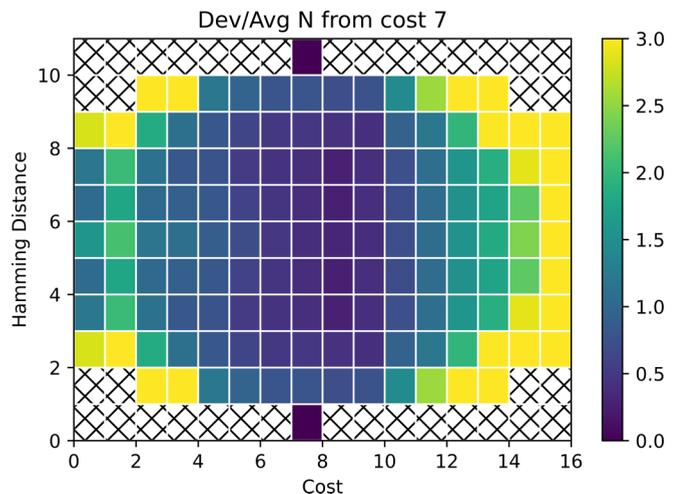


FIG. 2. Heat map of the standard deviation/mean of  $N(c'; d, c)$  distributions for 10 instances of  $\mathcal{G}(10, 1/3)$  graphs, with  $c'$  fixed at 7. Hatched out squares correspond to those elements of the distribution for which  $N(c'; d, c)$  is always 0, meaning that no bitstring at distance  $d$  from a bitstring with cost 7 has cost  $c' = 7$  was chosen because, for  $\mathcal{G}(10, 1/3)$  graphs,  $P(c')$  is peaked near 7, so this represents a typical instance of an  $N(c'; d, c)$  distribution.

$x \in \{0, 1\}^n$ , all  $d \in [0, n]$ , and all  $c \in [0, m]$ , with  $m$  being an upper bound on the maximum cost [here we use  $p_e \binom{n}{2}$ , as described in Sec. III A; in this case  $m = 15$ ]. For each cost  $c'$ , we consider  $n(x; d, c)$  for all states  $x$  with cost  $c'$  across all 10 graphs. In order to evaluate the viability of replacing  $n(x; d, c)$  with  $N(c'; d, c)$ , we present the following intuition: the better that  $N(c'; d, c)$  estimates the average of  $n(x; d, c)$  over all  $x$  with  $c(x) = c'$ , and the less  $n(x; d, c)$  deviates over  $x$  with  $c(x) = c'$ , the better  $N(c'; d, c)$  should estimate  $n(x; d, c)$  for all  $x$  with  $c(x) = c'$ . We thus aim to numerically demonstrate the extent to which both the analytically derived  $N(c'; d, c)$  estimate the average  $n(x; d, c)$  and to which  $n(x; d, c)$  deviates from its average. We first demonstrate the latter. To do this, we take the elementwise averages of these distributions. This average is one way of computing the distributions  $N(c'; d, c)$ , as described in Sec. II A. We also take the elementwise standard deviations of these distributions. In Fig. 2 we display the elementwise ratio of standard deviation to mean of these distributions for  $c' = 7$ , chosen because  $P(c')$  is maximal near 7, such that this term has large weight in Eq. (5).

From the figure, we see that for costs near  $m/2$  (7.5) and distances near  $n/2$  (5), there is minimal deviation in the  $N(c'; d, c)$  distributions among multiple instances of Erdős-Rényi model graphs and multiple bitstrings  $x$  with cost  $c'$ . We note that the relative deviation is highest at the edges of the plot, where  $d \rightarrow 0, d \rightarrow n, c \rightarrow c_{\text{min}}$ , and  $c \rightarrow c_{\text{max}}$ . We note, however, that, at these points, the expected value of  $n(x; d, c)$  is smaller, such that the contribution of these deviations to the sum in Eq. (4) is less than those with distance and cost near the center of the distribution. As an example, there are  $\binom{10}{5} = 252$  bitstrings at distance 5 from a given bitstring  $x$ , as opposed to  $\binom{10}{1} = 10$  bitstrings at distance 1. A similar argument can be made using the values of  $P(c')$  derived in Sec. III. This numerical evidence thus suggests that replacing  $n(x; d, c)$  with  $N(c'; d, c)$  determined via an averaging over all  $x$  with cost  $c'$

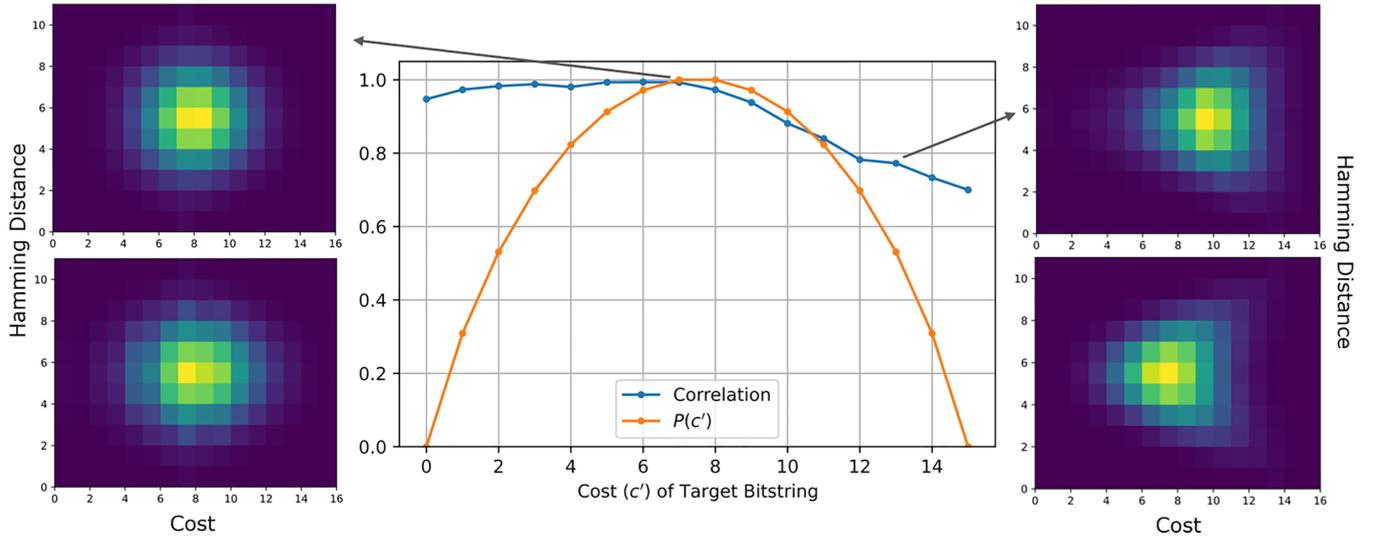


FIG. 3. (Middle) Pearson correlation coefficients between  $N(c'; d, c)$ , calculated through the averaging over 10  $\mathcal{G}(10, 1/3)$  graph instances and through the analytical method presented in Sec. III.  $P(c')$ , the probability of a random bitstring having cost  $c'$ , is also displayed to elucidate dominant terms in the sum of Eq. (5). Insets display the distribution for fixed  $c'$  for (left)  $c' = 7$  using (top) the analytical approach and (bottom) the averaging approach. (Right) Inset displays the same for  $c' = 13$ .

over multiple instances may introduce deviations from Eq. (5) precisely in cases that contribute to less Eq. (2), allowing for near-homogeneous evolution.

This result provides evidence that the deviation in  $n(x; d, c)$  is small for the “bulk” of contributions to the sum in Eq. (2), such that we can replace  $n(x; d, c)$  with  $N(c'; d, c)$ , the average over  $x$  with  $c(x) = c'$  for the *entire class* of problems. We then would like to understand how the analytically derived expressions for  $N(c'; d, c)$  in Sec. III estimate these averaged distributions. We perform the comparison of  $N(c'; d, c)$  computed by averaging and the methods in Sec. III for MaxCut on Erdős-Rényi model graphs in Fig. 3, showing the Pearson correlation coefficient between the two distributions for each  $c'$  in  $[0, m]$ . We likewise display  $P(c')$ , in order to elucidate the dominant distributions in the sum of Eq. (5).

From the figure we see that, for dominant terms [with high  $P(c')$ ], the two distributions align well visually, corresponding to a high correlation coefficient. For less important terms, the analytical distributions do not match the average over many instances, but the effect of this mismatch may be reduced due to the lesser weight on these terms as determined by  $P(c')$  in the sums of Eq. (5).

Combined, Figs. 2 and 3 show that, for dominant terms, there is little deviation in  $n(x; d, c)$  distributions from their average  $N(c'; d, c)$  and the analytically computed distributions match these average distributions well. Thus they together indicate that the analytical methods for calculating  $N(c'; d, c)$  should approximate  $n(x; d, c)$  with  $c(x) = c'$  for terms that dominate the sum in Eq. (5).

## B. Numerical overlaps

In this section we numerically show that the proxy maintains large overlaps with the true QAOA state for certain parameter schedules of interest. To do this, we perform numerical simulations of the proxy on MaxCut on Erdős-Rényi graphs  $\mathcal{G}(10, 1/2)$  and display the squared overlap between

classical full state vector simulation and the proxy as a function of  $p$  for various parameter schedules motivated by QAOA literature. For this analysis, we choose linear ramp parameter schedules, inspired by quantum annealing. In particular, we fix a starting and ending point for  $\bar{\gamma}$  and  $\bar{\beta}$ , which is kept constant regardless of  $p$ , and the schedule is defined as

$$\gamma_\ell = \gamma_1 + (\gamma_f - \gamma_1) \frac{\ell}{p}, \quad \beta_\ell = \beta_1 + (\beta_f - \beta_1) \frac{\ell}{p} \quad (23)$$

for each layer  $\ell$ . Given these linear ramp schedules, the squared overlaps between the replaced and original quantities are calculated as follows. First, state vector simulation was performed using HYBRIDQ, an open-source package for large-scale quantum circuit simulation [44]. Next, the  $N(c'; d, c)$  distributions are computed according to Sec. III. Then, for the purposes of comparison, we compute the proxy slightly differently from above, by starting with the uniform superposition over all  $2^n$  bitstrings and simply plugging in  $N(c(x); d, c)$  for all  $n(x; d, c)$  in Eq. (4), keeping all  $2^n$  amplitudes  $q^\ell(x)$  at each step  $\ell$ . This allows us to compute the overlap between the proxy and true state using standard quantum state overlap, although we lose the gain in complexity from performing the proxy on only the set of unique costs, as we are working with  $2^n$  bitstrings rather than  $|C|$  costs. Then, using this method, we plot the squared overlaps between the replace and original quantities as a function of  $p$  with varying values of  $\gamma_1$  and  $\gamma_f$  in Fig. 4.

From the figure, we can see that the overlap gradually decreases as the number of QAOA layers increases. However, the decline is less dramatic when  $\gamma_1$  and  $\gamma_f$  are lower in magnitude. Thus we see that as we move towards the  $\gamma \ll 1$  regime for these problems (or, more precisely  $\gamma \ll \|C\|$  [39]), the true QAOA state remains closer to the proxy even as the algorithm progresses, as remarked in Sec. II. We stress that this behavior is empirical and the numerics are limited to the MaxCut examples analyzed presently. This behavior does however align with the analytical fact mentioned in Sec. II

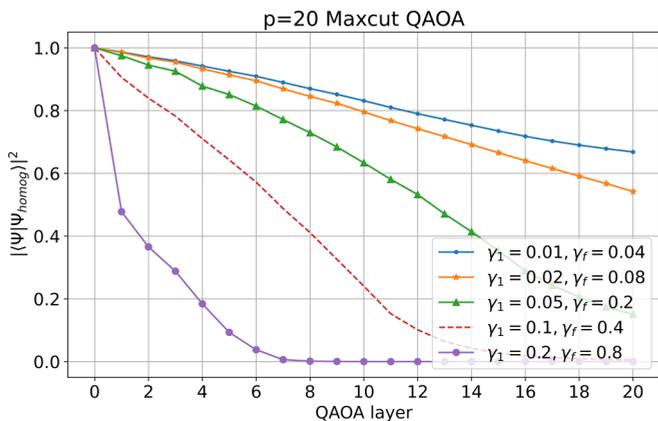


FIG. 4. Squared overlap between  $p = 20$  MaxCut QAOA runs on a  $\mathcal{G}(8, 1/2)$  Erdős-Rényi model graph instance using full state simulation vs Eq. (5), with distributions calculated as in Sec. III as a function of the current QAOA layer. A fixed linear ramp parameter schedule is chosen, with  $\gamma$  increasing and  $\beta$  decreasing at each step. Each curve corresponds to differing values of  $\gamma_1$  and  $\gamma_f$ .

that, to first order in  $\gamma$ , QAOA states are homogeneous for strictly  $k$ -local Hamiltonians.

### C. Parameter objective function landscapes at low depth

In this section we provide a toy example for a small graph at  $p = 3$ , empirically showing that the typical parameter objective function correlates significantly with the corresponding approximations obtained from the homogeneous proxy for important parameter regimes. In order to provide an explicit illustrative example for the efficacy of the homogeneous heuristic for parameter setting in Algorithm 3, we depict both the typical and homogeneous parameter objective function as a function of  $\gamma_3$  and  $\beta_3$  for a randomly drawn  $\mathcal{G}(8, 1/2)$  graph and QAOA with  $p = 3$ . In this example, our aim is to visualize similarities between the two parameter

objective functions rather than to exhaustively find optimal parameters for the graph. As such, we borrow  $\gamma_1, \gamma_2, \beta_1,$  and  $\beta_2$ , optimized from a 20 node instance in Sec. VIA. These landscapes are shown in Fig. 5.

It is visually clear that the two landscapes have significant differences. For the typical parameter objective function, there exists a clearly defined, Gaussian-like peak (yellow) and valley (blue). For the homogeneous parameter objective function, there exists a similarly located peak, albeit vertically compressed, and the corresponding valley comprises almost the entire rest of the landscape. However, we can see that, in the small  $\gamma$  regime in particular, the landscapes qualitatively look very similar. This behavior is suggested by Fig. 4, where we see quantitatively that the extent to which the classical homogeneous proxy for QAOA overlaps with the true QAOA evolution grows as  $\gamma$  decreases. Additionally, as seen in previous studies of QAOA parameters [10,11,13,14], optimal values of  $\gamma$  tend to remain relatively small (exact values are not described as they depend on  $n, p$ , and the cost function being used), especially at the beginning of the algorithm. This suggests that, while the homogeneous and typical parameter objective functions may deviate significantly in general, they maintain significant correlations in *relevant* parameter regimes, specifically those which are near the maximum in the landscape. Indeed, for the task of parameter setting, we expect qualitative feature mapping of the landscape to be much more important than a precise matching of objective function values.

It is also worthwhile to consider the difference in computational resources needed to produce the two plots. For the typical parameter objective function, in order to classically evaluate the evolution of the algorithm in each cell of the presented  $30 \times 30$  landscapes, we perform full state vector simulation. Farhi *et al.* show in [2] that, in order to compute Pauli observable expectation values for the typical parameter objective function, one only needs to include qubits that are within the reverse causal cone generated by the qubits

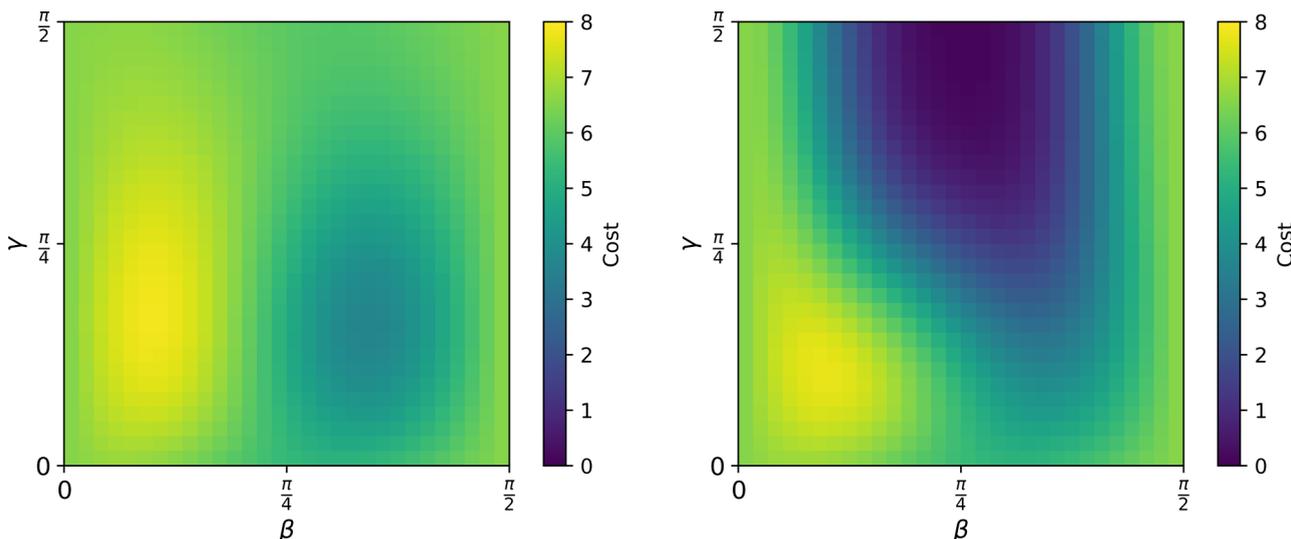


FIG. 5. Parameter objective function landscapes displayed as heat maps as a function of  $\gamma_3$  and  $\beta_3$  for  $p = 3$  MaxCut QAOA on a  $\mathcal{G}(8, 1/2)$  Erdős-Rényi model graph with  $\gamma_1, \gamma_2, \beta_1,$  and  $\beta_2$  fixed. (Left) The typical parameter objective function, computed via classical full state vector simulation. (Right) The homogeneous parameter objective function computed via the classical homogeneous proxy for QAOA.

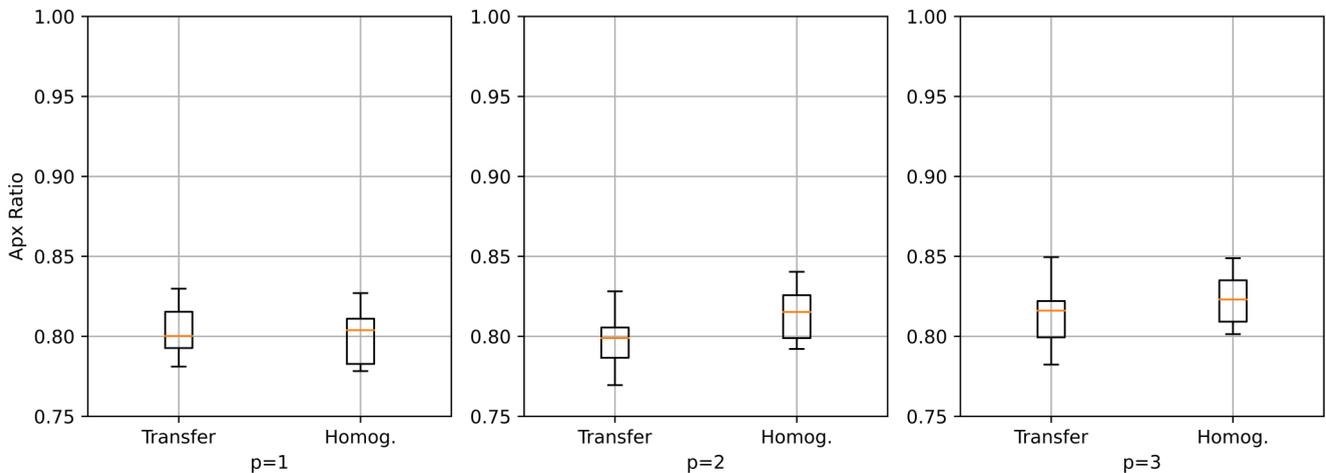


FIG. 6. Box plots of approximation ratios obtained by the transfer of the median of optimal parameters from 10  $\mathcal{G}(9, 1/2)$  graphs to 10  $\mathcal{G}(20, 1/2)$  graphs vs parameters found via a homogeneous parameter setting for the same 10  $\mathcal{G}(20, 1/2)$  graphs, for  $p = 1, 2$ , and 3.

involved in the observable and the quantum circuit implementing QAOA. However, for the example analyzed here, at  $p = 3$ , this reverse causal cone includes all 8 qubits for each observable, so in order to classically compute the evolution, we perform a full-state simulation. Thus deriving evolution of the proxy took roughly one-fiftieth of the time required for simulating full QAOA. We note that it is possible to efficiently evaluate each cell on an actual quantum computer and that, if one only wants expectation values given parameters rather than the full state evolution, there are more efficient classical methods (e.g., Refs. [2,31]). Current difficulties for this approach, however, include noise resulting both from finite sampling error as well as the effects of imperfect quantum hardware.

## VI. RESULTS

In this section we present numerical evidence supporting the homogeneous heuristic for parameter setting, again using MaxCut on Erdős-Rényi model graphs as a target application. Due to the array of possible techniques implementing the parameter-update scheme as mentioned in Sec. IV, we do not wish to provide an exhaustive comparison of the heuristic to previous literature, but rather demonstrate regimes where the heuristic provides parameters that are either comparable with previous results or that yield increasing performance out to larger values of  $p$ , where we are not aware of prior methods in the literature successfully returning well-performing parameter schedules.

### A. Global optimization at low depth

Here we present results of the homogeneous heuristic for parameter setting, as well as comparisons to the transfer of parameters method outlined in Lotshaw *et al.* [10], implemented using the QAOAKIT software package [11]. Lotshaw *et al.* show that using one set of median (over the entire data set at a given  $n$  and  $p$ ) parameters performs similarly to optimized parameters for each instance. Thus we directly pull the obtained parameters from QAOAKIT, which first calculates optimal parameters for all connected nonisomorphic graphs

up to size 9 at  $p = 1, 2$ , and 3. For each  $p$ , the median over all parameters is calculated and these median parameters are directly applied to 10 Erdős-Rényi graphs from  $\mathcal{G}(20, 1/2)$ , yielding average and standard deviation of expectation values for these median parameters over the 10 graphs. To compare with these transferred parameters, we display the approximation ratio achieved by parameters that are optimized with the heuristic, as described in Sec. IV, over the same 10 graphs. Here the approximation ratio is defined as follows:

$$\text{Apx Ratio} = \langle C \rangle / c_{\text{opt}}, \quad (24)$$

i.e., the expected cost value returned by true QAOA, divided by the true optimal value  $c_{\text{opt}}$ , as determined via brute-force search over all  $2^n$  bitstrings. For this experiment, as well as all experiments below, the state throughout the algorithm and  $\langle C \rangle$  are computed via full state vector simulation, even for parameters returned via the heuristic. The comparison between the heuristic and parameter transfer is shown in Fig. 6.

As we can see in Fig. 6, for low depth, the heuristic performs comparably well to parameter transfer. On an instance-by-instance basis, the approximation ratio achieved by homogeneous parameter setting minus that achieved by parameter transfer was  $-0.0037 \pm 0.0062$ ,  $0.0164 \pm 0.0148$ , and  $0.0097 \pm 0.0183$  for  $p = 1, 2$ , and 3, respectively. We do not see statistically significant differences between the two methods for any three of the depths analyzed, although the average performance of the heuristic is slightly higher in the latter two cases. This numerical evidence indicates that the method is competitive. Furthermore, the optimal parameters in the transfer case require the optimization of smaller QAOA instances, which clearly may incur some tradeoff between the size of the problem one wishes to train parameters on versus the accuracy of the parameter transfer onto larger and larger instances. The parameters for comparison were pulled directly from QAOAKIT data tables, so our purpose here is not to provide a full timing comparison between the two methods. However, this demonstrates that our polynomially scaling heuristic performs comparably with other techniques used in literature.

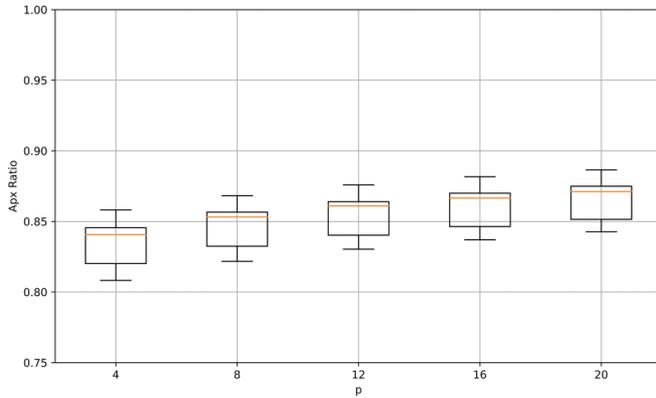


FIG. 7. Box plots of approximation ratios for parameters found via a homogeneous parameter setting for the same 10  $\mathcal{G}(20, 1/2)$  graphs, for  $p = 4, 8, 12, 16,$  and  $20$ , restricted to linear ramp schedules as described in Sec. VC.

### B. Parameter optimization at higher depth

To elucidate how the homogeneous heuristic for parameter setting scales with QAOA depth  $p$ , we further depict box plots of the approximation ratio for a new set of 10 Erdős-Rényi graphs  $\mathcal{G}(20, 1/2)$  at  $p$  up to 20. For these experiments, we further restrict to linear-ramp parameter schedules as described in Eq. (23) to reduce the number of parameters from  $2p$  to 4. We introduce this reparametrization because having  $2p$  free parameters, even for this relatively moderately sized  $p$ , results in optimization routines that do not converge in a reasonable time on the device as specified below. The results for these runs are shown in Fig. 7.

From this figure, we see that the heuristic, when implemented with linear ramp schedules, results in monotonic improvement of approximation ratios as  $p$  increases. Notably, for this regime of  $N = 20$ ,  $p = 20$ , we were not able to find previous works that efficiently returned optimized parameter schedules, even when restricted to linear ramps. Thus these results demonstrate a regime in which the heuristic is able to return parameters that appear intractable for current devices and strategies, whether quantum, classical, or hybrid.

*Numerical details.* For our simulations in this section, all calculations (excluding those pulled from the QAOAKIT database) were performed using a laptop with Intel i7-10510U CPUs @ 1.80 GHz and 16 GB of RAM, with no parallelization utilized. For the 20 node graphs, all experiments clocked in under 6 h, where the longest times were for fully parametrized  $p = 3$  circuits (six parameters). Parameters were seeded using linear ramp schedules from [13] and parameter optimization was performed using the standard Broyden-Fletcher-Goldfarb-Shanno algorithm [45] from the SCIPY package [46].

## VII. DISCUSSION

In this work we formalized the concepts of homogeneous states and the classical homogeneous proxy for QAOA. We demonstrated how to derive the necessary quantities and efficiently evaluate the proxy for combinatorial satisfaction problems with a fixed, polynomial number of randomly chosen clauses. We then provided numerical evidence to support

the use of the proxy for estimating the evolution and cost expectation value of QAOA. Finally, we applied these results to construct the homogeneous heuristic for QAOA and implemented this strategy for a class of MaxCut instances on graphs up to  $n = 20$  and  $p = 20$ . Our results show that the heuristic on this class easily yields parameters at  $p = 1, 2,$  and  $3$  that are comparable to those returned by parameter transfer. We further demonstrated that we are able to optimize parameters out to  $p = 20$  by restricting to a linear ramp schedule, obtaining the desirable property of monotonically increasing approximation ratios as the number of QAOA layers is increased. Notably, we found that the proxy seems to well estimate both the state and cost expectation of QAOA in the particular cases when  $\gamma$  remains relatively small throughout the algorithm, as well as for quantum annealing-inspired linear ramp schedules. These ramp schedules have been frequently proposed as empirically well-performing schedules [10,13,26], which supports that the proxy may more accurately estimate QAOA expectation values for important parameter regimes and schedules of interest, even if these estimates may diverge somewhat in the case of arbitrarily chosen parameters.

Several interesting research questions and future directions directly follow from our results. An immediate question is to better understand the relationship between the problem class specified, the resulting distributions  $N(c'; d, c)$  and  $P(c')$  used for the proxy, and the effect on the parameters returned by the homogeneous heuristic for QAOA, especially with respect to a given problem instance to be solved. For example, a fixed instance can be drawn from a number of different possible classes, so changing the class considered can have a significant effect on the parameters returned and resulting performance. One approach to address this issue would be to extend the derivations of  $N(c'; d, c)$  and  $P(c')$  to incorporate instance-specific information beyond just the problem class. A naive example in this vein would be to estimate the distributions via Monte Carlo sampling of bitstrings and their costs for the given instance. Furthermore, including instance-specific information appears a promising route to explicitly extending the heuristic beyond random problem classes, which can be used to study parameter schedules and performance in the worst-case setting. It may also be worthwhile to explore adaptations of our approach to cases where the number of unique possible costs may become large. In this case, one could imagine binning together costs close in value such that an effective cost function with much fewer possible costs is produced and to which the proxy may be applied. For applying our methods more generally, an interesting direction is to understand whether problem class characteristics such as the overlap gap property (OGP) [47–49] can be used to derive better performing approximate  $n(x; d, c)$  or  $N(c'; d, c)$  distributions and whether or not techniques for computing these distributions can lead to progress on determining the existence of OGPs.

In terms of generalizing both the methods and scope of our approach, we first reemphasize that parameter optimization for parametrized quantum circuits consists of two primary components: a parameter update scheme outer loop and a parameter objective function evaluation subroutine. The inner subroutine is typically evaluated using the quantum computer. The key idea of our approach is to replace the inner subroutine

with an efficiently computable classical strategy based on the assumption of homogeneous states. Hence a natural extension is to consider other efficiently computable proxies for the inner loop. For example, in cases where the problem instance comes with a high degree of classical symmetries, the dimension of the effective Hilbert space can be drastically reduced, and so the evaluation and optimization of the typical parameter objective can be sped up significantly [23]. Similarly, different proxies may follow from related ideas and results in the literature such as the small-parameter analysis of [39], the pseudo-Boltzmann approximation of [38], and classical or quantum surrogate models [35,36]. We remark that a promising direction that appears relatively straightforward in light of our results is to extend the analysis of [38] to QAOA levels beyond  $p = 1$ . Finally, an important direction is to explicitly generalize our approach to algorithms beyond QAOA and, more generally, problems beyond combinatorial optimization, such as the parameter setting problem for variational quantum eigensolvers. Generally, it is important to better understand and characterize regimes where such classical proxies are most advantageous, such as when the noisy computation and measurements of real-world quantum devices is taken into

account, as well as to what degree undesirable effects such as barren plateaus may apply when such proxies are utilized for parameter setting. Lastly, a future extension of our work could be to derive rigorous approximation bounds for the classical homogeneous proxy on various problem classes or instances. This work could determine, among other behavior, how the accuracy of the proxy scales with increasing  $\gamma$  and number of layers, as explored in Fig. 4.

## ACKNOWLEDGMENTS

We are grateful for support from the NASA Ames Research Center. We acknowledge funding from the NASA ARMD Transformational Tools and Technology (TTT) Project. Part of the project was funded by the Defense Advanced Research Projects Agency (DARPA) via IAA8839 annex 114. J.S., S.H., and T.H. are thankful for support from NASA Academic Mission Services, Contract No. NNA16BD14C. We also acknowledge XSEDE computational Project No. TG-MCA93S030 on Bridges-2 at the Pittsburgh supercomputer center. The majority of the work of J.S. was completed at USRA.

- 
- [1] S. Hadfield, Z. Wang, B. O’Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, *Algorithms* **12**, 34 (2019).
- [2] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [3] T. Hogg, Quantum search heuristics, *Phys. Rev. A* **61**, 052311 (2000).
- [4] T. Hogg and D. Portnov, Quantum optimization, *Inf. Sci.* **128**, 181 (2000).
- [5] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021).
- [6] D. Wierichs, C. Gogolin, and M. Kastoryano, Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer, *Phys. Rev. Res.* **2**, 043246 (2020).
- [7] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nat. Commun.* **9**, 4812 (2018).
- [8] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan, P. J. Coles, and M. Cerezo, Diagnosing barren plateaus with tools from quantum optimal control, *Quantum* **6**, 824 (2022).
- [9] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, Noise-induced barren plateaus in variational quantum algorithms, *Nat. Commun.* **12**, 6961 (2021).
- [10] P. C. Lotshaw, T. S. Humble, R. Herrman, J. Ostrowski, and G. Siopsis, Empirical performance bounds for quantum approximate optimization, *Quantum Inf. Process* **20**, 403 (2021).
- [11] R. Shaydulin, K. Marwaha, J. Wurtz, and P. C. Lotshaw, QAOAKit: A toolkit for reproducible study, application, and verification of the QAOA, *2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)* (IEEE, St. Louis, MO, USA, 2021), pp. 64–71.
- [12] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash, Learning to optimize variational quantum circuits to solve combinatorial problems, *Proc. AAAI Conf. Artif. Intell.* **34**, 2367 (2020).
- [13] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices, *Phys. Rev. X* **10**, 021067 (2020).
- [14] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem, [arXiv:1811.08419](https://arxiv.org/abs/1811.08419) [quant-ph].
- [15] R. Shaydulin, I. Safro, and J. Larson, Multistart methods for quantum approximate optimization, in *2019 IEEE High Performance Extreme Computing Conference (HPEC)* (IEEE, New York, 2019), pp. 1–8.
- [16] M. Alam, A. Ash-Saki, and S. Ghosh, Accelerating quantum approximate optimization algorithm using machine learning, in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (IEEE, New York, 2020), pp. 686–689.
- [17] 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](https://arxiv.org/abs/1907.05415) [quant-ph].
- [18] M. Wilson, R. Stromswold, F. Wudarski, S. Hadfield, N. M. Tubman, and E. G. Rieffel, Optimizing quantum heuristics with meta-learning, *Quantum Mach. Intell.* **3**, 13 (2021).
- [19] A. Skolik, J. R. McClean, M. Mohseni, P. van der Smagt, and M. Leib, Layerwise learning for quantum neural networks, *Quantum Mach. Intell.* **3**, 5 (2021).
- [20] J. Rivera-Dean, P. Huembeli, A. Acín, and J. Bowles, Avoiding local minima in variational quantum algorithms with neural networks, [arXiv:2104.02955](https://arxiv.org/abs/2104.02955).
- [21] L. Bittel and M. Kliesch, Training variational quantum algorithms is NP-hard, *Phys. Rev. Lett.* **127**, 120502 (2021).

- [22] M. Streif and M. Leib, Training the quantum approximate optimization algorithm without access to a quantum processing unit, *Quantum Sci. Technol.* **5**, 034008 (2020).
- [23] R. Shaydulin, S. Hadfield, T. Hogg, and I. Safro, Classical symmetries and the quantum approximate optimization algorithm, *Quantum Inf. Process.* **20**, 359 (2021).
- [24] L. T. Brady, C. L. Baldwin, A. Bapat, Y. Kharkov, and A. V. Gorshkov, Optimal protocols in quantum annealing and quantum approximate optimization algorithm problems, *Phys. Rev. Lett.* **126**, 070505 (2021).
- [25] S. H. Sack and M. Serbyn, Quantum annealing initialization of the quantum approximate optimization algorithm, *Quantum* **5**, 491 (2021).
- [26] J. Wurtz and P. J. Love, Counterdiabaticity and the quantum approximate optimization algorithm, *Quantum* **6**, 635 (2022).
- [27] F. G. S. L. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances, [arXiv:1812.04170](https://arxiv.org/abs/1812.04170) [quant-ph].
- [28] X. Lee, Y. Saito, D. Cai, and N. Asai, Parameters fixing strategy for quantum approximate optimization algorithm, in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Broomfield, CO, USA, 2021), pp. 10–16.
- [29] A. Galda, X. Liu, D. Lykov, Y. Alexeev, and I. Safro, Transferability of optimal QAOA parameters between random graphs, in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Broomfield, CO, USA, 2021), pp. 171–180.
- [30] L. T. Brady, L. Kocia, P. Bienias, A. Bapat, Y. Kharkov, and A. V. Gorshkov, Behavior of analog quantum algorithms, [arXiv:2107.01218](https://arxiv.org/abs/2107.01218).
- [31] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, Quantum approximate optimization algorithm for MaxCut: A fermionic view, *Phys. Rev. A* **97**, 022304 (2018).
- [32] K. Marwaha, Local classical MAX-CUT algorithm outperforms  $p=2$  QAOA on high-girth regular graphs, *Quantum* **5**, 437 (2021).
- [33] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, Improving variational quantum optimization using CVaR, *Quantum* **4**, 256 (2020).
- [34] L. Li, M. Fan, M. Coram, P. Riley, and S. Leichenauer, Quantum optimization with a novel Gibbs objective function and ansatz architecture search, *Phys. Rev. Res.* **2**, 023074 (2020).
- [35] K. J. Sung, J. Yao, M. P. Harrigan, N. C. Rubin, Z. Jiang, L. Lin, R. Babbush, and J. R. McClean, Using models to improve optimizers for variational quantum algorithms, *Quantum Sci. Technol.* **5**, 044008 (2020).
- [36] R. Shaffer, L. Kocia, and M. Sarovar, Surrogate-based optimization for variational quantum algorithms, *Phys. Rev. A* **107**, 032415 (2023).
- [37] J. Mueller, W. Lavrijsen, C. Iancu, and W. de Jong, Accelerating noisy VQE optimization with Gaussian processes, in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Broomfield, CO, USA, 2022), pp. 215–225.
- [38] P. Díez-Valle, D. Porras, and J. J. García-Ripoll, Quantum approximate optimization algorithm pseudo-Boltzmann states, *Phys. Rev. Lett.* **130**, 050601 (2023).
- [39] S. Hadfield, T. Hogg, and E. G. Rieffel, Analytical framework for quantum alternating operator ansätze, *Quantum Sci. Technol.* **8**, 015017 (2023).
- [40] B. Barak, A. Moitra, R. O'Donnell, P. Raghavendra, O. Regev, D. Steurer, L. Trevisan, A. Vijayaraghavan, D. Witmer, and J. Wright, Beating the random assignment on constraint satisfaction problems of bounded degree, [arXiv:1505.03424](https://arxiv.org/abs/1505.03424).
- [41] M. Hastings, Classical and quantum bounded depth approximation algorithms, *Quantum Inf. Comput.* **19**, 1116 (2019).
- [42] K. Marwaha and S. Hadfield, Bounds on approximating Max-kXOR with quantum and classical local algorithms, *Quantum* **6**, 757 (2022).
- [43] C.-N. Chou, P. J. Love, J. S. Sandhu, and J. Shi, Limitations of local quantum algorithms on random Max-k-XOR and beyond, in 49th International Colloquium on Automata, Languages, and Programming (ICALP 2022).
- [44] S. Mandrà, J. Marshall, E. G. Rieffel, and R. Biswas, HybridQ: A hybrid simulator for quantum circuits, [arXiv:2111.06868](https://arxiv.org/abs/2111.06868) [quant-ph].
- [45] R. (Roger) Fletcher, *Practical Methods of Optimization* (Wiley, Chichester, New York, 1987).
- [46] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat *et al.*, SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nat. Methods* **17**, 261 (2020).
- [47] D. Gamarnik and Q. Li, Finding a large submatrix of a Gaussian random matrix, *Ann. Stat.* **46**, 2511 (2018).
- [48] M. Mézard, T. Mora, and R. Zecchina, Clustering of solutions in the random satisfiability problem, *Phys. Rev. Lett.* **94**, 197205 (2005).
- [49] D. Achlioptas and F. Ricci-Tersenghi, On the solution-space geometry of random constraint satisfaction problems, in *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC'06 (ACM, New York, 2006), pp. 130–139.