Quantum Local Search with Quantum Alternating Operator Ansatz

Teague Tomesh,[1,*] Zain H. Saleem,[2,†] and Martin Suchara[2,‡]

[1] *Department of Computer Science, Princeton University, Princeton, NJ 08540, USA.*
[2] *Argonne National Laboratory, 9700 S. Cass Ave., Lemont, IL 60439, USA.*

(Dated: July 12, 2021)

We present a new hybrid, local search algorithm for quantum approximate optimization of constrained combinatorial optimization problems. We focus on the Maximum Independent Set problem and demonstrate the ability of quantum local search to solve large problem instances on quantum devices with few qubits. The quantum local search algorithm iteratively finds independent sets over carefully constructed neighborhoods and combines these solutions to obtain a global solution. We compare the performance of this algorithm on 3-regular graphs with up to 100 nodes against the well known classical Boppana-Halldórsson algorithm for the Maximum Independent Set problem.

I. INTRODUCTION

Quantum Approximate Optimization Algorithm (QAOA) [1] is a hybrid quantum-classical algorithm for finding the approximate solution to combinatorial optimization problems. This hybrid approach first encodes the problem's objective function as a Hamiltonian whose ground state corresponds to the optimal solution. Then the classical and quantum processors work together within a variational loop to find the ground state. The classical computer runs an optimization algorithm which traverses the optimization landscape searching for the extrema. During the course of the optimization the quantum processor is used to evaluate the expectation value of the objective function.

For unconstrained combinatorial optimization problems the optimization is performed over the entire Hilbert space generated by the variational ansatz (i.e. a parameterized quantum circuit). A new ansatz, the Quantum Alternating Operator Ansatz (QAO-Ansatz), was proposed in [2, 3] for solving *constrained* combinatorial optimization problems. This variational ansatz is designed in such a way that the constraints are satisfied at all times and the optimization is performed only over the space of feasible solutions.

Quantum local search (QLS) utilizes the QAO-Ansatz to find independent sets within local neighborhoods (i.e. subgraphs) whose size matches the capabilities of the quantum hardware. One of the main building blocks of the QAO-Ansatz is the mixing unitary which is defined only up to a permutation of its components (i.e. partial mixers). QLS exploits this permutation freedom to search for optimal solutions within a neighborhood. The QLS algorithm draws on methods from classical local search [4] which are useful for problems where computing the global solution is intractable, but are amenable to decomposition into tractable subproblems. QLS constructs a global solution by iterating through many local subproblems and involves a dynamical update of the variational ansatz such that a constant amount of quantum resources are utilized. Local search strategies relying on graph partitioning have been previously applied in the quantum context to unconstrained optimization problems such as network community detection [5], but the algorithmic components introduced in this work have not been previously studied.

Techniques based on quantum local search are advantageous to classical local search because they use entanglement to explore the solution landscape of the neighborhood all-at-once instead of one-by-one. In this work, we study the performance of the QLS algorithm on the Maximum Independent Set (MIS) problem [6]. MIS is one of the most widely studied constrained combinatorial optimization problems, in part, due to its broad applicability in a variety of domains and the fact that it is equivalent to other important problems such as minimum vertex cover and maximum clique on its complement graph. We introduce a method for constructing quantum circuits within a local neighborhood of a larger graph that are tunable to the size of the quantum hardware that is available. We simulate this algorithm on 3-regular graphs and find larger independent sets than Boppana-Halldórsson evaluated over the full graph. The tunability of QLS allows us to target graphs containing many more nodes than the number of qubits that are available in the hardware.

In Section II we review the quantum approximate optimization algorithm and cover prior strategies for quantum constrained optimization. We introduce the quantum local search algorithm, provide its pseudocode, and provide an open source implementation [7] in Section III. Section IV provides the simulation results which compare the performance of QLS and Boppana-Halldórsson, and study the impact of available quantum resources on runtime. Section V concludes and suggests future directions of this work.

* ttomesh@princeton.edu
† zsaleem@anl.gov
‡ msuchara@anl.gov

II. BACKGROUND

A. Quantum Approximate Optimization Algorithm

Hybrid variational algorithms, like QAOA [1], solve optimization problems by iteratively searching through the solution space with the combined efforts of a classical and quantum computer. The classical processor runs an optimization routine and calls the quantum processor to evaluate the computationally difficult objective function. For combinatorial optimization problems such as MIS, the problem is defined on a graph with n vertices, and 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 written as a quantum operator diagonal in its computational basis:

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

The expectation value of this objective function is measured with respect to the variational state,

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

where $|s\rangle$ is the state on which we act with unitary operators to build our variational ansatz. The ansatz in Eq. 2 is composed of two repeating parts: the phase separator unitary $e^{i\gamma_i C}$ and the mixing unitary $e^{i\beta_i M}$. The phase separator is a diagonal operator in the computational basis and typically takes the same form as the objective operator. The mixers are used to increase or decrease the amplitudes of different states – effectively "mixing" the state of the current wavefunction. The variational parameters γ and β define the optimization landscape and correspond to the rotation angles of quantum gates within the ansatz.

For any variational state, the expectation value of C_{obj}

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

is evaluated on a quantum computer and then passed to a classical optimizer which attempts 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 combinatorial optimization problem.

B. Constrained Optimization: Maximum Independent Set

When applying variational algorithms to unconstrained optimization problems, every basis state is a valid solution and therefore the optimization takes place over the entire Hilbert space generated by the variational ansatz. In contrast, constrained optimization is restricted to those basis states which satisfy the problem specific requirements (e.g. a valid independent set cannot contain two vertices which are neighbors). Hybrid variational algorithms have been adapted to constrained optimization problems in two main ways. Either the objective function is modified to heavily penalize invalid basis states, effectively turning the constrained problem into an unconstrained one [8, 9], or the variational ansatz is structured in a way that keeps the optimization within the valid subspace [2, 3, 10].

Maximum Independent Set (MIS) is an NP-Hard constrained combinatorial optimization problem defined on the graph $G = (V, E)$ with nodes V, edges E and number of nodes $n = |V|$ [6]. An independent set is defined as a subset $V' \subset V$ of the graph's nodes such that no two vertices in V' share an edge. The goal of MIS is to find the independent set containing the largest number of nodes.

The Quantum Alternating Operator Ansatz (QAO-Ansatz) [2, 3] is an example of an ansatz which imposes constraints at the quantum circuit level. The ansatz is constructed in such a way that we never leave the set of feasible states during the variational optimization (e.g. the set of all valid independent sets). For the MIS problem the objective function is the Hamming weight operator,

$$C_{obj} = H = \sum_{i\in V} b_i \tag{4}$$

where $b_i = \frac{1}{2}(1 - Z_i)$ and Z_i is the Pauli-Z operator acting on the i-th qubit. Each vertex in the graph is assigned a value $b_i \in \{0,1\}$ indicating whether it is excluded (0) or included (1) in the independent set.

The initial state $|s\rangle$ for the variational optimization can be any feasible state or superposition of feasible states. Similar to regular QAOA, the phase separator unitary for the QAO-Ansatz, $U_C(\gamma) \coloneqq e^{i\gamma H}$, is constructed using the objective function. However, the mixing unitary $U_M(\beta)$ is non-trivial and requires multi-qubit gates for its execution.

$$U_M(\beta) \coloneqq \prod_i e^{i\beta M_i} \qquad M_i = X_i \bar{B} \tag{5}$$

and we have defined,

$$\bar{B} \coloneqq \prod_{j=1}^{\ell} \bar{b}_{v_j}, \quad \bar{b}_{v_j} = \frac{1 + Z_{v_j}}{2}, \tag{6}$$

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

$$U_M(\beta) = \prod_{i=1}^{n} V_i(\beta) = \prod_{i=1}^{n} \left(I + (e^{-i\beta X_i} - I)\,\bar{B}\right), \tag{7}$$

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 in Eq. 7 can be implemented using multi-controlled Toffoli gates and controlled-X rotations. The

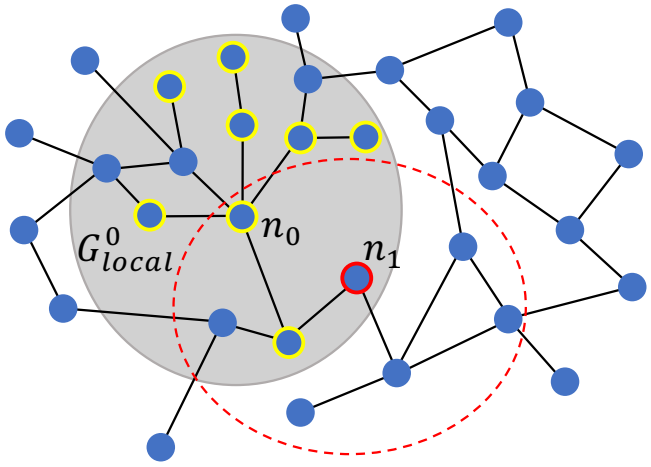

FIG. 1: QLS example: the neighborhood (grey circle) G^0_{local} of distance $N_s = 2$ surrounds the root node n_0. The yellow highlighted nodes indicate where the partial mixers may be applied since all of their neighbors are contained within G^0_{local}. After the variational optimization is complete, a new root node, n_1 is selected and will induce a new subgraph (red dotted circle). The size of the neighborhood N_s can be scaled to match the problem instance or the available quantum resources.

effect of applying a partial mixer $V_i(\beta)$ can be stated in words as: if all of node i's neighbors are in the $|0\rangle$ state (i.e. are not included in the current independent set), then rotate qubit i's state around the X-axis by an angle β.

III. QUANTUM LOCAL SEARCH

The Quantum Local Search (QLS) algorithm finds approximate solutions to the MIS problem on a graph $G = (V, E)$ with n vertices by iteratively optimizing a variational ansatz over small neighborhoods within G. We give an outline of the QLS pseudocode in Alg. 1 and an implementation is available via Github [7].

(1) Neighborhood Initialization: Select a root node n_0 and its corresponding local subgraph G^0_{local} where all the nodes in this subgraph are a node distance N_s away from n_0 (see Fig. 1 for an example). The distance N_s is a free parameter used to set the size of the neighborhood. This parameter should be set according to the density of the target graph such that the number of nodes m in the neighborhood does not exceed the number of qubits available in the quantum hardware. In this work, we set the initial state $|s\rangle$ for the neighborhood to be the all zero state $|s\rangle = |00\cdots0\rangle$, but more interesting states could also be used [11, 12].

(2) Neighborhood Ansatz Construction: For simplicity, the example ansatz construction covered here will be restricted to the case with depth parameter $p = 1$, but this method can be readily extended to $p > 1$. For depth $p = 1$ the variational ansatz takes the form

$$|\psi(\vec{\beta},\gamma)\rangle = U_{qls}\,|s\rangle = U_M(\vec{\beta})U_C(\gamma)\,|s\rangle\,. \tag{8}$$

The phase separator $U_C(\gamma) = e^{i\gamma H}$ is identical to that used in the QAO-Ansatz. The mixer unitary is parameterized by a set of m angles $\vec{\beta} = (\beta_1, \beta_2, ..., \beta_m)$ corresponding to the partial mixers of each node in the subgraph.

$$U_M(\vec{\beta}) = \prod_{i=1}^{m} V_i(\beta_i) = \prod_{i=1}^{m} \left(I + (e^{-i\beta_i X_i} - I)\,\bar{B}\right), \tag{9}$$

Applying all m partial mixers at once may require an intractable amount of quantum resources, especially for larger neighborhoods. Instead, we use a hyperparameter k which sets the number of nonzero β_i within $\vec{\beta}$. With k set, we start by applying the partial mixers first to the central node n_0, then to the nodes that are distance one away from the central node and then distance two and so on until we have either exhausted the quota of the k partial mixers or the nodes in the neighborhood.

Algorithm 1: Quantum Local Search (QLS)

Input : $G = (V, E)$, r = # of permutation rounds, k = # of partial mixers, N_s = neighborhood size
Output: Approximate MIS of G

local_solutions ← [];
repeat
  `/* Neighborhood Initialization */`
  s ← initial state;
  n_{root} ← select a root node;
  G_{local} ← subgraph of n_{root} up to distance N_s;
  `/* Neighborhood Ansatz Construction */`
  $\vec{\beta} \leftarrow [0_0, 0_1, \ldots, 0_m]$;
  $\gamma \leftarrow \text{random}(0, 2\pi)$;
  `/* Randomly initialize` k `partial mixers starting from` n_{root} `*/`
  for $i \in [k]$ **do**
    $\beta_i \leftarrow \text{random}(0, 2\pi)$;
  end
  $U_{qls} \leftarrow U_M(\vec{\beta})U_C(\gamma)$;
  `/* Neighborhood Solution Search */`
  bitstrs ← [];
  for $i \in r$ **do**
    $U_M(\vec{\beta}) \leftarrow \mathcal{P}\big(V_1(\beta_1)V_2(\beta_2)\cdots V_m(\beta_m)\big)$;
    while *not converged* **do**
      counts ← execute($U_{qls}(\vec{\beta},\gamma)\,|s\rangle$);
      $E \leftarrow$ expectation_value(H, counts);
      $\vec{\beta}, \gamma \leftarrow$ updated_params(E);
    end
    bitstrs.append($\arg\max_b$ ([$H(b)$ for $b \in$ counts]);
  end
  local_solutions.append
    ← $\arg\max_b$ ([$H(b)$ for $b \in$ bitstrs]);
until *all nodes have been visited*;
return *recombine(local_solutions)*

In the process of constructing U_{qls} it is possible that we could find a high degree node within the neighborhood which has more neighbors than there are qubits available in the quantum hardware (or equivalently, exceeds the amount of allocated resources). We can handle this corner case by simply skipping the application of this node's partial mixer. This node can still participate in its neighbor's partial mixers as a control qubit.

(3) Neighborhood Solution Search: Once the circuit construction is finished, we run the quantum approximate optimization algorithm with the goal of finding the parameters $\vec{\beta}$ and γ in Eq. 8 that maximize

$$\left\langle \psi(\vec{\beta},\gamma) \right| H \left| \psi(\vec{\beta},\gamma) \right\rangle . \tag{10}$$

This will output a bitstring with a certain Hamming weight. Since the mixer unitary is defined up to a permutation of the partial mixers:

$$U_M(\vec{\beta}) \simeq \mathcal{P}\big(V_1(\beta_1)V_2(\beta_2)\cdots V_m(\beta_m)\big), \tag{11}$$

different permutations can return bitstrings with different Hamming weights. We can rerun this step r number of times, each time randomly choosing a different permutation of the partial mixers. From these r different rounds we will select the bitstring with the largest Hamming weight.

(4) Neighborhood Update: Once we have obtained an independent set on the neighborhood G^0_{local} we traverse the graph G by selecting a new root node. The root node n_1, red highlighted node in Fig. 1, of the next neighborhood G^1_{local}, red dashed circle in Fig. 1, is randomly selected from the set of vertices that are on the edge of the current neighborhood, a distance N_s away from n_0.

(5) Obtaining Approximate MIS on G: We then repeat steps 1 through 4 starting with the new root node, and continue this process until all nodes have participated in the local search. Once the graph has been traversed we recombine the solutions on all the G^i_{local} into a global approximate solution over the full graph G.

IV. SIMULATION RESULTS

To test the performance of QLS we implemented Algorithm 1 in Qiskit [13] and simulated its execution on 3-regular graphs containing 20, 60, and 100 vertices. We selected 3-regular graphs for our experiment because it is well known that MIS is NP-complete on 3-regular graphs and restricting the degree of the graph allows us to easily compare the performance of the algorithm for different choices of the numbers of partial mixers used in each neighborhood. Applying the algorithm on graphs with high degree might result in exhaustion of the quantum resources without being able to see quantum interference within each of the neighborhood since each of the multi-qubit gate will use up a lot of control qubits. For each graph size, we generated 20 random instances using the NetworkX Python package [14] and collected the results of 10 separate executions of the QLS algorithm. To keep the simulations tractable on our local machines on which the experiments were conducted we set the size of the neighborhood to $N_s = 2$ for all runs.

The typical measure of performance for MIS is the approximation ratio between the independent set found by the algorithm and the optimal MIS. Since finding the optimal MIS for large graphs quickly becomes intractable, we instead measure the performance of QLS against a well-known recursive algorithm from Boppana and Halldórsson [15]. In the results below we report the Boppana-Halldórsson approximation ratio, R_{BH}, which is given by the expression

$$R_{BH} = \frac{E_{QLS}}{E_{BH}}, \tag{12}$$

where E_{QLS} and E_{BH} are the sizes of the final maximum independent sets output by QLS and Boppana-Halldórsson.

Fig. 2 shows the performance of QLS, with different values of the hyperparameter k, on graphs with 20, 60, and 100 vertices. The hyperparameter k controls the maximum number of partial mixers that are allowed within the ansatz during each iteration of the algorithm. From Fig. 2 we see that the utilization of more quantum resources allows QLS to add more vertices to the independent set during the early iterations of the algorithm. As the execution progresses the number of nodes added in each round begins to decline until all nodes have been visited and the program halts. Interestingly, we also see that increasing k (i.e. adding more quantum resources) beyond a certain threshold value no longer improves the growth rate of the independent set. This is shown by the $k = 4, 10$ lines in Fig. 2 which lie on top of one another and in Fig. 3 which shows a constant approximation ratio is reached beyond $k \approx 4$.

The reason why we see constant performance beyond $k = 4$ is due to the regularity of the target graphs and our selection of $N_s = 2$. Within the neighborhood ansatz construction step of QLS, every circuit will contain approximately 10 qubits and on average only 4 of those qubits will have all of their neighbors present within the subgraph and thus can be hit with a partial mixer. Therefore, we do not see any significant advantage of the $k > 4$ over the $k = 4$ runs. This observation highlights the tunability of the QLS algorithm to specific problem instances and the potential for more sophisticated protocols within the neighborhood construction function. Modifying the values of k and N_s allows the QLS algorithm to tradeoff between quantum and classical resources: larger N_s, k means more of the graph can be explored at once using wider quantum circuits but requires optimizing over more parameters, smaller N_s, k results in quantum circuits with fewer qubits but requires more

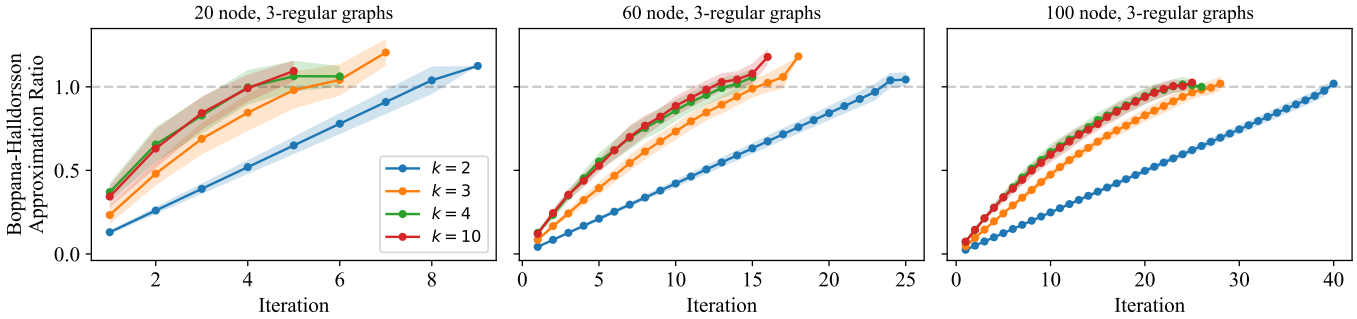

FIG. 2: Average Boppana-Halldórsson approximation ratio (Eq. 12) of the MIS found by QLS at each iteration of the algorithm. Shaded areas indicate one standard deviation from the mean. QLS converges to a solution faster as the number of partial mixers is increased up until a threshold value is crossed.

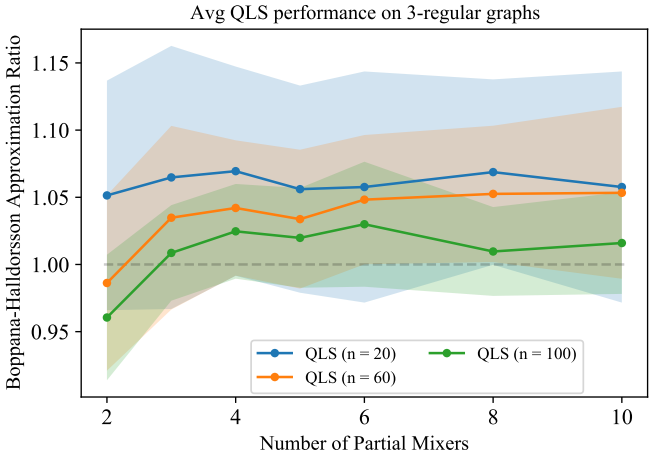

FIG. 3: Average approximation ratio over random 3-regular graphs as a function of increasing k. All benchmarks use the all-zero, $|0\rangle$, initial state.

iterations to fully explore the graph.

Finally, in Fig. 3 the approximation ratios trend towards 1 as the graph size increases. This seems to suggest that QLS is able to find a constant improvement over the MIS's found by Boppana-Halldórsson. However, the QLS implementation in Algorithm 1 is the most straightforward specification for local search. In practice, many local search algorithms decide on the next vertex to visit using much more sophisticated methods that incorporate current problem information.

V. CONCLUSION AND FUTURE DIRECTIONS

While this work has focused on MIS, the QLS approach with the QAO-Ansatz can be applied to many more constrained combinatorial optimization problems. The non-commutativity of the components of the mixing unitary within the QAO-Ansatz can be exploited by QLS for many of these problems. Additionally, while we have restricted our simulations to 100 node graphs, this technique is easily scalable to much larger problem sizes.

In this work, we evaluated the performance of QLS against the Boppana-Halldórsson algorithm which is an efficient heuristic algorithm, but does not promise the best solutions that are obtainable via other available classical techniques. We demonstrated the applicability of QLS to large problem sizes, but there are still many improvements that can be made to the neighborhood solution search and neighborhood update steps by drawing upon the extensive work that has been done in the classical community on local search[4, 16–18]. One advantage that QLS maintains over classical approaches to local search is its ability to exploit quantum entanglement to search the solution landscape within a neighborhood all-at-once instead of one-by-one.

The multi-qubit gates required to implement this algorithm can be quite expensive when decomposed into the single- and two-qubit gates required by current superconducting and trapped-ion architectures [19, 20]. However, emerging quantum computer architectures, such as neutral atoms [21], are especially promising because they support the ability to natively implement these multi-qubit operations. It will be interesting to study the implementation of QLS on current and future quantum computing systems.

ACKNOWLEDGMENTS

T.T. is supported in part by EPiQC, an NSF Expedition in Computing, under grant CCF-1730082. Z.S. and M.S. are supported by the National Science Foundation under Award No. 2037984, and by the US Department of Energy (DOE) Office of Science Advanced Scientific Computing Research (ASCR) Accelerated Research in Quantum Computing (ARQC).

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

[2] S. A. Hadfield, *Quantum algorithms for scientific computing and approximate optimization* (Columbia University, 2018).

[3] 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).

[4] E. Aarts, E. H. Aarts, and J. K. Lenstra, *Local search in combinatorial optimization* (Princeton University Press, 2003).

[5] R. Shaydulin, H. Ushijima-Mwesigwa, I. Safro, S. Mniszewski, and Y. Alexeev, Network community detection on small quantum computers, Advanced Quantum Technologies **2**, 1900029 (2019).

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

[7] Quantum-Software-Tools, quantum-local-search, `https://github.com/Quantum-Software-Tools/quantum-local-search` (2021).

[8] 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).

[9] 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).

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

[11] D. J. Egger, J. Mareček, and S. Woerner, Warm-starting quantum optimization, Quantum **5**, 479 (2021).

[12] Z. H. Saleem, T. Tomesh, B. Tariq, and M. Suchara, Approaches to constrained quantum approximate optimization (2021), arXiv:2010.06660 [quant-ph].

[13] A. Cross, The ibm q experience and qiskit open-source quantum computing software, in *APS March Meeting Abstracts*, Vol. 2018 (2018) pp. L58–003.

[14] A. Hagberg, P. Swart, and D. S Chult, *Exploring network structure, dynamics, and function using NetworkX*, Tech. Rep. (Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008).

[15] R. Boppana and M. M. Halldórsson, Approximating maximum independent sets by excluding subgraphs, BIT Numerical Mathematics **32**, 180 (1992).

[16] D. V. Andrade, M. G. Resende, and R. F. Werneck, Fast local search for the maximum independent set problem, Journal of Heuristics **18**, 525 (2012).

[17] M. Krivelevich, T. Mészáros, P. Michaeli, and C. Shikhelman, Greedy maximal independent sets via local limits, arXiv preprint arXiv:1907.07216 (2019).

[18] Q. Wu and J.-K. Hao, A review on algorithms for maximum clique problems, European Journal of Operational Research **242**, 693 (2015).

[19] V. V. Shende and I. L. Markov, On the cnot-cost of toffoli gates, Quantum Info. Comput. **9**, 461–486 (2009).

[20] Y. He, M.-X. Luo, E. Zhang, H.-K. Wang, and X.-F. Wang, Decompositions of n-qubit toffoli gates with linear circuit complexity, International Journal of Theoretical Physics **56**, 2350 (2017).

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