Approaches to Constrained Quantum Approximate Optimization

Zain H. Saleem,[1,*] Bilal Tariq,[1,2,†] and Martin Suchara[1,‡]

[1] *Argonne National Laboratory, 9700 S. Cass Ave., Lemont, IL 60439, USA.*
[2] *Department of Physics, University at Buffalo, SUNY, Buffalo, NY 14260, USA.*
(Dated: October 14, 2020)

We analyze different quantum approaches to finding the approximate solutions of constrained combinatorial optimization problems. The Maximum Independent Set (MIS) problem is used as a toy example for pointing out the strengths and weaknesses of the existing approaches. We also introduce a new algorithm based on a "Dynamic Quantum Variational Ansatz" (DQVA) for the MIS problem that dynamically reduces the depth of the circuit used in preparing the variational ansatz employed in the quantum optimization. The approaches used in our analysis and our proposed algorithm can be also applied to other constrained combinatorial optimization problems.

I. INTRODUCTION

Rapid progress is being made in building quantum hardware with increasing qubit counts and fidelities [1–5]. However the capabilities of quantum hardware are still limited, and therefore the task of developing and optimizing algorithms that can be solved on these near-term intermediate scale noisy quantum devices [6, 7] is important. NP-hard combinatorial optimization [8] is an example of problems that have been gaining attention because of their potential to demonstrate quantum advantage. There are two types of these optimization problems: constrained and unconstrained. Unconstrained combinatorial optimization problems minimize an objective function that depends on discrete variables with no restrictions on their values, whereas in constrained optimization problems there are restrictions on the values.

Quantum Approximate Optimization Algorithm (QAOA) [9] is a variational algorithm that uses both classical and quantum resources to find the approximate solution to combinatorial optimization problems. There are two main components of QAOA, the objective function that encodes the solution of the problem in its ground state and the variational ansatz that is a quantum circuit that allows us to vary over a specified space of solutions. The algorithm attempts to find the best solution by by classically optimizing the expectation value of the objective function in the variational ansatz. Extensive work has been published on applying QAOA to unconstrained optimization problems [10–12]; however, much less attention has been paid to constrained problems [13–16]. For the latter class of problems there are two ways of imposing constraints in QAOA. We can impose these constrains by adding a penalty term to the objective function of our problem. This essentially converts our constrained optimization problem into an unconstrained one. The algorithm outputs all the possible solutions with the feasible ones having a higher probability of appearing in the output. A pruning step is used to find the best solution. This approach was studied for the maximum independent set problem in [17, 18] and was referred to as QAOA+. The other method, which is introduced in [14, 15] is called the Quantum Alternating Operator Ansatz (QAO-Ansatz). It involves constructing our variational ansatz in such a way that the constraints are satisfied at all times and the extremization is performed only over the space of feasible solutions.

Both the QAOA+ and Quantum Alternating Operator Ansatz approaches have its advantages and disadvantages. The QAOA+ algorithm requires fewer quantum resources during its execution; however, it requires pruning. The amount of pruning needed depends on the size of the penalty term, and there is no obvious choice how large the penalty should be. The QAO-Ansatz requires larger quantum circuits that must be evaluated but has advantages of not requiring any pruning and guaranteeing a feasible state in the output. To tackle the large quantum resource requirement of executing the QAO-Ansatz algorithm, this paper proposes a Dynamical Quantum Variational Ansatz (DQVA) that dynamically reduces the depth of quantum circuits used in the ansatz.

In Section II and III we introduce the quantum approximate optimization algorithm and the maximum independent set problem respectively. We then analyze and discuss QAOA+ and QAO-Ansatz in Sections IV and V, respectively. In Section VI we present our new algorithm for solving constrained combinatorial optimization problems. Section VII concludes and suggests future directions.

II. QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

QAOA solves combinatorial optimization problems by converting a classical optimization problem formulation into the problem of characterizing a quantum operator. The graph dependent classical objective function $C(\mathbf{b})$ which we are looking to optimize is defined on n-bit strings $\mathbf{b} = \{b_1, b_2, b_3 \dots b_n\} \in \{0,1\}^n$. It can be writ-

* zsaleem@anl.gov
† bilaltar@buffalo.edu
‡ msuchara@anl.gov

ten as a quantum operator diagonal in its computational basis as

$$C_{obj}|b\rangle = C(\mathbf{b})|b\rangle. \tag{1}$$

The main goal in QAOA is to optimize this operator in a variational ansatz. The variational ansatz comprises three main components:

1. Initial State $|s\rangle$: This is the state on which we act with unitary operators to build our variational ansatz.

2. Phase Separator Unitary $e^{i\gamma C}$: C is a diagonal operator in computational basis related to the problem at hand and is usually the same as the objective operator. This unitary plays the role of a phase separator between successive applications of the mixing operator. γ is a variational parameter with values between $[0, 2\pi]$.

3. Mixing Unitary $e^{i\beta M}$: M can be nondiagonal in computation basis, and its job is to mix the states among each other during the optimization. β is also a variational parameter with values between $[0, 2\pi]$.

The variational ansatz is made by combining these three components and is

$$|\psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = e^{-i\beta_p M} e^{-i\gamma_p C} \dots e^{-i\beta_1 M} e^{-i\gamma_1 C}|s\rangle, \tag{2}$$

where p controls the number of times the unitary operators are applied. The expectation value of C_{obj} in this variational state,

$$E_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta})|C_{obj}|\psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle, \tag{3}$$

is found on a quantum computer and is passed to a classical optimizer to find the optimal parameters that extremize $\max_{\boldsymbol{\gamma},\boldsymbol{\beta}} E_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$. Since the eigenstates of C_{obj} are computational basis states, this maximization is achieved for the states corresponding to the solutions of the original classical problem.

III. MAXIMUM INDEPENDENT SET

The Maximum Independent Set problem is one of Karp's 21 NP-hard [19] computational problems. This is the problem we will study using QAOA, and it is formulated as follows. Consider a graph $G = (V, E)$ where V is the set of nodes of the graph and E is the set of edges. A subset V' of V is represented by a vector $\mathbf{x} = (x_i) \in \{0, 1\}^{|V|}$, where $x_i = 1$ indicates i is in the subset and $x_i = 0$ indicates i is not in the subset. A subset $\mathbf{x}$ is called an independent set if no two nodes in the subset are connected by an edge: $(x_i, x_j) \neq (1, 1)$ for all $(i, j) \in E$. The maximum independent set (MIS) is the independent set with the largest number of nodes. We are interested in finding a MIS $\mathbf{x}^*$. This optimization problem has the following integer programming formulation:

$$\begin{aligned} &\text{maximize} \sum_{i \in V} \mathbf{x}_i \\ &\text{subject to } \; x_i x_j \neq 1 \; \text{ where } \; (i, j) \in \text{E}. \end{aligned} \tag{4}$$

Fig. 1 shows a pictorial representation of the problem. The MIS is an NP-hard problem, and the best that we can do is find an approximate solution. To find this approximation using QAOA, we start by encoding all the states of the graph in the Hilbert space of our qubits and relate the classical objective function to a quantum operator via Eq. 1. After this encoding we proceed with executing the quantum algorithm.

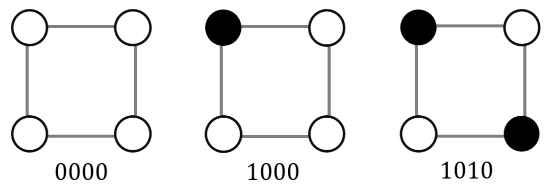

FIG. 1. Square ring graph. The three strings 0000, 1000, and 1010 represent the three possible independent sets of size 0, 1, and 2, respectively. 1010 is the string representing the maximum independent set of the graph.

IV. PENALTY TERM OPTIMIZATION: QAOA+

The task of finding the maximum independent set in QAOA is essentially equivalent to finding the state with the maximum Hamming weight subject to the constraint that no two nodes (encoded as qubits) in the state share an edge. In QAOA+, the constraint is imposed at the objective function level by adding a penalty term that decreases its value whenever two nodes have a common edge. The objective operator is therefore given by

$$C_{obj} = H - \lambda P = \sum_{i \in V} b_i - \lambda \sum_{i,j \in E} b_i b_j, \tag{5}$$

where

$$b_i = \frac{1}{2}(1 - Z_i) \tag{6}$$

and Z is the Pauli-Z operator. H gives us the Hamming weight of the state it acts on, and P is the term that penalizes every time two nodes in a state share an edge. λ is the penalty factor that controls the contribution coming from the penalty term. This effectively reduces our constrained problem to an unconstrained one in the sense that now the optimization will be performed over all the bit strings $\{0, 1\}^n$ during the variational optimization.

The three components of the variational ansatz for QAOA+ are as follows:

1. Initial State: $|s\rangle = |++\cdots+\rangle$.
2. Phase Separator Unitary: The phase separator unitary is $U_C(\gamma) := e^{i\gamma P}$.
3. Mixing Unitary: The mixing unitary is $U_M(\beta) := e^{i\beta\sum_i X_i}$, where X is the Pauli-X operator.

All components of this algorithm require at most nearest neighbor interactions and therefore it can be implemented with a relatively low-depth circuit if p is not too high. The number that quantifies the performance of the algorithm is the approximation ratio R, defined as the maximum value of the expectation value of the objective function C_{obj} in the variational ansatz.

$$R_p = \max_{\boldsymbol{\gamma},\boldsymbol{\beta}} \; E_p(\boldsymbol{\gamma},\boldsymbol{\beta}) \tag{7}$$

Because of the simplicity of our objective function, this is a number we can potentially calculate for $p = 1$. We have

$$\begin{aligned} E_1 &= \langle s|U_C^\dagger(\gamma)U_M^\dagger(\beta)|C_{obj}|U_M(\beta)U_C(\gamma)|s\rangle \\ &= \langle H\rangle - \lambda\langle P\rangle \end{aligned} \tag{8}$$

We see that the approximation ratios will depend on λ, the penalty factor, and we need to figure out a way to select its optimal value. If the penalty factor is too big, we will reduce the probability of outputting strings with a high Hamming weight. If λ is too small, we will be increasing the probability of outputting strings with large Hamming weight, but we will be decreasing their probability of being independent sets. We are studying the optimal selection of the penalty factor in separate work via simulation and analytics.

V. QUANTUM ALTERNATING OPERATOR ANSATZ

In quantum alternating operator ansatz we impose constraints at the variational ansatz level instead of in the objective function. Here we build the ansatz in such a way that we never leave the set of feasible states during the variational optimization. The objective function here is just the Hamming weight operator,

$$C_{obj} = H = \sum_{i\in V} b_i. \tag{9}$$

The three components of the variational ansatz are as follows:

1. Initial State: This can be any feasible state or the superposition of the feasible states.
2. Phase Separator Unitary: The unitary is $U_C(\gamma) := e^{i\gamma H}$ and P is the penalty term.
3. Mixing Unitary: The mixing unitary $U_M(\beta) := \prod_i e^{M_i}$, where $M_i = X_i\bar{B}$ and we have defined

$$\bar{B} := \prod_{j=1}^{\ell} \bar{b}_{v_j}, \quad \bar{b}_{v_j} = \frac{1+Z_{v_j}}{2}, \tag{10}$$

where v_j are the neighbors and ℓ is the number of neighbors for the ith node. We can also write our mixer as

$$U_M(\beta) = \prod_{i=1}^{n} V_i(\beta) = \prod_{i=1}^{n} \left(I + (e^{-i\beta X_i} - I)\,\bar{B}\right), \tag{11}$$

where we have used $\bar{b}_{v_j}^2 = \bar{b}_{v_j}$. The unitary mixer above is a product of n partial mixers V_i, in general not all of which commute with each other $[V_i, V_j] \neq 0$. The partial mixers are executed on a digital quantum computer by the circuit (see Fig. 2). We have the freedom of choosing the ordering of these partial mixers in the product, and different orderings can have distinct outputs for different problem instances. Our variational ansatz therefore is defined up to a permutation

$$U_M(\beta) \simeq \mathcal{P}\big(V_1(\beta)V_2(\beta)\cdots V_n(\beta)\big), \tag{12}$$

where $\mathcal{P}$ is the permutation's function of labels from 1 to n and the $\simeq$ symbol represents that the mixer is defined up to permutations of V_i. This is the permutation freedom in the ansatz. It is easy to see that there are $n!$ choices, not all of which are distinct for a particular graph.

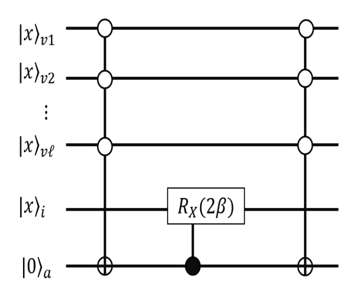

FIG. 2. V_i are implemented on a digital quantum computer via a multicontrolled Toffoli gate. $|x\rangle_i$ is the qubit on which we are applying the partial unitary, and $|x\rangle_{v_1}$ to $|x\rangle_{v_l}$ are the neighbors of the ith qubit. We can throw away the ancilla $|0\rangle_a$ after applying V_i.

Let us try to calculate the expectation value of our objective function at $p = 1$ depth for an initial state $|s\rangle$. We get

$$\begin{aligned} E_1 &= \langle s|U_C^\dagger(\gamma)U_M^\dagger(\beta)|C_{obj}|U_M(\beta)U_C(\gamma)|s\rangle \\ &= \frac{n}{2} - \langle s|U_C^\dagger(\gamma)U_M(\beta)|Z_u|U_M^\dagger(\beta)U_C(\gamma)|s\rangle. \end{aligned} \tag{13}$$

In the second term above we can expand out the operator in the expectation value as

$$U_M(\beta)^\dagger Z_i U_M(\beta) = C_0(I,\beta) + C_1(Z,\beta) + C_2(Z,\beta)X_i + C_3(Z,\beta)X_iX_j + \cdots , \quad (14)$$

where the operators C are functions of only the Pauli-I and Pauli-Z operator. One can easily see that we will not be able to get a nontrivial γ dependence for any choice of an initial state such as $|s\rangle = |f\rangle$, where f indicates a feasible set. The reason is that the γ dependence factors out:

$$\langle f|U_C(\gamma)^\dagger U_M(\beta)^\dagger Z_i U_M(\beta) U_C(\gamma)|f\rangle = g(\beta). \quad (15)$$

However, for a good choice of ordering if we choose our initial state to be

$$|s'\rangle = \sum_f |f\rangle, \quad (16)$$

which is a sum of different feasible states, we are more likely to get a nontrivial gamma dependence.

$$\sum_f \langle f|U_C(\gamma)^\dagger U_M(\beta)^\dagger Z_i U_M(\beta) U_C(\gamma)|f\rangle = g(\beta,\gamma). \quad (17)$$

The γ dependence will not necessarily factor out depending on the ordering chosen. The reason is that terms that are products of Pauli-X and Pauli-Y operators in the expansion in Eq. (14) can give a finite amplitude depending on what ordering one chooses. What we have shown is that for small-depth circuits $|s'\rangle$ is a better choice of an initial state than $|f\rangle$. Preparing arbitrary initial states is costly, however. An interesting choice in this regard [13] is the state

$$W = \frac{1}{\sqrt{N}} \left(|100..0\rangle + |010..0\rangle + \cdots |00..1\rangle\right), \quad (18)$$

which is the sum of all the single-node feasible states and can be prepared with $\mathcal{O}(n)$ CNOT gates[20, 21]. We can also circumvent this problem by going to $p = 1.5$.

VI. DYNAMIC QUANTUM VARIATIONAL ANSATZ

We propose a new algorithm that takes advantage of the ordering freedom in our ansatz. We will restrict ourselves to the $p = 1.5$ version of the algorithm, which is the minimum depth we need in order to take full advantage of all the variational parameters in the ansatz, as shown in the preceding section. The steps we describe below for the DQVA algorithm can be generalized to higher values of p.

Step 1: Jump Start. Run a classical algorithm to find a collection of bit strings representing independent sets. A trivial choice for this collection of strings can be the bit strings representing single nodes such as $|00..1\rangle$, $|01..0\rangle$, and $|00..1\rangle$. In order to take full advantage of classical resources, however, it is advantageous to use a polynomial time classical approximate algorithm to find a set of larger Hamming weight strings representing independent sets. We will use these strings as our initial state. Let us call this set I_{cl} and let

$$I_{cl} = \{|\mathbf{c}_1\rangle, |\mathbf{c}_2\rangle \cdots |\mathbf{c}_m\rangle\}, \quad (19)$$

where $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{c}_m$ are strings obtained by running the classical algorithm. We can prepare states $|\mathbf{c}_i\rangle = |c_i^1 c_i^2 \cdots c_i^n\rangle$ with $\mathcal{O}(1)$ depth circuits, where $c_i^j \in \{0,1\}$ represents the jth bit of the string.

Step 2: Mixer Initialization. Select any of these strings, say, $|\mathbf{c}_1\rangle$, and apply two mixing unitaries $U_M(\alpha_1)$ and $U_M(\alpha_2)$ separated by the phase separator $U_C(\gamma)$ between them.

$$|\alpha_1, \gamma, \alpha_2\rangle = U_M^2(\alpha_2) U_C(\gamma) U_M^1(\alpha_1) |c_1\rangle \quad (20)$$

We choose the parameters for the partial mixers to be all different.

$$U_M^k(\alpha_k) = \mathcal{P}\big(V_1^k(\alpha_k^1) V_2^k(\alpha_k^2) \cdots V_n^k(\alpha_k^n)\big), \quad k = 1, 2 \cdots p \quad (21)$$

This choice differs from QAO-Ansatz, where all variational parameters of the partial mixers were the same. In DQVA we do not need a phase separator between the partial mixers because they act on different qubits. This also adds more parameters to our variational ansatz. Wherever the jth string $c_1^j = 1$, we set the parameter in the corresponding partial mixer to be $\alpha_k^j = 0$. For example, if $|\mathbf{c}_1\rangle = |010010\rangle$, where we have 1's at the position 2 and 5, then the mixer unitaries will be

$$U_M^k(\alpha_k) = V_1^k(\alpha_k^1)\; I_2\; V_3^k(\alpha_k^3)\; V_4^k(\alpha_k^4)\; I_5\; V_6^k(\alpha_k^6). \quad (22)$$

Step 3: Dynamic Ansatz Update. Run the quantum approximate optimization algorithm for the choice of mixing unitaries from the previous step.

$$\langle \alpha_2, \gamma, \alpha_1 | C_{obj} | \alpha_1, \gamma, \alpha_2 \rangle \quad (23)$$

If this improves the Hamming weight and we get a new state $|\mathbf{q}_1\rangle$ with a Hamming weight larger than $|\mathbf{c}_1\rangle$, then we replace the initial state $|\mathbf{c}_1\rangle$ with $|\mathbf{q}_1\rangle$. We also update out mixing unitaries such that if $q_1^j = 1$, we set all the kth parameters of our partial mixers to zero: $\alpha_i^j = 0$. In the example above let us write the new state as $|\mathbf{q}_1\rangle = |010110\rangle$. Then we have

$$\tilde{U}_M^k(\alpha_k) = V_1^k(\alpha_k^1)\; I_2\; V_3^k(\alpha_k^3)\; I_4\; I_5\; V_6^k(\alpha_k^6). \quad (24)$$

We run the quantum optimization algorithm with this new state and updated unitaries $\tilde{U}$. As we set more and more parameters of the partial mixer to be zero, we can add another layer of mixing unitary U_M^3 to ensure maximal utilization of available quantum resources AND making sure that the Jth parameters in all mixing unitaries are turned off: $\alpha_1^j = \alpha_2^j = \alpha_3^j = 0$. We repeat this step until we can no longer increase the Hamming weight.

Step 4: Randomization. If we are not able to increase the Hamming weight of the state at the above step, then randomize over the position of the partial unitaries that have not been set equal to identity. We can set an upper limit on the number of times we do these randomizations, say m. In our example this randomization operation $\mathcal{R}$ could give

$$R(\tilde{U}_M^k(\alpha_k)) = V_3^k(\alpha_k^3)\ I_2\ V_6^k(\alpha_k^6)\ I_4\ I_5\ V_1^k(\alpha_k^1). \quad (25)$$

We run the quantum optimization with this new mixer. After we have performed r randomizations, we store the state $\mathbf{q}_1^r$. Note that we can perform some randomizations at Step 3 as well, but here we are doing so at the end for simplicity.

Step 5: Output and Repeat. Repeat Steps 1–4 with the other initial states $|\mathbf{c}_i\rangle$, and obtain the set

$$I_{qu} = \{|\mathbf{q}_1^r\rangle, |\mathbf{q}_2^r\rangle \cdots |\mathbf{q}_m^r\rangle\}. \quad (26)$$

This set has the state $|\mathbf{q}_i^r\rangle$ with the largest Hamming weight among the states in the set. This state will be our final result.

The DQVA algorithm takes maximum advantage of the existing classical polynomial time approximation algorithms for the maximum independent set. Moreover, by "dynamically" turning off parameters in our optimization we reduce the amount of quantum resources employed by our algorithm as well.

VII. CONCLUSION AND FUTURE DIRECTIONS

In this work we have reviewed the existing approaches to constrained optimization for a particular combinatorial optimization problem, the Maximum Independent Set. This is an important optimization problem with applications in scheduling [22], inference of phylogenetic trees [23], communications [24], or portfolio optimization [25]. For the penalty term approach, we have highlighted the significance of choosing the penalty factor. We have also analyzed the quantum alternating operator ansatz and pointed out that an initial state that is a superposition of feasible states is preferable for the performance of the algorithm. The penalty term algorithm has the advantage that it uses a low-depth quantum circuit, but it has the disadvantage that it requires pruning that is dependent on the choice of the penalty factor. The quantum alternating operator ansatz, on the other hand, outputs only the feasible states, but it uses very large-depth circuits.

Finally, in this work we have also introduced the dynamic quantum variational ansatz (DQVA) that addresses the large quantum resource requirement of the quantum alternating operator ansatz. The DQVA algorithm has three main components that allow us to take advantage of the available classical and quantum resources. The first one is the jump start step that takes advantage of available classical approximate polynomial time algorithms for giving the algorithm a jump start. The second component is the dynamic update of the ansatz based on the classical outputs that the quantum approximate optimization generates. The third component is the randomization that takes advantage of the ordering freedom that the mixing unitary has in choosing the ordering of the partial mixing unitaries. Using these components, we can maximize the performance of the quantum approximate optimization algorithm for constrained optimization problems using limited quantum resources.

For future work we plan to study the approaches presented here on classical simulators as well as on actual quantum hardware. Of interest is how much we can scale up in terms of problem size using the dynamic quantum variational ansatz. We also want to understand the optimal choice of the penalty factor in the penalty term approach using simulations. The methods we have used in our analysis and the algorithm we have suggested in this work can be applied to other constrained combinatorial optimization problems such as the Max k-Colorability, Max k-Colorable Induced Subgraph, traveling salesman problem, and max set packing. We hope to extend our study to these problems as well.

ACKNOWLEDGMENTS

We thank Kaiwen Gui, Ruslan Shaydulin, and James Stokes for helpful discussions. This material is based upon work partly supported by the National Science Foundation under Award No. 2037984, and partly by Laboratory Directed Research and Development (LDRD) funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under contract DE-AC02-06CH11357.

[1] J. M. Gambetta, J. M. Chow, and M. Steffen, Building logical qubits in a superconducting quantum computing system, npj Quantum Information **3**, 1 (2017).

[2] C. Monroe and J. Kim, Scaling the ion trap quantum processor, Science **339**, 1164 (2013).

[3] M. Saffman, Quantum computing with neutral atoms, National Science Review **6**, 24 (2019).

[4] F. Dolde, I. Jakobi, B. Naydenov, N. Zhao, S. Pezzagna, C. Trautmann, J. Meijer, P. Neumann, F. Jelezko, and J. Wrachtrup, Room-temperature entanglement between single defect spins in diamond, Nature Physics **9**, 139 (2013).

[5] H. Bernien, B. Hensen, W. Pfaff, G. Koolstra, M. Blok, L. Robledo, T. Taminiau, M. Markham, D. Twitchen, L. Childress, *et al.*, Heralded entanglement between solid-state qubits separated by 3 meters, APS **2013**, T5 (2013).

[6] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[7] A. M. Dalzell, A. W. Harrow, D. E. Koh, and R. L. La Placa, How many qubits are needed for quantum computational supremacy?, Quantum **4**, 264 (2020).

[8] A. Lucas, Ising formulations of many NP problems, Frontiers in Physics **2**, 5 (2014).

[9] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv preprint arXiv:1411.4028 (2014).

[10] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem, arXiv preprint arXiv:1811.08419 (2018).

[11] E. Farhi and A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm, arXiv preprint arXiv:1602.07674 (2016).

[12] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, Quantum approximate optimization algorithm for MaxCut: A fermionic view, Physical Review A **97**, 022304 (2018).

[13] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, XY mixers: Analytical and numerical results for the quantum alternating operator ansatz, Physical Review A **101**, 012320 (2020).

[14] S. Hadfield, Quantum algorithms for scientific computing and approximate optimization, arXiv preprint arXiv:1805.03265 (2018).

[15] 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).

[16] Z. H. Saleem, Max-independent set and the quantum alternating operator ansatz, International Journal of Quantum Information **18**, 2050011 (2020).

[17] E. Farhi, D. Gamarnik, and S. Gutmann, The Quantum Approximate Optimization Algorithm needs to see the whole graph: Worst case examples, arXiv preprint arXiv:2005.08747 (2020).

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

[19] R. M. Karp, Reducibility among combinatorial problems, in *Complexity of computer computations* (Springer, 1972) pp. 85–103.

[20] C. Schön, K. Hammerer, M. M. Wolf, J. I. Cirac, and E. Solano, Sequential generation of matrix-product states in cavity QED, Physical Review A **75**, 032311 (2007).

[21] H. Wang, S. Ashhab, and F. Nori, Efficient quantum algorithm for preparing molecular-system-like states on a quantum computer, Physical Review A **79**, 042335 (2009).

[22] C. Ambühl and M. Mastrolilli, Single machine precedence constrained scheduling is a vertex cover problem, Algorithmica **53**, 488 (2009).

[23] A. Auyeung and A. Abraham, The largest compatible subset problem for phylogenetic data, arXiv preprint cs/0405025 (2004).

[24] M. Safar and S. Habib, Hard constrained vertex-cover communication algorithm for WSN, in *International Conference on Embedded and Ubiquitous Computing* (Springer, 2007) pp. 635–649.

[25] A. Kalra, F. Qureshi, and M. Tisi, Portfolio asset identification using graph algorithms on a quantum annealer, Available at SSRN 3333537 (2018).