

Received May 6, 2021; revised October 18, 2021; accepted October 19, 2021; date of publication October 26, 2021; date of current version December 7, 2021.

Digital Object Identifier 10.1109/TQE.2021.3122568

Classically Optimal Variational Quantum Algorithms

JONATHAN WURTZ^{ID} AND PETER J. LOVE

Department of Physics and Astronomy, Tufts University, Medford, MA 02155 USA

Corresponding author: Jonathan Wurtz (jonathan.wurtz@tufts.edu)

This work was supported by the Defense Advanced Research Projects Agency under Contract HR001120C0068.

ABSTRACT Hybrid quantum-classical algorithms, such as variational quantum algorithms (VQAs), are suitable for implementation on noisy intermediate-scale quantum computers. In this article, we expand an implicit step of VQAs: the classical precomputation subroutine, which can nontrivially use classical algorithms to simplify, transform, or specify problem instance-specific variational quantum circuits. In VQA, there is a tradeoff between quality of solution and difficulty of circuit construction and optimization. In one extreme, we find VQA for MAXCUT, which are exact, but circuit design or variational optimization is NP-hard. At the other extreme are low-depth VQA, such as the Quantum Approximate Optimization Algorithm (QAOA), with tractable circuit construction and optimization but poor approximation ratios. Combining these two, we define the Spanning Tree QAOA to solve MAXCUT, which uses an ansatz whose structure is derived from an approximate classical solution and achieves the same performance guarantee as the classical algorithm and, hence, can outperform QAOA at low depth. In general, we propose integrating these classical precomputation subroutines into VQA to improve heuristic or guaranteed performance.

INDEX TERMS Hybrid algorithm, noisy intermediate-scale quantum (NISQ) algorithm, quantum computing, variational quantum algorithm (VQA).

I. INTRODUCTION

Today's noisy intermediate-scale quantum (NISQ) computers are bounded in power by size, noise, and decoherence [32]. Do there exist implementable hybrid quantum-classical algorithms that outperform the best classical algorithms? Such an algorithm would exhibit a quantum advantage, perhaps the most ambitious goal of the NISQ era. One class of algorithms which shows promise are variational quantum algorithms (VQAs) [7], [11], [26], [28], [31], which variationally optimize ansatz wavefunctions to extremize expectation values of objective functions. VQAs construct a parameterized quantum circuit $U(\vec{\alpha})$ in a classical precomputation step [see Fig. 1(b)], which is implemented on an NISQ device and optimized in an outer classical loop [see Fig. 1(c) and (d)].

Classical no-free-lunch (NFL) theorems for optimization imply that algorithmic advantages rely on problem structure [43]. Quantum NFL theorems for specific cases exist [29], [36], and for VQAs suggest that the ansatz should reflect problem structure; otherwise VQAs suffer from barren plateaus [27]. How can the structure of the problem, form of classical algorithms, or approximate solutions be used in the classical subroutines of VQA?

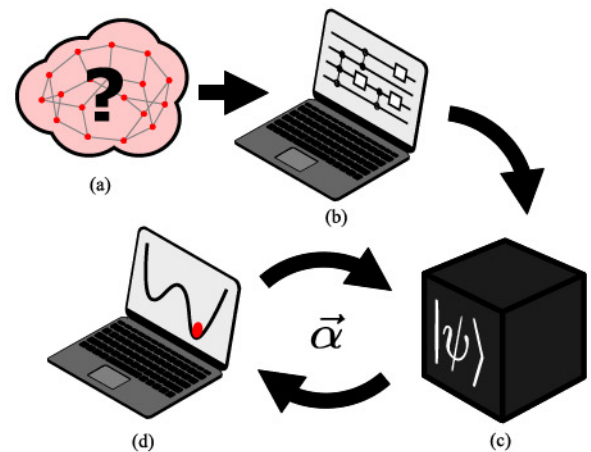


FIGURE 1. Pictorial representation of a VQA. (a) Given some problem instance, (b) a classical subroutine tailors a problem instance-specific circuit, by including problem structure, simplifying or transforming the problem, or using results and structure of classical algorithms. (c) Circuit is run by a near-term quantum machine and (d) variational parameters are optimized via repeated query of a classical optimizer.

A problem might be simplified or reduced as in [6], or by exactly solving weakly connected parts of a MAXCUT

graph as in [16]. Alternatively, a problem might be mapped or transformed to one with better heuristics [20], [30]. Classical insight may motivate the circuit structure as in the case of machine learning models [4], [37], or generate an analogous quantum version of a classical algorithm [5].

The coupled-cluster ansatz used in a quantum variational eigensolver [31], [35] and the Quantum Approximate Optimization Algorithm (QAOA) ansatz [11] reflect structure by including the terms of the objective function in the ansatz. Additionally, VQAs can use approximate classical solutions in their ansatz states using the concept of warm starts, which initialize the variational parameters with values known to mimic a good classical solution [10], [12], [17], [39]. If further variational optimization is possible, the VQA will improve upon the performance of the classical algorithm. In the worst case, improvement may not be possible due to complexity theoretic constraints [2], [21], [24], [25].

In this article, the warm start concept is generalized to construct problem instance-specific *circuits* instead of just choosing initial variational parameters that reproduce approximate solutions in a fixed ansatz circuit structure. We use the VQA precomputation step [see Fig. 1(b)] to generate problem instance-specific circuits that use problem structure from the form and solutions of classical algorithms as well as from the objective function. Similar adaptive algorithms have recently been proposed for iterative QAOA [47] and quantum chemistry [14], [40], which also attempt to better leverage classical resources.

While this precomputation step is general, we focus on solving the particular problem of MAXCUT inspired by a particular classical algorithm, with the hope of inspiring other VQA algorithms with nontrivial precomputation subroutines. We construct the spanning tree QAOA (ST-QAOA), a particular VQA to solve MAXCUT [15], [21]. This algorithm uses approximate solutions from a classical MAXCUT solver as a subroutine to construct a problem instance-specific circuit with r rounds of gates. We show that $r = 1$ is guaranteed to match the performance of the classical solver, and $r \rightarrow \infty$ approaches the exact result.

We introduce the ST-QAOA in a sequence of algorithms that illustrate the tradeoffs between quality of solution and classical computational difficulty of the precomputation step. First, we introduce the spanning tree algorithm, which can produce exact solutions at the expense that circuit generation is NP-hard. Next, we introduce the variational spanning tree (VST) algorithm, which can produce optimal answers at the expense that variational optimization is NP-hard. Finally, we introduce the ST-QAOA and present numerical evidence of its performance on random instances of 3-regular MAXCUT, demonstrating that ST-QAOA can always reproduce the performance of the classical algorithm it uses to construct its ansatz.

II. MAXCUT AND SPANNING TREES

First, let us define the SIGNED MAXCUT problem and the structure of spanning trees, which will be the algorithmic

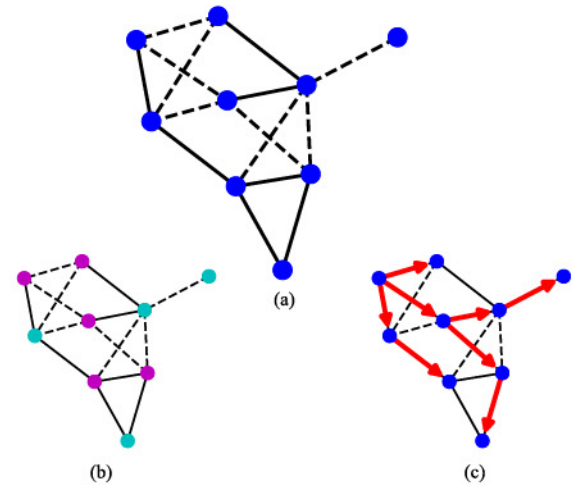


FIGURE 2. Example SIGNED MAXCUT graph instance and approximate solutions. A signed graph (a) is constructed of edges with positive (dashed) or negative (solid) signature. Dashed edges are satisfied if vertices are in the same partition, and solid edges are satisfied if vertices are in the opposite partition. (b) Example approximate bipartition z and (c) equivalent spanning tree $\mathcal{T}_z$ satisfy 9 out of 15 edges. The maximal number of cut edges is 12 out of 15 for this graph, leaving ST-QAOA room for improvement.

insight for ST-QAOA. A signed graph $\Gamma = (\mathcal{G}, \sigma)$ [45] is constructed of graph $\mathcal{G}$ and signature for each edge $\sigma = \pm 1$. The goal of SIGNED MAXCUT is to find a bipartition of vertices $\{X, Y\}$ (or binary string z labeling the bipartition) such that the maximal number of edges of Γ are satisfied (or “cut”). An edge with a negative signature is satisfied if its vertices are in opposite partitions, and unsatisfied otherwise. MAXCUT is the specific case where the signature of every edge is negative. An example signed graph is shown in Fig. 2.

SIGNED MAXCUT is closely related to balance in signed graphs [44]. A signed graph is balanced (bipartite) if there exists a bipartition of vertices such that every edge of Γ is satisfied [19]. SIGNED MAXCUT is equivalent to the maximum balanced subgraph problem [34]: given some signed graph Γ , what is the minimal number of edges e , which need be removed to make $\Gamma \setminus e$ balanced? Any solution z induces some subset of edges e , which remain unsatisfied, so that $\Gamma \setminus e$ is balanced. An optimal solution z will remove the smallest number of unsatisfied edges e .

The bipartition z of a balanced graph can be found with a directed spanning tree, as follows [22]. Given some balanced graph $\Gamma \setminus e$, construct any spanning tree $\mathcal{T}$ with a unique path between each vertex. Starting with some arbitrary origin vertex, traverse the tree to assign each vertex to a bipartition. If the signature of an edge is $(-)$, assign the next vertex in the path to the opposite partition as its parent, and the same if $(+)$. This satisfies every edge of the spanning tree and, because the graph is balanced, every edge in the reduced balanced graph $\Gamma \setminus e$. Any spanning tree over a balanced graph generates the satisfying bipartition z , as z is unique [46]. Thus, the maximal bipartition z of graph Γ is equivalent to some choice of spanning tree $\mathcal{T}_z$. In this way, the search space

of SIGNED MAXCUT solutions can be reduced to searching through the set of all possible spanning trees [33], as the optimal bipartition z is given by some particular spanning tree(s) $\mathcal{T}_z$ over the signed graph Γ .

Given some signed graph $\Gamma = (\mathcal{G}, \sigma)$, the optimal bipartition of vertices is encoded in the maximal eigenstate of the objective function

$$\hat{C} = \frac{1}{2} \sum_{(ij)} (1 + \sigma_{ij} \hat{\sigma}_z^i \hat{\sigma}_z^j) \quad (1)$$

where each clause in the sum represents an edge of the graph, with eigenvalue +1 if the edge is satisfied and eigenvalue 0 if the edge is not satisfied, and σ_{ij} the signature of the edge.

The goal of any VQA is to optimize the expectation value of the objective function with an ansatz wavefunction. We write an ansatz circuit of r rounds in the general form

$$|\psi\rangle = e^{i\hat{H}_1\alpha_1} e^{i\hat{H}_2\alpha_2} (\dots) e^{i\hat{H}_r\alpha_r} |+\rangle \quad (2)$$

where α_r are classically optimized variational parameters. Note that r may not correspond to circuit depth. The circuit is constructed from a restricted set of generators $\hat{H}_q$; for example, in QAOA, $\hat{\sigma}_z^i \hat{\sigma}_z^j$ operators acting on all edges of a graph $(ij) \in \mathcal{G}$, and $\hat{\sigma}_x^i$ operators acting on all vertices. The classical computers' challenge is specifying the circuit structure in (2) via the precomputation step [see Fig. 1(b)] and finding particular angles [see Fig. 1(d)], which maximize the objective function. Optimal bitstring solutions may be read out by observing $\langle \psi | \hat{\sigma}_z^i | \psi \rangle$ for each qubit and assigning the bitstring according to the measurement ± 1 .

We will now give exact VQA algorithms for MAXCUT. For the restricted generators of QAOA, eigenstates are cat states due to $\mathbb{Z}_2$ symmetry and the ansatz wavefunction lies in the +1 sector, so an optimal state must have the particular form of a “cat state” [12]

$$|\psi\rangle = \frac{1}{\sqrt{2}} (|z\rangle + |\bar{z}\rangle) \quad (3)$$

where z is the binary representation of the optimal MAXCUT solution, and $\bar{z} = \neg z$. The ansatz of (2) can generate such states, and so VQA can be exact [12]. Consider the sequence of unitaries between two vertices

$$|\psi\rangle = e^{i\pi\hat{\sigma}_x^1/4} e^{i\pi\hat{\sigma}_z^0\hat{\sigma}_z^1/4} |++\rangle = \frac{1+i}{2} (|01\rangle + |10\rangle). \quad (4)$$

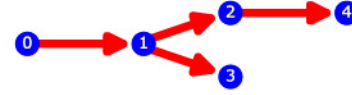
This is the desired “cat state,” up to a global phase. Changing the sign of $\hat{\sigma}_z \hat{\sigma}_z$ generates the state $(|11\rangle + |00\rangle)/\sqrt{2}$. In general, a unitary $\hat{U}_{ij}^\pm$ written as a directed arrow between two vertices

$$\hat{U}_{ij}^\pm = e^{i\pi\sigma_x^i/4} e^{\mp i\pi\sigma_z^i\sigma_z^j/4} \quad (5)$$


evolves an initially unentangled X product state into a maximally entangled Bell pair with either the same (+) or opposite (−) correlation. It is simple to generalize that products

of these unitaries along a directed tree will generate Z cat states [12]. For example

$$\hat{U}_{2,4}^- \hat{U}_{1,2}^- \hat{U}_{1,3}^+ \hat{U}_{0,1}^- |+\rangle = \frac{-1}{\sqrt{2}} (|01011\rangle + |10100\rangle). \quad (6)$$



If these unitaries map to a directed spanning tree of $\mathcal{G}$, they may prepare any eigenstate z of (3), by choosing signs $\hat{U}^\pm$ of each directed edge depending if the vertices are in the same (+) or opposite (−) partitions. These unitaries are Clifford: (4) is equivalent to a Hadamard on register 1, then a CNOT between registers 0 and 1.

This is a unitary version of the spanning tree algorithm of [22], if one chooses the sign of each unitary to be the sign of the signature of its edge. Given some bipartition z and associated spanning tree $\mathcal{T}_z$, one can construct the state as an ordered product of these unitaries over directed edges (up to a global phase)

$$\frac{|z\rangle + |\bar{z}\rangle}{\sqrt{2}} = \mathcal{T} \prod_{(ij) \in \mathcal{T}_z} \hat{U}_{ij}^{\sigma_{ij}} |+\rangle. \quad (7)$$

Here, $\mathcal{T}$ denotes inverse path ordering of unitaries along the directed spanning tree $\mathcal{T}_z$, and σ_{ij} is the signature of edge (ij) in signed graph Γ . Note that not every bipartition z may be constructed in this manner, as the spanning tree requires the reduced balanced subgraph $\Gamma \setminus e$ to be connected. We call this algorithm “spanning tree.”

This algorithm is exact in the following case. As part of the precomputation step, some classical algorithm finds the optimal partition z and an associated spanning tree $\mathcal{T}_z$. Then, implement the circuit of (7) to generate a maximal eigenstate of the objective function. However, this exactness comes at the cost that *generating* the sequence of gates is classically hard. The classical algorithm that creates the optimal circuit must first find the spanning tree(s) whose bipartition provides the solution to the SIGNED MAXCUT problem, which is known to be NP-hard.¹ This demonstrates a case in which their exist exact quantum circuits that provide solutions to hard problems, generating the circuit may itself be a hard problem classically.

The spanning tree algorithm can be made variational by allowing the unitaries of the directed edges of any spanning tree to be a function of angles

$$\hat{U}_{ij}^\pm(\gamma, \beta) = e^{i\beta\hat{\sigma}_x^i} e^{\mp i\gamma\hat{\sigma}_z^i\hat{\sigma}_z^j}. \quad (8)$$

The variational algorithm is as follows. For a signed graph Γ of N vertices, a classical algorithm generates a random

¹SIGNED MAXCUT can be reduced to MAXCUT by replacing every positive edge with two negative edges and a connectivity 2 vertex.

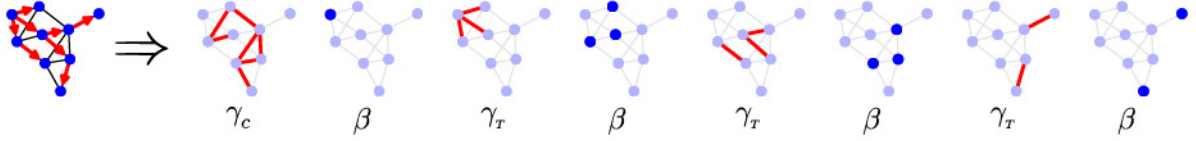


FIGURE 3. Example round of ST-QAOA. A classical algorithm computes some approximately optimal SIGNED MAXCUT solution by finding an appropriate spanning tree (left). A QAOA round alternates between $\sigma_z \sigma_z$ unitaries on edges (red) and σ_x unitaries on vertices (blue). First, the complement unitary is applied, then the tree unitary, which mimics the directed tree graph via repeated application of $\sigma_z \sigma_z$ and σ_x . This is repeated r times. For $r = 1$ this is guaranteed to match the performance of the classical algorithm, and for $r \rightarrow \infty$ the approximation ratio approaches 1.

spanning tree $\mathcal{T}$ and outputs a circuit, which is a function of $2N - 2$ angles

$$\hat{U}(\vec{\gamma}, \vec{\beta}) = \mathcal{T} \prod_{q \in \langle ij \rangle \in \mathcal{T}_z} \hat{U}_{ij}^{\pm}(\gamma_q, \beta_q) \quad (9)$$

where q indexes the edges of the tree, and $\mathcal{T}$ indicates the path ordering of unitaries along the randomly chosen directed spanning tree. We call this algorithm “variational spanning tree” (VST).

By (5), extremal values $\gamma_q \in \pm\pi/4$ and $\beta_q = \pi/4$ can construct any state $(|z\rangle + |\bar{z}\rangle)/\sqrt{2}$. It is the job of a classical optimizer to optimize the angles and find $N - 1$ signs of γ_p , which construct the optimal state. However, this VST algorithm is classical. In the Heisenberg picture, expectation values of operators are

$$\langle \hat{\sigma}_z^i \hat{\sigma}_z^j \rangle = \prod_{q \in \text{path } i \leftrightarrow j} \sin(2\gamma_q) \sin(2\beta_q) \quad (10)$$

where q index all of the edges of the spanning tree on the unique path between vertices i and j . Thus, the expectation value of the objective function for any graph may be computed classically. Bitstrings can also be efficiently sampled using tensor networks [13] by recursively contracting leaves of the spanning tree. Hence, VST is purely classical and cannot exhibit any quantum advantage. Instead of generating the circuit being NP-hard, the optimization itself is NP-hard [3], [9], [38]. Ultimately, this is because the optimization algorithm is a discrete search, finding $N - 1$ signs $\gamma = \pm\pi/4$ or analogously the optimal bitstring z . VST demonstrates that while circuit generation may be easy, optimizing parameters in and of itself may be a hard problem classically.

III. SPANNING TREE QAOA

It is reasonable to expect that if the number of variational parameters is constant in problem size, the optimization is more efficient. QAOA has this property [11] due to having a more constrained ansatz. Instead of choosing individual terms $\hat{\sigma}_z \hat{\sigma}_z$ and $\hat{\sigma}_x$ acting in serial, QAOA alternates between acting with the objective function $\hat{C}$ and a sum of Pauli $\hat{\sigma}_x$ terms $\hat{B}$. The number of variational angles is $2p$, independent of problem size

$$|\gamma, \beta\rangle = e^{-i\hat{B}\beta_p} e^{-i\hat{C}\gamma_p} (\dots) e^{-i\hat{B}\beta_1} e^{-i\hat{C}\gamma_1} |+\rangle. \quad (11)$$

There has been much work on QAOA. The approximation ratio uniformly increases in p , with the $p \rightarrow \infty$ limit converging to the exact state with an adiabatic schedule [8],

[11], [42]. For large p , it has been observed that the optimal parameters exhibit concentration and become independent of graph instance [1], and optimal parameters for p can induce the parameters for $p + 1$ [48]. These facts suggest that the classical optimization may be efficient. This comes at the cost that performance guarantees are combinatorially difficult to compute [11], [41], and to date QAOA has not outperformed the best classical algorithms [11], [41]. QAOA can be contrasted with the VQAs we define above, which can generate eigenstates of the objective function, including the maximal state, at expense of generating the gate set, or the optimization procedure, being NP-hard.

We now propose a combination of the spanning tree algorithm and QAOA, which we call ST-QAOA. To avoid the difficulty of parameter optimization, the circuit optimizer will use a nonextensive number of variational parameters, like QAOA. To avoid the difficulty of circuit design, instead of finding the exact maximal spanning tree, the circuit construction will use an *approximate* solution from a classical MAXCUT solver to generate spanning trees. Finally, to mimic QAOA, the ansatz will repeat p times, and be able to reproduce QAOA as a special case. The ST-QAOA is as follows.

Given some signed graph Γ , a classical MAXCUT solver P outputs some bitstring $P(\Gamma) = z$ and associated directed spanning tree $\mathcal{T}_z$ with root vertex v . Next, partition the edges of the graph $\mathcal{G}$ into those in the spanning tree and its complement to define a ST-QAOA round

$$\hat{U}(\gamma_c, \gamma_T, \beta) \equiv \left(\mathcal{T} \prod_{\langle ij \rangle \in \mathcal{T}_z} \hat{U}_{ij}^{\sigma_{ij}}(\gamma_T, \beta) \right) e^{i\beta \hat{\sigma}_x^v} \left(\prod_{\langle ij \rangle \notin \mathcal{T}_z} e^{i\gamma_c \sigma_{ij} \hat{\sigma}_z^i \hat{\sigma}_z^j} \right). \quad (12)$$

An example of this circuit is shown in Fig. 3. This applies a unitary generated by the complement edges of the spanning tree, then the spanning tree unitary with some global angle. An ST-QAOA procedure repeats this unitary p times as a

²Any classical algorithm can find solutions where the reduced balanced subgraph $\Gamma \setminus e$ is connected and so will have an associated spanning tree by adding the following subroutine: for all vertices within some disconnected subgraph of $\Gamma \setminus e$, swap the bipartition $X \leftrightarrow Y$. This will satisfy all of the edges of the original graph between the two previously disconnected subgraphs, increasing the number of satisfied edges and making the reduced balanced graph connected.

function of $3p$ variational parameters

$$|\psi\rangle = \hat{U}_{\text{ST-QAOA}}|+\rangle = \prod_{q=1}^r \hat{U}(\gamma_c^q, \gamma_T^q, \beta^q)|+\rangle. \quad (13)$$

Like QAOA, this algorithm has a number of parameters independent of the problem size. Unlike QAOA, the ansatz depends on the approximate classical solution, and can generate extensive correlations even for $r = 1$. Such a circuit includes all terms in the objective function and mixing term, except reordered to include the structure of spanning trees, which allows the algorithm to include QAOA as a special case. Let us now inspect the performance of this algorithm.

For $r = 1$, ST-QAOA can return the bitstring z produced by the classical subroutine P . By choosing the angles $\gamma_c = 0$, $\gamma_T = \pi/4$, $\beta = \pi/4$, the unitary is equivalent to (7) for the particular choice of spanning tree generated by P , and so ST-QAOA can give the same solution as the classical subroutine. Therefore, ST-QAOA has the same performance guarantee as its classical subroutine.

For $r = 2$, it is possible to reproduce a round of QAOA. Given QAOA angles γ_* and β_* , for the first round, choose angles $\gamma_c = \gamma_T = \gamma_*$, and $\beta = 0$. For the second round, choose angles $\gamma_c = \gamma_T = 0$ and $\beta = \beta_*$. The first round of unitaries is equivalent to the unitary generated by the objective function, as each $\sigma_z \sigma_z$ term commutes. This is the reason why the complement unitary, which includes all edges not in the spanning tree, must be added to the ansatz. The second round of unitaries is equivalent to the unitary generated by the mixing function for the same reason. Thus, the approximation ratio of level $2r$ ST-QAOA will always be *at least* that of level p QAOA. As p increases for QAOA, the approximation ratio increases, approaching 1 in the $p \rightarrow \infty$ limit [11]. Because the ST-QAOA includes QAOA as a special case, ST-QAOA will also approach the exact result as $r \rightarrow \infty$.

ST-QAOA is a useful algorithm with which to interrogate the possibility of various forms of quantum advantage. Can ST-QAOA exceed the performance of the best classical algorithms for worst case graphs? This would be the case if further variational optimization is possible in ST-QAOA for all graphs when the best classical algorithm is used as a subroutine, and would represent quantum advantage. Such a case cannot be established numerically, and may not be possible due to complexity-theoretic constraints [2], [21], [24], [25]. A simpler question is whether, above some threshold value of r , there exist subsets of graphs for which ST-QAOA has strictly better performance. This is not quantum advantage, as it only shows improved performance relative to a particular algorithm for a subset of graphs, which imposes additional structure that could be exploited by another specialized classical or quantum algorithm.

To make a quantitative comparison of algorithms we use the performance ratio

$$B(\Gamma) = C_Q(\Gamma) / C_C(\Gamma) \quad (14)$$

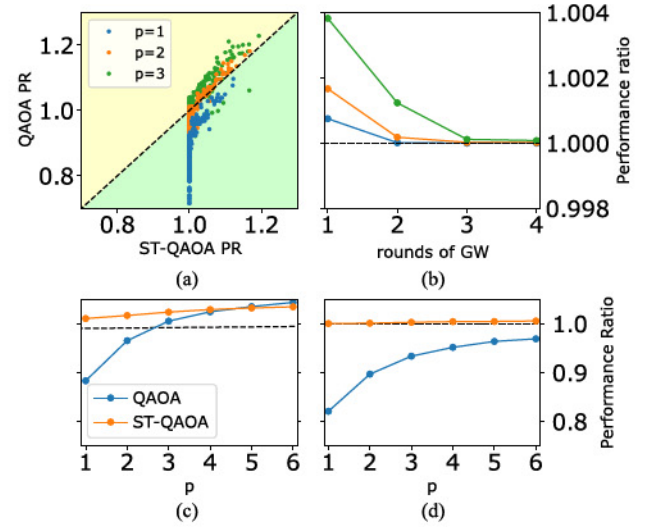


FIGURE 4. Comparing ST-QAOA, QAOA, random spanning tree (RST), and GW algorithms for an ensemble of 250 random 3-regular graphs with 16 vertices. (a) Plots the performance ratio $B(\Gamma)$, comparing ST-QAOA (horizontal axis) and QAOA (vertical axis) for each graph in the ensemble. It is clear that the ST-QAOA has a performance guarantee $B(\Gamma) \geq 1$ for $p \geq 1$. (b) Plots the average performance ratio versus the number of repetitions of the GW algorithm. As GW is a randomized algorithm, repeated sampling uniformly increases its performance, and the performance ratio appears to converge to 1 from above. (c) and (d) Plots the average performance ratio over the ensemble versus the random spanning tree (c) and GW (d) algorithms. QAOA has advantage over the random algorithm (black dashed) with $p \geq 3$ and advantage over ST-QAOA with the random algorithm for $p \geq 5$.

where $C_Q(\Gamma)$ is the optimized expectation value of the objective function for a VQA applied to the signed graph Γ , and $C_C(\Gamma)$ is the number of satisfied edges given an output from the competing classical algorithm P . A value $B(\Gamma) > 1$ indicates that the quantum algorithm can find better solutions than the classical algorithm for particular problem instance Γ . If $\langle B \rangle > 1$ for graphs in some ensemble $\{\Gamma\}$, then the quantum algorithm has average case quantum advantage over its classical competitor, as the quantum algorithm can produce better solutions than the classical algorithm in at least some of the graphs. For the ST-QAOA, the worst case $B(\Gamma) \geq 1$ for all graphs relative to the classical algorithm employed by ST-QAOA to generate the ansatz.

As a numerical demonstration, we implement ST-QAOA on an ensemble of 3-regular graphs. We choose two classical algorithms to generate spanning trees. The first is that of Goemans and Williamson (GW) [18], which uses semidefinite programming to achieve an approximation ratio of at least 0.878 in general graphs, and 0.932 in 3-regular graphs [23]. The second algorithm samples a random spanning tree and achieves an approximation ratio of $2/3(1 - 1/n)$ for a 3-regular graph with n -vertices [33]. Numerical results for an ensemble of 250 3-regular graphs with 16 vertices are shown in Fig. 4. Optimization used gradient ascent initialized from 100 random points in parameter space. From Fig. 4(a), it is clear that $B(\Gamma) \geq 1$ for ST-QAOA and that there exist graph instances for which the ST-QAOA exhibits advantage over its competing classical algorithm and performs better

than QAOA. Fig. 4(c) and (d) shows that the average performance ratio $\langle B(\Gamma) \rangle \geq 1$ for all p as expected, indicating an average case advantage for the spanning tree algorithm over its classical subroutine. Fig 4(b) shows that care must be taken to ensure that the classical algorithm is also performing optimally by illustrating the convergence of average performance ratio to one with increased number of rounds of GW. Each round of GW improves the classical solution quality, reducing the room for improvement from ST-QAOA. It may be the case that the improvement over GW is only due to atypically poor individual solutions. Nonetheless, the performance advantage is typically extremely low, suggesting that GW may be a close to optimal algorithm; further study is beyond the scope of this work. Fig. 4(b) illustrates that careless use of classical algorithms can create the illusion of quantum advantage.

IV. CONCLUSION

In this article, we have expanded VQA with problem instance-specific circuits precomputed by a classical subroutine. We also highlight the computational complexity pitfalls, which may arise from such constructions. One may design an algorithm whose circuit yields exact answers to MAXCUT, at the cost that the algorithm is NP-hard, as exemplified by the spanning tree algorithm. Similarly, an algorithm which constructs the circuit may be simple at the cost that the variational optimization algorithm is NP-hard, as exemplified by the VST algorithm. The intermediate algorithm, the ST-QAOA, combines QAOA with the concept of spanning trees to improve approximate classical solutions.

However, the ST-QAOA requires the use of its competition as a subroutine to generate the circuit. In practice, any classical algorithm could integrate a similar scheme by running multiple algorithms in parallel and choosing the more optimal result, or use additional classical algorithms to improve the output of one classical algorithm. This is clear in Fig. 4(b), when the optimal result among multiple GW queries is used as a classical solution. Because GW is randomized, the performance uniformly increases and removes any additional advantage from the ST-QAOA. Due to this subtlety, we make no claims of quantum advantage, even though the ST-QAOA can only increase the quality of solutions of its classical subroutine.

While we focus on the problem MAXCUT using the concept of spanning trees, the precomputation step is more general. Using these ideas, constructing VQAs that take advantage of the precomputation step to nontrivially generate problem instance-specific circuits may improve heuristic and guaranteed performance on the limited quantum resources of today's NISQ devices.

REFERENCES

- [1] G. S. L. Fernando, M. Brandao, E. Broughton, S. F. Gutmann, and H. Neven, "For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances," 2018, *arXiv:1812.04170*.
- [2] P. Berman and M. Karpinski, "On some tighter inapproximability results (extended abstract)," in *Automata, Languages and Programming*. Berlin, Germany: Springer, 1999, pp. 200–209, doi: [10.1007/3-540-48523-6_17](https://doi.org/10.1007/3-540-48523-6_17).
- [3] L. Bittel and M. Kliesch, "Training variational quantum algorithms is NP-hard—Even for logarithmically many qubits and free fermionic systems," 2021, [Online]. Available: <http://dx.doi.org/10.1103/PhysRevLett.127.120502>.
- [4] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Sci. Technol.*, vol. 4, no. 4, Nov. 2019, Art. no. 043001, doi: [10.1088/2058-9565/ab4eb5](https://doi.org/10.1088/2058-9565/ab4eb5).
- [5] F. G. S. L. Brandao and K. Svore, "Quantum speed-ups for semidefinite programming," 2017, *arXiv:1609.05537*.
- [6] S. Bravyi, G. Smith, and J. A. Smolin, "Trading classical and quantum computational resources," *Phys. Rev. X*, vol. 6 no. 2, Jun. 2016, Art. no. 021043, doi: [10.1103/PhysRevX.6.021043](https://doi.org/10.1103/PhysRevX.6.021043).
- [7] M. Cerezo *et al.*, "Variational quantum algorithms," *Nature Rev. Phys.*, vol. 3, pp. 625–644, 2020, doi: [10.1038/s42254-021-00348-9](https://doi.org/10.1038/s42254-021-00348-9).
- [8] G. E. Crooks, "Performance of the quantum approximate optimization algorithm on the maximum cut problem," 2018, *arXiv:1811.08419*.
- [9] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost-function-dependent barren plateaus in shallow quantum neural networks," 2020, [Online]. Available: <http://dx.doi.org/10.1038/s41467-021-21728-w>.
- [10] D. J. Egger, J. Marecek, and S. Woerner, "Warm-starting quantum optimization," *Quantum*, vol. 5, 2020, Art. no. 479, doi: [10.22331/q-2021-06-17-479](https://doi.org/10.22331/q-2021-06-17-479).
- [11] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm. 2014," *arXiv:1411.4028*.
- [12] E. Farhi, J. Goldstone, S. Gutmann, and H. Neven, "Quantum algorithms for fixed qubit architectures," 2017, *arXiv:1703.06199*.
- [13] J. A. Ferris and G. Vidal, "Perfect sampling with unitary tensor networks," *Phys. Rev. B*, vol. 85, Apr. 2012, Art. no. 165146, doi: [10.1103/PhysRevB.85.165146](https://doi.org/10.1103/PhysRevB.85.165146).
- [14] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, "An adaptive variational algorithm for exact molecular simulations on a quantum computer," *Nature Commun.*, vol. 10, no. 1, Jul. 2019, Art. no. 3007, doi: [10.1038/s41467-019-10988-2](https://doi.org/10.1038/s41467-019-10988-2).
- [15] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," *Theor. Comput. Sci.*, vol. 1, no. 3, pp. 237–267, 1976, doi: [10.1016/0304-3975\(76\)90059-1](https://doi.org/10.1016/0304-3975(76)90059-1).
- [16] M. Grötschel and G. L. Nemhauser, "A polynomial algorithm for the max-cut problem on graphs without long odd cycles," *Math. Program.*, vol. 29, no. 1, pp. 28–40, May 1984, doi: [10.1007/BF02591727](https://doi.org/10.1007/BF02591727).
- [17] J. Gondzio, "Warm start of the primal-dual method applied in the cutting-plane scheme," *Math. Program.*, vol. 83, no. 1–3, pp. 125–143, Jan. 1998, doi: [10.1007/BF02680554](https://doi.org/10.1007/BF02680554).
- [18] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, Nov. 1995, doi: [10.1145/227683.227684](https://doi.org/10.1145/227683.227684).
- [19] F. Harary, "On the notion of balance of a signed graph," *Michigan Math. J.*, vol. 2, no. 2, pp. 143–146, 1953, doi: [10.1307/mmj/1028989917](https://doi.org/10.1307/mmj/1028989917).
- [20] A. W. Harrow, "Small quantum computers and large classical data sets," 2020, *arXiv:2004.00026*.
- [21] J. Håstad, "Some optimal inapproximability results," *J. ACM*, vol. 48, no. 4, pp. 798–859, 2001, doi: [10.1145/502090.502098](https://doi.org/10.1145/502090.502098).
- [22] F. Harary and J. A. Kabell, "A simple algorithm to detect balance in signed graphs," *Math. Social Sci.*, vol. 1, no. 1, pp. 131–136, 1980, doi: [10.1016/0165-4896\(80\)90010-4](https://doi.org/10.1016/0165-4896(80)90010-4).
- [23] E. Halperin, D. Livnat, and U. Zwick, "Max cut in cubic graphs," *J. Algorithms*, vol. 53, no. 2, pp. 169–185, 2004, doi: [10.1016/j.jalgor.2004.06.001](https://doi.org/10.1016/j.jalgor.2004.06.001).
- [24] S. Khot, "On the power of unique 2-prover 1-round games," in *Proc. 34th Annu. ACM Symp. Theory Comput.*, 2002, pp. 767–775, doi: [10.1145/509907.510017](https://doi.org/10.1145/509907.510017).
- [25] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, "Optimal inapproximability results for MAX-CUT and other 2-variable CSPs?" *SIAM J. Comput.*, vol. 37, no. 1, pp. 319–357, Apr. 2007, doi: [10.1137/S0097539705447372](https://doi.org/10.1137/S0097539705447372).
- [26] N. Moll *et al.*, "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Sci. Technol.*, vol. 3, no. 3, Jun. 2018, Art. no. 030503, doi: [10.1088/2058-9565/aab822](https://doi.org/10.1088/2058-9565/aab822).

- [27] J. R. McClean, S. Boixo, V.N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Commun.*, vol. 9, no. 1, Nov. 2018, Art. no. 4812, doi: [10.1038/s41467-018-07090-4](https://doi.org/10.1038/s41467-018-07090-4).
- [28] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New J. Phys.*, vol. 18, no. 2, Feb. 2016, Art. no. 023023, doi: [10.1088/1367-2630/18/2/023023](https://doi.org/10.1088/1367-2630/18/2/023023).
- [29] K. Poland, K. Beer, and Tobias J. Osborne, "No free lunch for quantum machine learning," 2020, *arXiv:2003.14103*.
- [30] T. Peng, A. W. Harrow, M. Ozols, and X. Wu, "Simulating large quantum circuits on a small quantum computer," *Phys. Rev. Lett.*, vol. 125, Oct. 2020, Art. no. 150504, doi: [10.1103/PhysRevLett.125.150504](https://doi.org/10.1103/PhysRevLett.125.150504).
- [31] A. Peruzzo *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nature Commun.*, vol. 5, no. 1, Jul. 2014, Art. no. 4213, doi: [10.1038/ncomms5213](https://doi.org/10.1038/ncomms5213).
- [32] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, Aug. 2018, Art. no. 79, doi: [10.22331/q-2018-08-06-79](https://doi.org/10.22331/q-2018-08-06-79).
- [33] S. Poljak and D. Turzik, "A polynomial time heuristic for certain subgraph optimization problems with guaranteed worst case bound," *Discrete Math.*, vol. 58, no. 1, pp. 99–104, 1986, doi: [10.1016/0012-365X\(86\)90192-5](https://doi.org/10.1016/0012-365X(86)90192-5).
- [34] S. Poljak and Z. Tuza, "Maximum cuts and largest bipartite subgraphs," in *Combinatorial Optimization*. Providence, RI, USA: Amer. Math. Soc., 1993, doi: [10.1090/dimacs/020/04](https://doi.org/10.1090/dimacs/020/04).
- [35] I. G. Ryabinkin, R. A. Lang, S. N. Genin, and A. F. Izmaylov, "Iterative qubit coupled cluster approach with efficient screening of generators," *J. Chem. Theory Comput.*, vol. 16, no. 2, pp. 1055–1063, 2020, doi: [10.1021/acs.jctc.9b01084](https://doi.org/10.1021/acs.jctc.9b01084).
- [36] K. Sharma, M. Cerezo, Z. L. Holmes, A. C. Sornborger, and P. J. Coles, "Reformulation of the no-free-lunch theorem for entangled data sets," 2020, *arXiv:2007.04900*.
- [37] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Phys. Rev. Lett.*, vol. 122, Feb. 2019, Art. no. 040504, doi: [10.1103/PhysRevLett.122.040504](https://doi.org/10.1103/PhysRevLett.122.040504).
- [38] R. Shaydulin, I. Safro, and J. Larson, "Multistart methods for quantum approximate optimization," in *Proc. IEEE High Perform. Extreme Comput. Conf.*, 2019, pp. 1–8, doi: [10.1109/HPEC.2019.8916288](https://doi.org/10.1109/HPEC.2019.8916288).
- [39] R. Tate, M. Farhadi, C. Herold, G. Mohler, and S. Gupta, "Bridging classical and quantum with SDP initialized warm-starts for QAOA," 2020, *arXiv:2010.14021*.
- [40] N. M. Tubman *et al.*, "Postponing the orthogonality catastrophe: Efficient state preparation for electronic structure simulations on quantum devices," 2018, *arXiv:1809.05523*.
- [41] J. Wurtz and P. Love, "MaxCut quantum approximate optimization algorithm performance guarantees for $p > 1$," *Phys. Rev. A*, vol. 103, Apr. 2021, Art. no. 042612, doi: [10.1103/PhysRevA.103.042612](https://doi.org/10.1103/PhysRevA.103.042612).
- [42] J. Wurtz and Peter J. Love, "Counterdiabaticity and the quantum approximate optimization algorithm," 2021, *arXiv:2106.15645*.
- [43] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [44] T. Zaslavsky, "Signed graphs," *Discrete Appl. Math.*, vol. 4, no. 1, pp. 47–74, 1982, doi: [10.1016/0166-218X\(82\)90033-6](https://doi.org/10.1016/0166-218X(82)90033-6).
- [45] T. Zaslavsky, "A mathematical bibliography of signed and gain graphs and allied areas," *Electron. J. Combinatorics*, vol. 1000, Dec. 2018, doi: [10.37236/29](https://doi.org/10.37236/29).
- [46] T. Zaslavsky, "Negative (and positive) circles in signed graphs: A problem collection," *AKCE Int. J. Graphs Combinatorics*, vol. 15, pp. 31–48, 2018, doi: [10.1016/j.akcej.2018.01.011](https://doi.org/10.1016/j.akcej.2018.01.011).
- [47] L. Zhu *et al.*, "An adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer," 2020, *arXiv:2005.10258*.
- [48] 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*, vol. 10 no. 2, Jun. 2020, Art. no. 021067, doi: [10.1103/PhysRevX.10.021067](https://doi.org/10.1103/PhysRevX.10.021067).