Divide and Conquer-based Quantum Algorithms for Maximum Independent Set on Large Separable Graphs

Hanjing Xu
Department of Computer Science
Purdue University
West Lafayette, Indiana USA
xu675@purdue.edu

Alex Pothen
Department of Computer Science
Purdue University
West Lafayette, Indiana USA
apothen@purdue.edu

2024 IEEE International Conference on Quantum Computing and Engineering (QCE) | 979-8-3315-4137-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/QCE60285.2024.00020

***Abstract*—Noisy Intermediate-Scale Quantum (NISQ) computers currently available have a few thousand qubits, and could potentially solve combinatorial optimization problems efficiently. However, the sizes of the problems that could be solved are limited by the number of qubits, their connectivity, high noise, and short coherence times. In this work, we propose hybrid quantum-classical algorithms based on the divide and conquer paradigm for solving larger Maximum Independent Set (MIS) and Maximum Weighted Independent Set (MWIS) problems on graphs than would otherwise be possible on NISQ devices. The machines include D-Wave Quantum Annealers and QuEra Quantum Computers with Neutral Atoms. Our algorithms are designed for separable graphs, which are classes of graphs with good vertex separators; these include planar graphs, finite element meshes with good aspect ratios, nearest neighbor graphs, and certain classes of geometrically defined graphs. The algorithms recursively employ vertex separators to create a collection of small subgraphs, from which NISQ computers obtain disjoint maximal independent sets, which are then augmented by independent vertices from the separators. We demonstrate competitive results compared to the classical Luby's algorithm, and KaMIS, a state-of-the-art classical MIS solver, on graphs with several thousand vertices. These divide and conquer-based algorithms are also well-suited for distributed quantum computer architectures.**

***Index Terms*—Maximum Independent Set, Divide and Conquer, Quantum Annealing, Quantum Computing with Neutral Atoms, Vertex Separators**

I. Introduction

We describe divide-and-conquer (DC)-based quantum algorithms to compute heuristic solutions of the Maximum Independent Set (MIS) and Maximum Weighted Independent Set (MWIS) problems on graphs with several thousands of vertices. Since problems of these sizes do not fit on current NISQ computers, we employ the DC framework to partition the graphs (recursively) into roughly equal-sized components of small enough sizes by finding vertex separators, and then solve each subproblem on quantum computers. This approach is effective for *separable* graphs, which are graphs in which a small set of vertices could be removed to obtain two disconnected subgraphs with roughly equal numbers of vertices.

Current NISQ quantum computer systems are limited by low qubit counts and high error rates, and are thus unable to solve large optimization problems whose sizes are beyond the capability of classical computers. Scaling the size of quantum computers is a challenge because qubit coherence times, gate fidelities, and processor yield rates deteriorate with increasing qubit counts. A distributed quantum computer architecture that brings together several smaller quantum processors represents an alternative approach to solving larger problems. However, performing remote qubit operations across quantum processors is more demanding in terms of time and fidelity than doing such operations within a quantum processor. The DC-based algorithms described here are also eminently suited for distributed quantum computer architectures.

We report results from two types of quantum computers for the MIS problem: D-Wave Quantum Annealers and QuEra Quantum Computer with Neutral Atoms. Quantum Annealers implement annealing algorithms to solve optimization problems by using quantum tunneling from a manifold of high-energy states to ground or near-ground states. D-Wave Quantum Annealers specifically minimize the energy of Ising Models and equivalently, solve Quadratic Unconstrained Binary Optimization (QUBO) problems. On the other hand, QuEra Quantum Computers with Neutral Atoms use the Rydberg blockade effect to solve MIS Problems on unit disk graphs. For gate-based quantum computers, Quantum Approximate Optimization Algorithm [1], Quantum Imaginary Time Evolution [2], Quantum Random Access Optimization [3] and many more [4] have been proposed, but their implementations are hamstrung by the limited size, connectivity and coherence time of the devices available today.

Given a graph $G = (V, E)$, an *independent set* in G is a subset of its vertices where no two vertices in the set are joined by an edge. The MIS problem is to find an independent set with the maximum cardinality, and it is NP-hard [5] to solve exactly or even approximately. When there are weights on the vertices, the MWIS problem is to find an independent set of vertices with the maximum sum of weights, and it is also NP-hard. Hence all algorithms discussed in this paper find

979-8-3315-4137-8/24/$31.00 ©2024 IEEE
DOI 10.1109/QCE60285.2024.00020

heuristic solutions to the MIS and MWIS problems.

The MIS problem is closely related to the Minimum Vertex Cover (MVC) and Maximum Clique (MC) problems. A vertex set I is an MIS of G if and only if it is a maximum clique in the complement graph $\bar{G}$. Furthermore, a minimum vertex cover of G can be obtained from the set of vertices $V \setminus I$. These three problems have applications in diverse areas including constructing virtual backbones from *ad hoc* wireless networks [6], covering problems via clique partition [7], matching molecular structures using clique detection [8], macromolecular docking [9], and analyzing genome mapping data [10]. Many exact and heuristic algorithms have been proposed to solve MIS problems: clique enumeration [11], [12], simulated annealing [13], [14], genetic algorithms [15] and more.

Due to the infeasibility of solving MIS problems, we turn to solvers that find a *maximal independent set*, a set of independent vertices that is contained in no larger independent set. Luby's algorithm [16] is a randomized parallel algorithm for finding such an independent set in polylogarithmic time. However, it is not designed to maximize the cardinality of the independent set. The Karlsruhe Maximum Independent Sets (KaMIS) [17] package is a state-of-the-art software library for solving unweighted and weighted MIS instances on classical computers, and it will be described in Section III. Apart from finding a maximal independent set, KaMIS also attempts to maximize the set cardinalities and weights in a heuristic manner. We will compare cardinalities or weights computed by our algorithms with those obtained by Luby's algorithm and KaMIS.

The paper is structured as follows. Section II includes the technical background on D-Wave Quantum Annealers and QuEra Quantum Computers with Neutral Atoms. In Section III we introduce the QUBO model for MIS problems, describe our divide and conquer MIS and MWIS algorithms based on vertex separators, and discuss earlier quantum algorithms for MIS and other combinatorial optimization problems. In Section IV, we describe two classical MIS algorithms: KaMIS represents the state of the art among practical algorithms, while Luby's algorithm represents a significant theoretical contribution. In Section V we compare our algorithms to Luby's algorithm and KaMIS using both D-Wave and QuEra quantum computers, and in Section VI we conclude with future research directions.

II. Background

A. Separable Graphs and Unit Disk Graphs

A *vertex separator* in a graph $G = (V, E)$ is a subset of vertices S whose removal from G leads to two subgraphs such that no edge in G joins a vertex in the first subgraph to a vertex in the second subgraph. k-way separators are a generalization of separators to a higher number of subgraphs. For simplicity of implementations, we fix $k = 2$ for the remainder of the paper. We look for *balanced separators*, such that the larger subgraph has no more than αn vertices, where n is the number of vertices in G, and α is a constant such that $1/2 \leq \alpha < 1$. A subset of vertices S is a separator if and only if any path between a vertex in the first subgraph and a vertex in the second subgraph must include at least one vertex from S. A class of graphs satisfies an n^c-separator property for $c < 1$, if every graph G with n vertices from the class has a balanced separator S with at most $O(n^c)$ vertices, and each resulting subgraph also satisfies such a property with respect to the number of its vertices.

Planar graphs are graphs that can be drawn in the plane such that their edges do not cross, and they have $n^{1/2}$-separators. Many graphs embeddable in three dimensions (such as finite element meshes with good aspect ratios) have $n^{2/3}$-separators [18]. It is NP-Hard to find MIS on planar graphs with maximum vertex degree of 3 [19] or of large girth [20], although polynomial-time approximation algorithms exist for any approximation ratio $c < 1$ [21]. Nearest neighbor graphs embeddable in three dimensions also have small separators, and many graphs with small separators can be characterized geometrically [22]. Although Erdös-Rényi random graphs do not have small vertex separators, many graphs that occur in scientific and engineering domains do. We consider graphs with small separators in this paper, and will refer to them as *separable* graphs.

We could also consider edge separators, which would be a subset of edges such that its removal from the graph would disconnect the graph into two or more connected subgraphs. Small edge separators would be useful in solving other graph problems by the divide and conquer approach described here, but we leave that for future work.

QuEra quantum computers naturally compute maximum independent sets in unit disk graphs, and hence we discuss them now. The vertices of a *unit disk graph* can be put in one-to-one correspondence with a set of circles that have equal radii, such that two vertices are joined by an edge if the circles intersect or are tangent to each other. Unit disk graphs are not planar, nor do they belong to the class of perfect graphs. Computing the disk representation of a unit disk graph is NP-hard. Unit disk graphs possess separators with $O(\sqrt{m + n} \log n)$ vertices (where m is the number of edges in G), and the separator vertices can be chosen to lie along a line [23]. Solving MIS problems exactly on UDGs is NP-hard [24]. Marathe et al. [25] have designed 3-approximation algorithms for such problems. However, unless the disk representation of these graphs is given, the algorithm has a high time complexity of $O(n^4)$, and is not practical for graphs with thousands of vertices.

B. D-Wave Quantum Annealer

Adiabatic Quantum Computation (AQC) [26], [27] is a quantum computing model that can be used to solve optimization problems. It relies on the Adiabatic Theorem [28] to find the ground states of problem Hamiltonians, which is often beyond the capabilities of today's quantum computers. Quantum Annealing (QA) [29]–[31], however, serves as an intermediate step towards AQC. It is a heuristic algorithm for finding approximate solutions to the following Ising problem:

$$\min_{s} \ E(s) = \sum_{i} h_i s_i + \sum_{i,j} J_{ij} s_i s_j, \quad s_i \in \{-1, 1\}. \quad (1)$$

QA can be considered as a quantum analog of the classical thermal/simulated annealing [13], where the disorder is introduced quantum mechanically instead of thermally. The D-Wave Quantum Annealer is a physical implementation of the QA algorithm and the latest Advantage System 6.4 features more than 5600 qubits. However, qubits in the annealers are connected sparsely using the Pegasus topology [32] with a maximum degree of 15. Thus, it requires general Ising Models to be minor-embedded on the Quantum Annealer [33], [34]. In this work, we use the software *minorminer* [35] for the embedding task.

By mapping x_i to $\frac{1+s_i}{2}$ in Eq 1, one can obtain the formulation for the Quadratic Unconstrained Binary Optimization(QUBO) problem:

$$\min_{x} \ Q(x) = \sum_i h_i x_i + \sum_{i,j} J_{ij} x_i x_j, \quad x_i \in \{0, 1\}. \tag{2}$$

The general QUBO minimization problems are NP-hard [36]. Many combinatorial optimization problems have been formulated as QUBOs [37]–[40], including the MIS problem, which we will describe in Section III-A.

C. QuEra Quantum Computer with Neutral Atoms

QuEra Quantum Computers place atoms on a 2D plane using optical tweezers [41], [42], each of which is initialized to the ground state $|0\rangle$. The system is then evolved under the Hamiltonian [43]

$$H = \sum_v (\Omega_v \sigma_v^x - \Delta_v n_v) + \sum_{v<w} V(|\bar{x}_v - \bar{x}_w|) n_v n_w, \tag{3}$$

where Ω_v and Δ_v are the Rabi frequency and laser detuning at atom v. Also, $n_v = |1\rangle_v \langle 1|$ is the number operator at site v, and $V(x) = C/x^6$ with C being the constant interaction strength. σ_v^x is set to $|0\rangle \langle 1| + |1\rangle \langle 0|$ which induces quantum tunneling between spin configurations and $\bar{x}_v$ encodes the position of atom x. At the end of the simulation, atoms will try to enter excited state $|1\rangle$ with the constraint that two atoms within the Rydberg blockade radius cannot be excited at the same time. This is analogous to solving the MIS problem on unit disk graphs. MIS problems on arbitrary graphs can be transformed to equivalent unit disk graph instances with at most quadratic overhead in the number of qubits [44]. However, since QuEra's Aquila device currently supports only up to 256 atoms, we restrict ourselves to native unit disk graphs in the experiments.

III. Methods

A. QUBO for the Maximum Independent Set Problem

Recall that in the MWIS problem, given a graph $G = (V, E, w)$, where $w \in \mathbb{R}_+^n$ is the vector of vertex weights, we want to find an independent set I which has the maximum sum of weights $\sum_{u \in I} w_u$. The MWIS problem can be formulated as an integer program:

$$\begin{aligned} \max_{w} \quad & \sum_{i=1}^{n} w_i x_i \\ \text{s.t.} \quad & x_i x_j = 0, \ \forall (i,j) \in E \\ & x_i \in \{0,1\}, \ i = 1, \ldots, n. \end{aligned} \tag{4}$$

We can convert Eq 4 to the QUBO formulation in Eq 2 as follows:

$$Q(x_1, \ldots, x_n) = -\sum_{i \in V} w_i x_i + \sum_{(i,j) \in E} J_{ij} x_i x_j, \tag{5}$$

where $(x_1, \ldots, x_n) \in \{0,1\}^n$. The variable $x_i = 1$ if and only if vertex i is included in the independent set. It has been shown that if $J_{ij} > \min\{w_i, w_j\}$, for all $(i,j) \in E$, then the independent set corresponding to $\underset{(x_1,\ldots,x_n)}{\operatorname{argmin}} Q(x_1, \ldots, x_n)$ has maximum weight [45]. Eq 5 can then be rewritten as:

$$Q(x_1, \ldots, x_n) = -\sum_{i \in V} w_i x_i + p \sum_{(i,j) \in E} \min\{w_i, w_j\} x_i x_j, \tag{6}$$

with p as the tunable penalty coefficient. Although $p > 1$ guarantees the correctness of the QUBO formulation, its value has a significant effect on the solution qualities on quantum computers, as we will discuss in Section V-B2.

B. Divide-and-Conquer Algorithm

Algorithm 1 provides the pseudo-code for the algorithm that solves MWIS problems on D-Wave and QuEra quantum computers. We treat unweighted instances as a special case of MWIS where each vertex has a weight of one. The input graph is recursively partitioned via vertex separators until the number of vertices is smaller than some pre-defined constant, C. Each subproblem is then solved via QUBO reformulation on D-Wave or natively on QuEra. To reduce the number of accesses to quantum computers, when the problem size is too small (fewer than 15 vertices), we solve the MIS problem using exact, classical algorithms. For subproblems solved on quantum computers, we request N solution samples each time. Instead of selecting only the sample with lowest energy (D-Wave) or solution cardinality (QuEra), we collect the best $\alpha\%$ of all samples, repair all of them by removing adjacent vertices such that the remaining set is independent, and then greedily improve each sample. Finally, we pick the solution with the highest weight. This process has shown to greatly improve the MIS solution qualities with linear-time computational overhead.

Assuming we can find balanced separators for graph $G = (V, E)$ at each level and use a constant number of samples for post-processing, the recursion tree will have a height of $\log n$. At each level, each vertex will be processed once therefore the total overhead is $O(n \log n)$.

Algorithm 1 MIS Solver using Quantum Computers

```
function SOLVE_MIS($G = \{V, E\}$, $N$, $\alpha$, $C$)
    // $N$ dictates the total number of samples from quantum computers and C is the cutoff for subproblems
    if $n \leq 15$ then
        $I \leftarrow$ Exact_Maximum_Independent_Set($G$)
    else if $n \leq C$ then
        $I \leftarrow$ Solve_MIS_Quantum($G$, $N$, $\alpha$)
    else
        $A, B, S \leftarrow$ Find_Graph_Separator($G$)
        $I_A \leftarrow$ Solve_MIS($A$, $N$, $\alpha$, $C$)
        $I_B \leftarrow$ Solve_MIS($B$, $N$, $\alpha$, $C$)
        for $(v, u)$ in $E$, $v \in S$ and $u \in I_A \cup I_B$ do
            $S = S \setminus \{v\}$
        $I_S \leftarrow$ Solve_MIS($S$, $N$, $\alpha$, $C$)
        $I \leftarrow I_A \cup I_B \cup I_S$
        $I \leftarrow$ MIS_Greedy_Improvement($I$, $G$)
    return $I$
function SOLVE_MIS_QUANTUM($G = \{V, E\}$, $N$, $\alpha$)
    Reformulate MIS on $G$ to a QUBO $Q$ by Eq 6
    $I_{\text{total}} \leftarrow N$ samples from solving $Q$ on quantum computers
    $I_{\text{best}} \leftarrow \emptyset$
    for $I$ in lowest $\alpha\%$ of $I_{\text{total}}$ in terms of energy do
        $I \leftarrow$ Repair_Quantum_Solution($I$, $G$)
        $I \leftarrow$ MIS_Greedy_Improvement($I$, $G$)
        if weight($I$) $\geq$ weight($I_{\text{best}}$) then
            $I_{\text{best}} = I$
    return $I_{\text{best}}$
function REPAIR_QUANTUM_SOLUTION($I$, $G = (V, E)$)
    if $I$ is an independent set of $G$ then
        return $I$
    else
        while $I$ is not an independent set of $G$ do
            Remove $v$ from $I$ where $v = \underset{v \in I}{argmax}$ deg(v)
        return $I$
function MIS_GREEDY_IMPROVEMENT($I$, $G = (V, E)$)
    Sort $V$ in descending order of weights
    for $v$ in $V$ do
        if $(v, u) \notin E, \forall u \in I$ then
            $I = I \cup \{v\}$
    return $I$
```

C. Previous Quantum Solvers for MIS

Now we turn to previous solutions of MIS problems on quantum computers.

The DC paradigm works well for solving combinatorial optimization problems on certain classes of graphs called *separable graphs*. See the discussion in Sec. II-A for more details.

Tomesh et al. [46] have proposed a DC approach for solving combinatorial optimization problems on distributed quantum architectures. They use a QAOA formulation in which in each iteration they solve a variational optimization subproblem on each processor. The subproblems are obtained by partitioning the circuit on a gate-model quantum computer. This approach involves an exponential number of operations on the cut qubits on different processors on which the constraints of the combinatorial optimization problem are imposed. They have described such an approach for the MIS problem in which they employ edge separators to divide the graph into smaller subgraphs.

The divide and conquer approach we have proposed for MIS here has similarities as well as significant differences with the QAOA approach of Tomesh et al. Both approaches employ the divide and conquer paradigm, but we use the more natural vertex separators for MIS rather than edge separators employed by Tomesh et al. Our approach is capable of computing maximal (rather than maximum) independent sets by computing an independent set in each subgraph, and then augmenting these with additional vertices from the separators. The advantage of this solution is that the communication operation among the quantum processors is simple. However, it may be extended to do more involved local exchange operations to potentially increase the size of the maximal independent set.

We solve larger MIS instances on current quantum computers than earlier authors, including Tomesh et al. Here the subproblems are solved using a quantum processor, and the partial solutions are augmented with additional vertices from the small-sized vertex separators using a classical algorithm. We have implemented our algorithm on a quantum annealer and a neutral atom quantum processor. We compare our algorithm against one of the state-of-the-art classical MIS solvers, which we will discuss in the next section, and show that our hybrid quantum-classical algorithm finds independent sets of comparable sizes.

D. Divide and Conquer for Other Combinatorial Problems on Quantum Computers

Now we discuss how divide and conquer approaches have been used to solve other combinatorial problems on quantum computers. For MaxCut problems, [47] solves them by decomposing graphs using edge partitions and obtains near-optimal solutions on 29-vertex graphs using QAOA over exact simulations. Similarly, [48] first partitions the graph into disjoint components that have vertices no larger than available qubits and solves MaxCut using QAOA for each subgraph. Each subgraph is then contracted into a single vertex with edges between them indicating the number of cuts. The merging step can then be formulated as a new MaxCut problem where vertices in each subgraph are allowed to change signs together in order to improve the global solutions while preserving the local solution objective. For 2000-vertex regular and Erdös-Rényi graphs, the algorithm is able to generate solutions over 85% of those from a semi-definite programming solver and bounds in [49]. [50] separates the graph into three disjoint components V_1, V_2, K where K are computed with max-flow algorithms. The smaller sets between V_1 and V_2 (assuming it is V_1 without loss of generality) are then removed from the graph and new weights for the edges in $V_2 \cup K$ are adjusted such that optimal objective for the smaller problem remains same as the original problem, which requires solving $2^{|K|}$ instances of MaxCut of size $|V_2|$. The new graph can then be decomposed again until the size is small enough for quantum computers. The authors apply this method for 100-vertex 3-regular graph and achieve 90 % approximation ratio on simulators as well as non-trivial improvement over random solutions on Quantinuum trapped-ion quantum computer H1-1.

Other than MaxCut, [51] uses quantum backtracking for tree search algorithms where the backtracking algorithm recursively explores possible assignments and simplifies the subproblem along the way. For some special cases, the authors prove polynomial speedup over classical algorithms, though no experimental results are provided. [52] clusters vertices defined by general QUBOs via community detection and each cluster is then reformulated as a Polynomial Unconstrained Binary Optimization (PUBO) problem whose degree is bounded by $|B_c|$, the cardinality of the boundary vertices belonging to community c. It is able to reduce qubit counts by 40% for solving MaxCut on 3-regular graphs. However, solving PUBOs on either QA or gate-based quantum computers requires quadratization of higher-order terms, which may incur additional qubit overhead in real-world experiments.

On QuEra quantum computers, [43] and [44] discuss hardware implementations, while [53] experiments with nine-point grid graph with up to 289 vertices, achieving superlinear speedup over classical simulated annealing (SA) on the hardest instances. For D-Wave Quantum Annealers (QA), [54] presents the first results of solving MIS problems using QAs on arbitrary graphs. QA is able to find optimal solutions on graphs up to 40 vertices, but struggles to do so with larger graphs (even with 50 and 60 vertices). [55] formulates the wireless network scheduling problem as a MIS problem and demonstrates the potential advantage of QA over SA on graphs up to 30 vertices. [56] employs the DBK (Decomposition, Bounds, K-core) algorithm to reduce the problem size and is able to exactly solve the Maximum Clique problem on graphs with 120 vertices and 6395 edges. Since the DBK algorithm is exact, pre-processing time may scale superpolynomially in the problem size, especially on denser graphs. For gate-based quantum computers, Quantum Approximate Optimization Algorithm and its variants have been proposed to solve MIS and related problems on graphs [1], [57]. However, due to the small number of qubits, sparse connectivity and low qubit coherence times, experimental results on actual gate-based quantum computers remain scarce.

IV. Classical Solvers for MIS

Since we compare the divide-and-conquer-based quantum algorithms we have developed with two classical algorithms for MIS, we discuss the latter now. Many classical solvers have been developed for MIS problems, including Luby's algorithm [16], KaMIS [17], [58], Intel-TreeSearch [59] and Learning what to Defer [60].

We consider Luby's algorithm, a randomized parallel algorithm for solving unweighted MIS, and KaMIS, a state-of-the-art software package for solving both unweighted (using the ReduMIS function to be described later) and weighted MIS problems (using Weighted Local Search).

In Luby's algorithm, each vertex v first marks itself with probability $1/(2d(v))$ unless it has a degree of zero, in which case it is immediately placed in the solution set. In an iteration, each marked vertex v notifies its neighbors of its current degree. If a marked vertex v receives a message from a marked neighbor with a higher degree, it unmarks itself. Otherwise, v notifies all neighbors about its intention to join the independent set. If v receives a message from a neighbor u that u is already in the independent set, no action is taken, and a new iteration begins. Luby's algorithm terminates in $O(\log n)$ iterations with probability greater than $(1 - \frac{1}{n})$.

ReduMIS consists of two components: reduction and evolution. During the reduction steps, the algorithm aims to find a reduced graph (or kernel graph K) from the original graph G such that any MIS found on K can be converted into an MIS for G. The algorithm applies multiple reduction rules sequentially and keeps track of a solution size offset γ such that $|\text{MIS}(K)| + \gamma$ is the solution size for G. For example, any vertex v of degree one (a pendant vertex) is in every MIS, and therefore v and its neighbors u can be removed from G. Additionally, for a degree two vertex v and its disjoint neighbors u and w, either v or $\{u, w\}$ are in every MIS. The vertices v, u and w can then be safely contracted to a single vertex. Other reduction rules are discussed in Section 4.1 in [17].

The second component - evolution - uses an evolutionary algorithm to obtain an MIS once the kernel graphs are obtained. The algorithm starts with a population of maximal independent sets from greedy algorithms and evolves them until no improvement can be made. At each round, two independent sets are selected from the population which then exchange blocks of vertices through vertex separators. Let I_1 and I_2 be selected from the population, and let (S, V_1, V_2) denote the separator and two disjoint subgraphs of G. We can create two new independent sets O_1, O_2 from $(V_1 \cap I_1) \cup (V_2 \cap I_2)$ and $(V_1 \cap I_2) \cup (V_2 \cap I_1)$. Since they are not necessarily maximal, O_1 and O_2 are improved further by additional iterations of local search [61].

Instead of returning after performing the evolutionary algorithm on K, ReduMIS selects an independent set of the highest cardinality from the population, removes λ vertices with the smallest degrees and their neighbors from K to obtain a new kernel graph K', and re-runs the algorithm for K'. (The intent is that the vertices removed will be added to the new independent set computed from the new graph.) The intuition behind this is that the smallest-degree vertices are more likely to belong to an MIS than higher-degree vertices. The process iterates until no more reductions can be found. Typically, λ is set to $|V|/10$ in ReduMIS. A pseudo-code for the algorithm is provided in Algorithm 2.

Algorithm 2 ReduMIS

```
function KaMIS_Solve_MIS($G = (V, E)$)
    global $S \leftarrow \emptyset$
    ReduMIS($G, 0$)
    Covert $S$ back to a MIS for $G$ and obtain $S_G$
    return $S_G$
function ReduMIS($G = (V, E), \gamma$)
    if $G$ is not reducible or time limit hits then
        return
    else
        Compute a kernel graph $K$ of $G$ with a solution offset $\theta$
        $I^* \leftarrow$ Evo_MIS($K$)
        if $|I^*| + \theta + \gamma > |S|$ then
            Update $S$
        $U \leftarrow \lambda$ smallest-degree vertices in $I^*$
        $N(U) \leftarrow$ neighbors of $U$ in $K$
        $K' \leftarrow K[V \setminus (U \cup N(U)]$
        ReduMIS($K', \gamma + \theta + |U|$)
    return
function Evo_MIS($K = (V, E)$)
    Greedily generate $N$ maximal independent sets of $K$ as the initial population $P$
    $I_1, I_2 \leftarrow$ two largest independent sets from $P$
    $O_1, O_2 \leftarrow$ Block_Swap($K, I_1, I_2$)
    Improve $O_1, O_2$ by local search
    Evict smaller independent sets from $P$ to make room for $O_1, O_2$
    $I^* =$ independent set from $P$ with highest cardinality
    return $I^*$
function Block_Swap($G = (V, E), I_1, I_2$)
    Compute a 2-way vertex separator $V = V_1 \cup V_2 \cup S$
    $O_1 \leftarrow (V_1 \cap I_1) \cup (V_2 \cap I_2)$
    $O_2 \leftarrow (V_1 \cap I_2) \cup (V_2 \cap I_1)$
    return $O_1, O_2$
```

KaMIS Weighted Local Search follows the same strategy as ReduMIS - reduce first and solve second. A similar set of reduction rules are defined in [58]. For example, if a vertex v and its neighbors $N(v)$ in the current solution I satisfy the condition $w(N(v) \cap I) < w(v)$, then v can be swapped in place of its neighbors. After reducing the graph, KaMIS uses the hybrid iterated local search (HILS) heuristic [62] to find the independent set.

V. Results

A. Effects of Subproblem Cutoff on Solution Qualities

We expect the independent set sizes from Algorithm 1 to increase as we enlarge the cutoff C and reduce the number of recursion steps. In Figure 1, we run Algorithm 1 on four graphs with varying cutoff sizes using simulated annealing [13] implemented in *dwave-neal* and compare the weights of the independent sets obtained from our algorithm to those of KaMIS. It is clear that larger cutoffs yield better solutions, and the growth rate decreases near the end due to the moderate sizes of the test graphs. We use a cutoff of $C = 200$ for our experiments on the D-Wave Quantum Annealer.

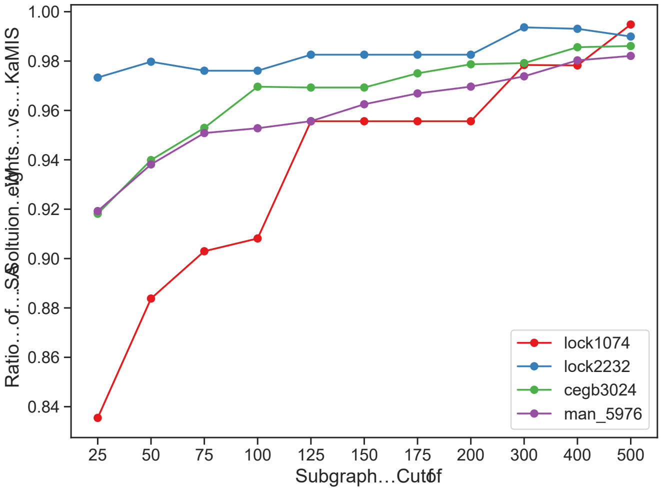


Fig. 1: Weights of independent sets obtained from Algorithm 1 using various cutoff values C.

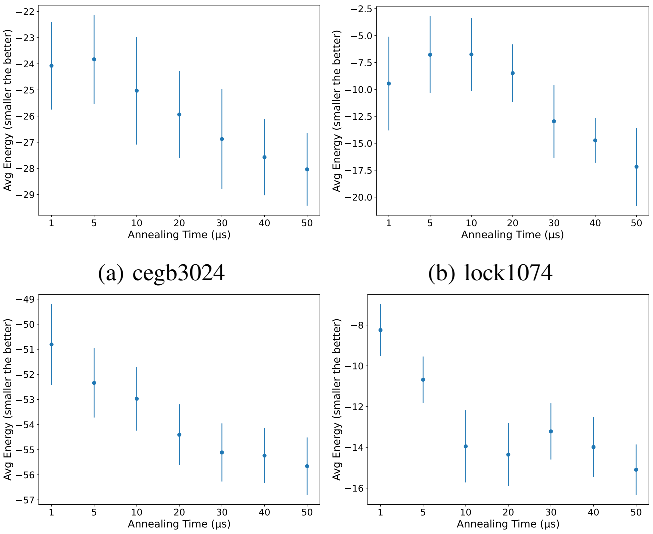
B. Results on D-Wave Quantum Annealer

We benchmark Algorithm 1 on the D-Wave Quantum Annealer Advantage System 6.4. Our test graphs are graphs with small balanced separators, as described in Section II-A, and are planar graphs or three-dimensional meshes downloaded from the *SuiteSparse* [63] collection.

Solution qualities of D-Wave Quantum Annealers are affected by several parameters, including annealing time and the penalty coefficient in Eq 6. However, finding the best parameters are usually as hard as solving the optimization problem itself [64]. In the next two Subsections we provide empirical support for our choices of these two parameters.

1) Setting Annealing Time: For optimization problems, longer annealing time generally leads to higher probability of success [65]. In Figure 2, we investigate the effects of varying annealing times for subgraphs from the following graphs in *SuiteSparse*: *cegb3024*, *lock1074*, *lock2232* and *man_5976*. The penalty coefficient in Eq 6 is set to 2.0 and the embedding stays constant. We confirm, empirically, that longer annealing time yields samples with lower energy, and in turn, better cardinalities or weights. Therefore, we set the annealing time to be 50 µs in our experiments.

2) Setting the Penalty Coefficient: Since the QUBO energy in Eq 6 is dependent on the penalty coefficients, we probe the effects of different penalty coefficients using independent set weights instead of sample energy. After we collect samples from QPUs, each sample is repaired as described in the previous section using Algorithm 1. We compare weights of independent sets against penalty coefficients in Figure 3. Though no penalty coefficient works best for all instances, we



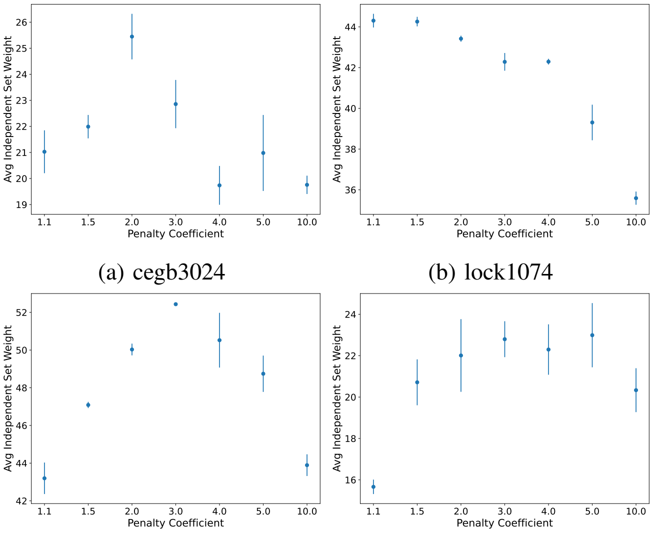
(a) cegb3024

(b) lock1074

(c) lock2232

(d) man_5976

Fig. 2: Effects of different annealing times on D-Wave sampling qualities. For each annealing time, we called D-Wave QPU 20 times requesting 100 samples each. The error bars are bootstrap standard deviations of sample energy averaged over QPU calls.



(a) cegb3024

(b) lock1074

(c) lock2232

(d) man_5976

Fig. 3: Effects of penalty coefficients on D-Wave sample qualities. For each penalty coefficient, we called D-Wave QPU 20 times requesting 100 samples each. The error bars are bootstrap standard deviations of sample energy averaged over QPU calls.

observe that 2.0 is a good starting value that works well in most cases, and it is used in our experiments.

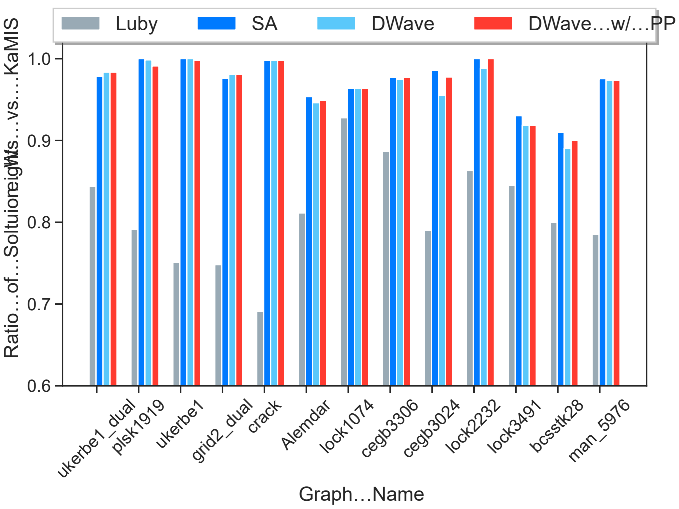


Fig. 4: Comparison of independent set cardinalities obtained from Algorithm 1 using simulated annealing, D-Wave Quantum Annealer, KaMIS and Luby's Algorithm. The y-axis tracks the ratios of solution cardinalities of different solvers against KaMIS, which is treated as the baseline. For KaMIS and Luby's algorithm, we run the programs ten times with different seeds and report the best results found.

3) Experimental Results: We benchmark Algorithm 1 on 13 unweighted graphs, described in Table I. Results are obtained using simulated annealing and D-Wave Quantum Annealer Advantage System 6.4. The cutoff C is set to 200 vertices and the separators are computed using KaHIP [66]. One thousand samples are collected from D-Wave over ten QPU calls for each subproblem, and α is set to ten when determining the best sample to return. Additionally, before pruning the samples, we run steepest descent on each sample until local minima are reached, and indicate these results as D-Wave with post-processing (PP) in Figure 4. The solutions from QUBO solvers are then compared with KaMIS and Luby's algorithm. We are able to outperform Luby's Algorithm consistently and obtain independent sets with cardinalities higher than 95% of those obtained using KaMIS.

To generate weighted instances, we take the same graphs listed in Table I, and randomly assign integers from one to hundred to each vertex. The benchmark is presented in Figure 5. The solution qualities drop slightly compared to the unweighted instances due to the additional ruggedness in the energy landscapes introduced by non-uniform vertex weights.

To measure runtime, we divide our algorithm into four components: separator computation with KaHIP, embedding computation, D-Wave sampling and post-processing (PP). D-Wave sampling time includes both annealing time and programming time. In Table II, we compare the runtime for solving MWIS instances using KaMIS and Algorithm 1 with D-Wave. The KaMIS algorithm relies heavily on graph reductions to solve MIS problems on smaller kernel graphs using fewer iterations, and this is the reason for its faster runtimes. For *ukerbe1* and *grid2_dual*, KaMIS is able to find a significantly smaller kernel graph than other instances, and the algorithm terminates in one

TABLE I: Statistics for graphs used in the experiments from *SuiteSparse*. Graphs are sorted in increasing order of $|E|$ with planar graphs at the top. For each problem we report its number of vertices, number of edges, maximum degree (Δ), average degree, standard deviation in the degree and source.

Name	Planar	$\|V\|$	$\|E\|$	Δ	Avg Deg	Deg Std	Source
ukerbe1_dual	Y	1866	3538	4	3.79	0.41	2D finite element problem
plsk1919	Y	1919	4831	6	5.03	1.16	Platzman skew-symmetric finite difference three ocean model
ukerbe1	Y	5981	7852	8	2.63	0.97	2D finite element problem
grid2_dual	Y	3136	6112	4	3.90	0.31	2D finite element problem
crack	Y	10240	30380	9	5.93	1.89	2D finite element problem
Alemdar	N	6245	18168	9	5.82	2.36	finite element problem from Bulent Alemdar
lock1074	N	1074	25275	95	47.07	15.88	finite element from Lockheed gyro problem
cegb3306	N	3306	35847	53	21.67	11.25	2D finite element problem
cegb3024	N	3024	38426	67	25.41	9.37	finite element from 2D reactor core section
lock2232	N	2232	39072	47	35.01	11.20	finite element from Lockheed tower problem
lock3491	N	3491	78514	112	44.98	12.88	finite element from Lockheed cross-cone problem
bcsstk28	N	4410	107307	93	48.67	9.68	Solid element model (MSC Nastran)
man_5976	N	5976	109582	50	36.67	10.92	finite element problem from Manteuffel

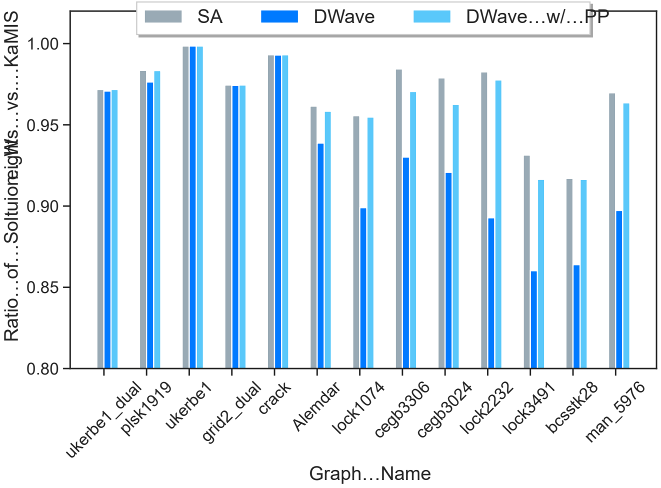


Fig. 5: Comparison of independent set weights obtained from Algorithm 1 using simulated annealing, D-Wave Quantum Annealer, and KaMIS. For KaMIS, we run the program ten times with different seeds and report the best results found.

iteration. It is worth noting that the time taken for solving MWIS instances on D-Wave, once embedded, is constant regardless the sizes of the problems. It is also observed that the time to embed the graphs on the Pegasus topology of the D-Wave Quantum Annealer is fairly large relative to the other steps. This can be avoided by embedding the complete graph on the D-Wave in every instance, but it would drastically reduce the size of the subgraphs that could be solved.

C. Results on QuEra Quantum Computer

We use nine-point grid graphs with m rows and n columns to benchmark Algorithm 3 and randomly remove $p\%$ vertices from it. We confine test cases to such graphs in order to ensure that any subgraph obtained from vertex separators are also unit disk graphs, therefore eliminating potential embedding overheads. Figure 6 is one such example with $m = n = 16$ and $p = 20$. The nine-point grid graphs are then embedded on a square lattice which can be easily prepared on QuEra Quantum Computers. Due to the unique geometric constraints of the chip where atoms are required to be placed at least $4\,\mu\mathrm{m}$ apart on a $76\,\mu\mathrm{m}$ and $75\,\mu\mathrm{m}$ square, instead of finding vertex separators using KaHIP as described in Section IV, we resort to using vertical (a column) or horizontal (a row) separators on the square lattice where the grid graphs are embedded. Algorithm 3 describes our updated algorithm. One notable change from Algorithm 1 is that cutoffs are defined in terms of sizes for both dimensions of the grid graphs $G = (m, n, p)$. For a cutoff D, we use the quantum computer to solve MIS problems on G if $m \leq D$ and $n \leq D$.

TABLE II: Runtime (in seconds) for solving MWIS instances using KaMIS and Algorithm 1 on D-Wave Quantum Annealer.

Name	KaMIS	Separator	Embedding	D-Wave	PP
ukerbe1_dual	9.85	2.74	4.43	4.88	5.2
plsk1919	11.74	3.77	7.97	4.83	5.13
ukerbe1	0.11	7.32	15.05	11.63	12.32
grid2_dual	0.12	4.9	12.11	5.74	13.47
crack	7.64	25.91	71.59	20.52	22.64
Alemdar	60.28	13	49.65	11.23	24.53
lock1074	8.18	11.14	214.98	2.6	1.39
cegb3306	2.31	12.42	135.98	6.85	5.96
cegb3024	26.05	20.95	216.7	5.69	5.82
lock2232	5.55	17.1	187.19	5.55	2.94
lock3491	8.94	31	595.22	6.44	6.15
bcsstk28	11.57	57.07	759.4	9.41	6.04
man_5976	50.6	62.8	738.14	11.18	9.84

Since our QuEra access lacks the ability to set the parameters Ω and Δ individually for each site as described in Eq 3, we can only benchmark our algorithm for unweighted MIS instances. For each subgraph, we request 200 samples from the device and only count ones without defects in the pre-sequence (in other words, samples whose initialization is correct). α is set to 100 due to the smaller sample counts. Algorithm 3 requires the same post-processing time as Algorithm 1 as the removed vertices in the nine-point grid graphs are randomly chosen which ensures balanced separators at each level. The test instances are described in Table III and results are presented in Figure 7. In all the instances, the quantum algorithm on QuEra computes weights for MIS that are higher than

Algorithm 3 MIS Solver using QuEra

```
function SOLVE_MIS($G = (m, n, p)$, $N$, $\alpha$, $D$)
    // $D$ is the cutoff for number of vertices in both dimensions and $\alpha$ the number of samples for post-processing.
    if $m \leq D$ & $n \leq D$ then
        $I \leftarrow$ Solve_MIS_Quantum($G$, $N$, $\alpha$)
    else
        $A, B, S \leftarrow$ Find_Grid_Separator($G$)
        $I_A \leftarrow$ Solve_MIS($A$, $N$, $\alpha$, $D$)
        $I_B \leftarrow$ Solve_MIS($B$, $N$, $\alpha$, $D$)
        for $(v, u)$ in $E$, $v \in S$ and $u \in I_A \cup I_B$ do
            $S = S \setminus \{v\}$
        $I_S \leftarrow$ Solve_MIS($S$, $N$, $\alpha$, $D$)
        $I \leftarrow I_A \cup I_B \cup I_S$
        $I \leftarrow$ MIS_Greedy_Improvement($I$, $G$)
    return $I$
function FIND_GRID_SEPARATOR($G = (m, n, p)$)
    if $m \geq n$ then
        $S \leftarrow$ set of vertices on the $\lceil \frac{m}{2} \rceil$-th row
        $A \leftarrow$ set of vertices above $S$
        $B \leftarrow$ set of vertices below $S$
    else
        $S \leftarrow$ set of vertices on the $\lceil \frac{n}{2} \rceil$-th column
        $A \leftarrow$ set of vertices on the left of $S$
        $B \leftarrow$ set of vertices on the right of $S$
    return A,B,S
```

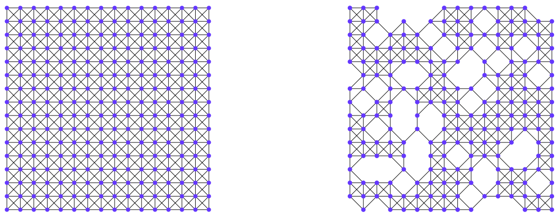
85% of the weights obtained from KaMIS. We also note that the denser the graph, the worse the hybrid algorithm performs. This can be attributed to the increasing number of MIS solutions in dense graphs, i.e., MIS degeneracy, which proves to be challenging to solve [53], [67].

TABLE III: Statistics for nine-point grid graphs with holes used in the experiments, including number of vertices, number of edges, maximum degree, average degree, and standard deviation in the degree.

m	n	p	Planar	$\lvert V \rvert$	$\lvert E \rvert$	Δ	Avg Deg	Deg Std
32	32	0.3	Y	307	338	6	2.20	1.29
32	32	0.5	N	512	980	7	3.83	1.52
32	32	0.8	N	819	2484	8	6.07	1.38
32	48	0.3	Y	460	530	6	2.30	1.27
32	48	0.5	N	768	1464	8	3.81	1.47
32	48	0.8	N	1228	3760	8	6.12	1.31
48	48	0.3	N	691	826	6	2.39	1.38
48	48	0.5	N	1152	2203	8	3.82	1.48
48	48	0.8	N	1843	5695	8	6.18	1.27

VI. Discussion

In this paper, we propose a hybrid quantum-classical algorithm based on divide and conquer to leverage the computational capabilities of different quantum computers for solving large-scale Maximum Independent Set and Maximum Weighted Independent Set problems. Our method focuses on separable graphs, i.e., classes of graphs with small vertex separators. We report results from planar and finite element meshes, which have provably small separators. The algorithm exploits vertex separators to efficiently divide the original problem into balanced subproblems that can be solved on current quantum computers. The MIS for the entire graph is obtained from the union of independent sets from the subgraphs together with independent sets computed from the separators. We benchmark our algorithms using both the D-Wave Quantum Annealer and the QuEra Quantum Computer with Neutral Atoms on graphs with up to 10,240 vertices and compute solutions comparable to those obtained from KaMIS, a state-of-the-art classical MIS solver. Furthermore, we demonstrate the scalability of our algorithms (Figure 1) which enables them to take advantage of new NISQ hardware with higher qubit counts, connectivity and coherence times. As we employ such hybrid algorithms in the NISQ-era, new generations of quantum hardware may finally bring the capabilities of solving classically intractable problems with potential quantum speed-up [4], [53].



(a) Original

(b) With holes

Fig. 6: An example 16x16 nine-point grid graph with 20% vertices randomly removed.

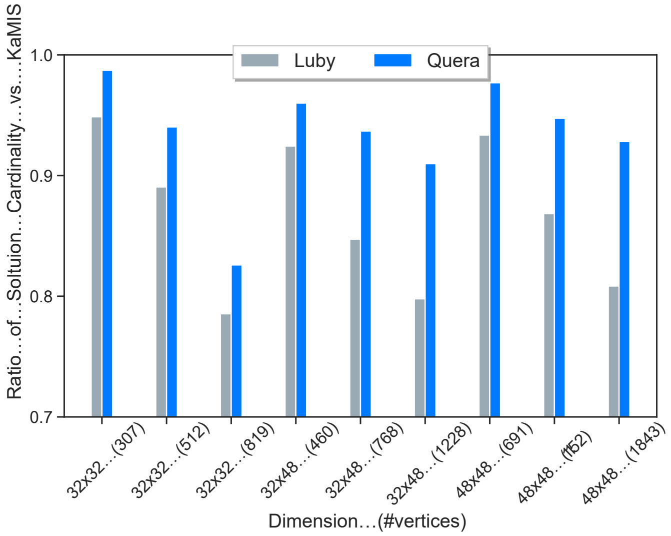


Fig. 7: Cardinalities of independent sets obtained from Algorithm 3 using QuEra Quantum Computer, KaMIS and Luby's Algorithm on nine-point grid graphs with holes. For KaMIS and Luby's algorithm, we run the programs ten times with different seeds and report the best results found.

Future research includes applying the vertex separator-based divide and conquer framework to other graph problems, including Maximum k-Colorable Subgraph and Minimum Dominating Set. Another interesting extension of this work is to graph problems where the use of edge-separators to obtain small subproblems is advantageous. In Section IV, we also mention algorithms for gate-based quantum computers which we have not implemented yet.

Acknowledgment

Our work was supported by the Center for Quantum Technologies (CQT) at Purdue University, which is an Industry-University Cooperative Research Center (IUCRC) funded through the US National Science Foundation (NSF) under Grant No. 2224960. We also thank Amazon Web Services for providing access to the QuEra quantum computer, and Dr. Brajesh Gupt for answering our questions. We are grateful to Prof. Arnab Banerjee and Prof. David Bernal (both at Purdue) for several discussions.

References

[1] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm," 2014, arXiv:1411.4028.

[2] M. Motta, C. Sun, A. T. K. Tan, M. J. O'Rourke, E. Ye, A. J. Minnich, F. G. S. L. Brandão, and G. K.-L. Chan, "Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution," *Nature Physics*, vol. 16, no. 2, pp. 205–210, 2020. [Online]. Available: https://www.nature.com/articles/s41567-019-0704-4

[3] B. Fuller, C. Hadfield, J. R. Glick, T. Imamichi, T. Itoko, R. J. Thompson, Y. Jiao, M. M. Kagele, A. W. Blom-Schieber, R. Raymond, and A. Mezzacapo, "Approximate Solutions of Combinatorial Problems via Quantum Relaxations," 2021, arXiv:2111.03167.

[4] A. Abbas, A. Ambainis, B. Augustino, A. Bärtschi, H. Buhrman, C. Coffrin, G. Cortiana, V. Dunjko, D. J. Egger, B. G. Elmegreen, N. Franco, F. Fratini, B. Fuller, J. Gacon, C. Gonciulea, S. Gribling, S. Gupta, S. Hadfield, R. Heese, G. Kircher, T. Kleinert, T. Koch, G. Korpas, S. Lenk, J. Marecek, V. Markov, G. Mazzola, S. Mensa, N. Mohseni, G. Nannicini, C. O'Meara, E. P. Tapia, S. Pokutta, M. Proissl, P. Rebentrost, E. Sahin, B. C. B. Symons, S. Tornow, V. Valls, S. Woerner, M. L. Wolf-Bauwens, J. Yard, S. Yarkoni, D. Zechiel, S. Zhuk, and C. Zoufal, "Quantum Optimization: Potential, Challenges, and the Path Forward," 2023, arXiv:2312.02279.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. USA: W. H. Freeman & Co., 1990.

[6] K. M. Alzoubi, P.-J. Wan, and O. Frieder, "Distributed heuristics for connected dominating sets in wireless ad hoc networks," *Journal of Communications and Networks*, vol. 4, no. 1, pp. 22–29, 2002. [Online]. Available: https://ieeexplore.ieee.org/document/6596929

[7] L. Brotcorne, G. Laporte, and F. Semet, "Fast heuristics for large scale covering-location problems," *Computers & Operations Research*, vol. 29, no. 6, pp. 651–665, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030505489900088X

[8] E. J. Gardiner, P. J. Artymiuk, and P. Willett, "Clique-detection algorithms for matching three-dimensional molecular structures," *Journal of Molecular Graphics & Modelling*, vol. 15, no. 4, pp. 245–253, 1997.

[9] E. J. Gardiner, P. Willett, and P. J. Artymiuk, "Graph-Theoretic Techniques for Macromolecular Docking," *Journal of Chemical Information and Computer Sciences*, vol. 40, no. 2, pp. 273–279, 2000. [Online]. Available: https://doi.org/10.1021/ci990262o

[10] E. Harley, A. Bonner, and N. Goodman, "Uniform integration of genome mapping data using intersection graphs," *Bioinformatics*, vol. 17, no. 6, pp. 487–494, 2001.

[11] E. Balas and C. S. Yu, "Finding a Maximum Clique in an Arbitrary Graph," *SIAM Journal on Computing*, vol. 15, no. 4, pp. 1054–1068, 1986. [Online]. Available: http://epubs.siam.org/doi/10.1137/0215075

[12] R. Carraghan and P. M. Pardalos, "An exact algorithm for the maximum clique problem," *Operations Research Letters*, vol. 9, no. 6, pp. 375–382, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/016763779090057C

[13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[14] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, Inc., 1989.

[15] T. N. Bui and P. H. Eppley, "A hybrid genetic algorithm for the maximum clique problem," *Proc. Sixth International Conf. on Genetic Algorithms*, pp. 478–484, 1995. [Online]. Available: https://cir.nii.ac.jp/crid/1570291224111795456

[16] M. Luby, "A simple parallel algorithm for the maximal independent set problem," in *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, ser. STOC '85. New York, NY, USA: Association for Computing Machinery, 1985, pp. 1–10.

[17] S. Lamm, P. Sanders, C. Schulz, D. Strash, and R. F. Werneck, "Finding near-optimal independent sets at scale," *J. Heuristics*, vol. 23, no. 4, pp. 207–229, 2017. [Online]. Available: https://doi.org/10.1007/s10732-017-9337-x

[18] R. J. Lipton and R. E. Tarjan, "A Separator Theorem for Planar Graphs," *SIAM Journal on Applied Mathematics*, vol. 36, no. 2, pp. 177–189, 1979.

[19] M. R. Garey and D. S. Johnson, "The Rectilinear Steiner Tree Problem is NP-Complete," *SIAM Journal on Applied Mathematics*, vol. 32, no. 4, pp. 826–834, 1977.

[20] O. J. Murphy, "Computing independent sets in graphs with large girth," *Discrete Applied Mathematics*, vol. 35, no. 2, pp. 167–170, 1992.

[21] B. S. Baker, "Approximation algorithms for NP-complete problems on planar graphs," in *24th Annual Symposium on Foundations of Computer Science (Sfcs 1983)*, 1983, pp. 265–273.

[22] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis, "Separators for sphere-packings and nearest neighbor graphs," *Journal of the ACM*, vol. 44, no. 1, pp. 1–29, 1997.

[23] P. Carmi, M. K. Chiu, M. J.Katz, M. Korman, Y. Okamoto, A. van Renssen, M. Roeloffzen, T. Shiitada, and S. Smorodinsky, "Balanced line separators of unit disk graphs," *Computational Geometry: Theory and Applications*, vol. 86, 2020, 101575.

[24] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, no. 1, pp. 165–177, 1990.

[25] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz, "Simple heuristics for unit disk graphs," *Networks*, vol. 35, pp. 59–68, 1995.

[26] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, no. 1, p. 015002, 2018.

[27] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum Computation by Adiabatic Evolution," 2000, arXiv:quant-ph/0001106.

[28] T. Kato, "On the Adiabatic Theorem of Quantum Mechanics," *Journal of the Physical Society of Japan*, vol. 5, no. 6, pp. 435–439, 1950.

[29] J. Brooke, D. Bitko, T. F., Rosenbaum, and G. Aeppli, "Quantum Annealing of a Disordered Magnet," *Science*, vol. 284, no. 5415, pp. 779–781, 1999.

[30] A. D. King, J. Raymond, T. Lanting, R. Harris, A. Zucca, F. Altomare, A. J. Berkley, K. Boothby, S. Ejtemaee, C. Enderud, E. Hoskinson, S. Huang, E. Ladizinsky, A. J. R. MacDonald, G. Marsden, R. Molavi, T. Oh, G. Poulin-Lamarre, M. Reis, C. Rich, Y. Sato, N. Tsai, M. Volkmann, J. D. Whittaker, J. Yao, A. W. Sandvik, and M. H. Amin, "Quantum critical dynamics in a 5,000-qubit programmable spin glass," *Nature*, vol. 617, no. 7959, pp. 61–66, 2023.

[31] G. E. Santoro, R. Martoňák, E. Tosatti, and R. Car, "Theory of Quantum Annealing of an Ising Spin Glass," *Science*, vol. 295, no. 5564, pp. 2427–2430, 2002.

[32] K. Boothby, P. Bunyk, J. Raymond, and A. Roy, "Next-Generation Topology of D-Wave Quantum Processors," 2020, arXiv:2003.00133.

[33] V. Choi, "Minor-embedding in adiabatic quantum computation: I. The parameter setting problem," *Quantum Information Processing*, vol. 7, no. 5, pp. 193–209, 2008.

[34] ——, "Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design," *Quantum Information Processing*, vol. 10, no. 3, pp. 343–353, 2011.

[35] J. Cai, W. G. Macready, and A. Roy, "A practical heuristic for finding graph minors," 2014, arXiv:1406.2741.

[36] F. Barahona, “On the computational complexity of Ising spin glass models,” *Journal of Physics A: Mathematical and General*, vol. 15, no. 10, pp. 3241–3253, 1982.

[37] F. Glover, G. Kochenberger, and Y. Du, “A Tutorial on Formulating and Using QUBO Models,” 2019, arXiv:1811.11538.

[38] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang, “The unconstrained binary quadratic programming problem: A survey,” *Journal of Combinatorial Optimization*, vol. 28, no. 1, pp. 58–81, 2014.

[39] A. Lucas, “Ising formulations of many NP problems,” *Frontiers in Physics*, vol. 2, 2014.

[40] D. Pastorello and E. Blanzieri, “Quantum annealing learning search for solving QUBO problems,” *Quantum Information Processing*, vol. 18, no. 10, p. 303, 2019.

[41] V. Lienhard, S. de Léséleuc, D. Barredo, T. Lahaye, A. Browaeys, M. Schuler, L.-P. Henry, and A. M. Läuchli, “Observing the Space- and Time-Dependent Growth of Correlations in Dynamically Tuned Synthetic Ising Models with Antiferromagnetic Interactions,” *Physical Review X*, vol. 8, no. 2, p. 021070, 2018.

[42] H. Labuhn, D. Barredo, S. Ravets, S. de Léséleuc, T. Macrì, T. Lahaye, and A. Browaeys, “Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum Ising models,” *Nature*, vol. 534, no. 7609, pp. 667–670, 2016.

[43] H. Pichler, S.-T. Wang, L. Zhou, S. Choi, and M. D. Lukin, “Quantum Optimization for Maximum Independent Set Using Rydberg Atom Arrays,” 2018, arXiv:1808.10816.

[44] M.-T. Nguyen, J.-G. Liu, J. Wurtz, M. D. Lukin, S.-T. Wang, and H. Pichler, “Quantum Optimization with Arbitrary Connectivity Using Rydberg Atom Arrays,” *PRX Quantum*, vol. 4, no. 1, p. 010316, 2023.

[45] V. Choi, “The Effects of the Problem Hamiltonian Parameters on the Minimum Spectral Gap in Adiabatic Quantum Optimization,” *Quantum Information Processing*, vol. 19, no. 3, p. 90, 2020.

[46] T. Tomesh, Z. H. Saleem, M. A. Perlin, P. Gokhale, M. Suchara, and M. Martonosi, “Divide and Conquer for Combinatorial Optimization and Distributed Quantum Computation,” in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE Computer Society, 2023, pp. 1–12.

[47] J. Li, M. Alam, and S. Ghosh, “Large-Scale Quantum Approximate Optimization via Divide-and-Conquer,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 6, pp. 1852–1860, 2023.

[48] Z. Zhou, Y. Du, X. Tian, and D. Tao, “QAOA-in-QAOA: Solving Large-Scale MaxCut Problems on Small Quantum Machines,” *Physical Review Applied*, vol. 19, no. 2, p. 024027, 2023.

[49] A. Dembo, A. Montanari, and S. Sen, “Extremal cuts of sparse random graphs,” *The Annals of Probability*, vol. 45, no. 2, pp. 1190–1217, 2017.

[50] M. Ponce, R. Herrman, P. C. Lotshaw, S. Powers, G. Siopsis, T. Humble, and J. Ostrowski, “Graph decomposition techniques for solving combinatorial optimization problems with variational quantum algorithms,” 2023.

[51] M. Rennela, S. Brand, A. Laarman, and V. Dunjko, “Hybrid divide-and-conquer approach for tree search algorithms,” *Quantum*, vol. 7, p. 959, 2023.

[52] G. G. Guerreschi, “Solving Quadratic Unconstrained Binary Optimization with divide-and-conquer and quantum algorithms,” 2021, arXiv:2101.07813.

[53] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, G. Liu, R. Samajdar, Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, S. Zhous, S. Choi, H. Pichler, S. T. Wang, M. Greiner, V. Vuletić, and M. D. Lukin, “Quantum optimization of maximum independent set using Rydberg atom arrays,” *Science*, vol. 376, pp. 1209–1215, 2022.

[54] S. Yarkoni, A. Plaat, and T. Back, “First Results Solving Arbitrarily Structured Maximum Independent Set Problems Using Quantum Annealing,” in *2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018, pp. 1–6.

[55] C. Wang and E. Jonckheere, “Simulated versus reduced noise quantum annealing in maximum independent set solution to wireless network scheduling,” *Quantum Information Processing*, vol. 18, no. 1, p. 6, 2018.

[56] E. Pelofske, G. Hahn, and H. N. Djidjev, “Solving larger maximum clique problems using parallel quantum annealing,” *Quantum Information Processing*, vol. 22, no. 5, p. 219, 2023.

[57] 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*, vol. 12, no. 2, p. 34, 2019.

[58] S. Lamm, C. Schulz, D. Strash, R. Williger, and H. Zhang, “Exactly solving the maximum weight independent set problem on large real-world graphs,” in *Proceedings of the Twenty-First Workshop on Algorithm Engineering and Experiments, ALENEX 2019*. SIAM, 2019, pp. 144–158. [Online]. Available: https://doi.org/10.1137/1.9781611975499.12

[59] Z. Li, Q. Chen, and V. Koltun, “Combinatorial Optimization with Graph Convolutional Networks and Guided Tree Search,” in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[60] S. Ahn, Y. Seo, and J. Shin, “Learning What to Defer for Maximum Independent Sets,” 2020, arXiv:2006.09607.

[61] D. V. Andrade, M. G. C. Resende, and R. F. Werneck, “Fast local search for the maximum independent set problem,” *Journal of Heuristics*, vol. 18, no. 4, pp. 525–547, 2012.

[62] B. Nogueira, R. G. S. Pinheiro, and A. Subramanian, “A hybrid iterated local search heuristic for the maximum weight independent set problem,” *Optimization Letters*, vol. 12, no. 3, pp. 567–583, 2018.

[63] T. A. Davis and Y. Hu, “The university of Florida sparse matrix collection,” *ACM Transactions on Mathematical Software*, vol. 38, no. 1, pp. 1:1–1:25, 2011.

[64] E. Alessandroni, S. Ramos-Calderer, I. Roth, E. Traversi, and L. Aolita, “Alleviating the quantum Big-M problem,” 2024, arXiv:2307.10379.

[65] D. Willsch, M. Willsch, C. D. Gonzalez Calaza, F. Jin, H. De Raedt, M. Svensson, and K. Michielsen, “Benchmarking Advantage and D-Wave 2000Q quantum annealers with exact cover problems,” *Quantum Information Processing*, vol. 21, no. 4, p. 141, 2022.

[66] P. Sanders and C. Schulz, “Think locally, act globally: Highly balanced graph partitioning,” in *Experimental Algorithms, 12th International Symposium, SEA 2013, Rome, Italy, June 5-7, 2013. Proceedings*, vol. 7933. Springer, 2013, pp. 164–175.

[67] R. S. Andrist, M. J. A. Schuetz, P. Minssen, R. Yalovetzky, S. Chakrabarti, D. Herman, N. Kumar, G. Salton, R. Shaydulin, Y. Sun, M. Pistoia, and H. G. Katzgraber, “Hardness of the Maximum Independent Set Problem on Unit-Disk Graphs and Prospects for Quantum Speedups,” *Physical Review Research*, vol. 5, no. 4, p. 043277, 2023.