

Identifying the minor set cover of dense connected bipartite graphs via random matching edge sets

Kathleen E. Hamilton · Travis S. Humble

Received: date / Accepted: date

Abstract Using quantum annealing to solve an optimization problem requires minor embeddings of a logic graph into a known hardware graph. In an effort to reduce the complexity of the minor embedding problem, we introduce the minor set cover (MSC) of a known graph \mathcal{G} : a subset of graph minors which contain any remaining minor of the graph as a subgraph. Any graph that can be embedded into \mathcal{G} will be embeddable into a member of the MSC. Focusing on embedding into the hardware graph of commercially available quantum annealers, we establish the MSC for a particular known virtual hardware, which is a complete bipartite graph. We show that the complete bipartite graph $K_{N,N}$ has a MSC of N minors, from which K_{N+1} is identified as the largest clique minor of $K_{N,N}$. The case of determining the largest clique minor of hardware with faults is briefly discussed but remains an open question.

Keywords minor embedding, adiabatic quantum computing, quantum annealing, clique minor, graph theory

1 Introduction

Adiabatic quantum computation uses a continuous-time process to evolve the state of a quantum register [1]. Whereas the register elements are represented by quantum physical subsystems that can store qubits of information, the continuous-time evolution depends explicitly on a Hamiltonian that defines the interactions between register elements [2]. An ideal Hamiltonian may allow for arbitrary interactions between elements, but physical and technological limitations often prevent fabrication of arbitrary interactions or forms of connections in actual devices. A prominent example is found in the quantum annealer developed by D-Wave Systems, Inc. [3], which uses a well-defined hardware

K.E. Hamilton · T.S.Humble
Oak Ridge National Laboratory, 1 Bethel Valley Rd, Oak Ridge, TN 37831, USA
E-mail: hamiltonke@ornl.gov

connectivity graph called the Chimera lattice and implements problems which can be described using the Ising Hamiltonian with two-body interactions.

The problem of expressing an arbitrary Hamiltonian in the presence of limited connections poses a prominent concern for quantum annealing applications [4]. Presently we consider the input to this programming process to be a well-defined logical Ising Hamiltonian. The logical Ising model is known to capture a broad class of different problems and may also be presented in quadratic unconstrained binary optimization (QUBO) form [5]. Choi originally formulated the process of programming a logical problem as graph minor embedding, in which a graph representing the dependency of the input Hamiltonian is mapped into the targeted hardware graph [6, 7]. In general, graph minor embedding requires each logical vertex to be mapped into a connected subtree of the hardware graph, and Choi’s TRIAD algorithm yielded a deterministic method for embedding a complete graph into the Chimera lattice. There are many approaches for determining if a graph can be embedded into another graph, including: identifying useful graph qualities[8, 9, 10], establishing a set of known embeddings as a “lookup table” [11, 12] or other heuristic methods [13, 14].

More recently, Cai, Macready, and Roy (CMR) have presented a randomized algorithm for generating an embedding [13]. Their approach is based on employing Djikstra’s algorithm to find the shortest-path between randomly mapped logical vertices. The CMR algorithm has proven useful for embedding arbitrary input graphs in current hardware because it can find smaller embeddings than using either the TRIAD algorithm [7] or maximal minor embedding [11]. However, the method is not guaranteed to succeed and has a worst case complexity that scales as $O(n^9)$ with the input graph order n (though average case behavior appears to be $O(n^3)$). The CMR embedding algorithm represents a significant portion of the time needed for a quantum annealing workflow, and for even modest problem sizes it can far exceed the time required for executing a quantum annealing schedule [15]. Problem instances represented by large but incompletely connected input graphs must use embeddings that are both resource efficient and time efficient in order to ensure fast and correct solutions. Examples include dynamic job scheduling [16] and route planning [17], as well as time-dependent fault-detection [18]. Alleviating the classical processing bottleneck while retaining the resource efficiency of the CMR algorithm is therefore an important problem for solving optimization problems with quantum annealing and integrating these quantum processing units into future computing systems [19].

In this contribution, we present a quasi-deterministic method for graph minor embedding that takes advantage of a virtual hardware abstraction. Our method builds upon two recent embedding concepts: “maximal minor embedding” developed in [11], which characterizes finding the most efficient embedding with respect to the minimal number of hardware nodes used in the em-

bedded graph; and recent ideas developed by Goodrich and collaborators¹ that uses the complete bipartite (biclique) minor of the Chimera graph as a virtual hardware (as shown in Fig. 2). The full embedding of a problem graph in the hardware graph is found by first embedding into a chosen virtual hardware or one of its minors. Our choice of a bipartite virtual representation for the hardware is motivated in part by the simplicity of the structure as well as its balance between size and order of the virtual representation, and additionally for its connection to associative memory recall and other variants of machine learning applications [20]. We note that alternative virtual representations are equally valid, e.g., a square grid.

Klymko et al.[11] established tight bounds on the largest complete graph that can be minor embedded in a Chimera graph as well as demonstrating methods for embedding into faulty hardware graphs and introduced the concept of “maximal minor” embedding. We rename the set of “maximal minors” as the *minor set cover* (MSC): this is the set of minors for a given graph \mathcal{G} where any subgraph or minor of \mathcal{G} will either be a member of the MSC or is a subgraph contained in one of the members. The MSC of the biclique virtual hardware is a finite set of embeddable graph minors which can be precomputed without reference to the input problem, and can act as a lookup table. This reframes the problem of graph minor embedding as a subgraph isomorphism search: an input graph is compared against each key, if it is found to be contained in a given key, the problem graph is now embedded into a member of the MSC, and then an embedding in the hardware is found. It has been proposed previously that reducing the problem complexity of graph embedding from graph homomorphism to subgraph isomorphism can lead to substantial speedup in processing time [21, 22].

The question of whether a graph is minor-embeddable has been explored through many different approaches, such as: forbidden minors and extremal graph theory. Forbidden minors are minors which a class of graphs is known to exclude [23]: Wagner’s theorem established that planar graphs cannot contain K_5 or $K_{3,3}$ minors [24], while the works of Robertson and Seymour [25, 26, 27, 28, 29] develop the theory of forbidden minors for planar and non-planar graphs (see also the review in Ref. [30]). Extremal graph theory identifies what classes of graphs, or graph qualities, ensure certain minors are contained by a graph [31, 32, 33, 34, 35]. We will show that the final minor in the MSC of a complete bipartite graph is always the K_{N+1} graph, and in establishing the robustness of this minor for more general bipartite graphs, we turn to research on the development of theorems for the existence of complete minors: [36, 37, 38, 39, 40]. However the class of bipartite graphs under consideration are not particularly sparse, nor are they random, of large order, size, girth or degree. In focusing on the K_{N+1} minor, we are searching for conditions that ensures a graph has the largest possible complete minor.

¹ Work presented at the Society for Industrial and Applied Mathematics Workshop on Network Science 2016, manuscript in progress.

Currently our method for constructing the MSC of a graph is only applied to the case of complete or near-complete bipartite graphs and in this paper we focus on covering non-planar bipartite graphs which are undirected and contain no multiple edges or self-loops. This excludes several cases of bipartite graphs which have a trivial MSC that only contains the original graph. For example, simple paths, cycles, and the star graph S_n . We also identify leaves (terminal vertices) as graph edges which do not contribute to the formation of a member of the MSC.

We derive the necessary requirements to build an edge set which generates a MSC beginning with the simplest case of the complete (fully connected) bipartite graph (Sec. 4). A complete bipartite graph $K_{N,N}$ which has minimum partition order N has a MSC which contains the complete graph K_{N+1} . This is the largest complete graph that can be embedded into the original graph as no other set of edge contractions can lead to a minor with a completely disconnected graph complement. For the Chimera(n, n, c) hardware graph, the MSC identifies K_{nc+1} as the largest complete graph which can be minor embedded into the virtual hardware $K_{nc,nc}$. This agrees with results found by Choi [7] and results given in [11] based on treewidth arguments.

Incomplete bipartite graphs are those graphs which are missing edges between partitions. We discuss the case of a complete bipartite graph missing a small number of edges in Sec. 5, and focus on the K_{N+1} minor robustness on a general incomplete bipartite graph. Criteria are derived which identify bipartite graphs of minimum partition order N which lack the minor K_{N+1} .

2 Definitions and notations

A graph $\mathcal{G} = \mathcal{G}(V, E)$ is defined by a vertex set $\{v_i\} \equiv V$ and an edge set $\{e_{ij}\} \equiv E$. In this paper we only consider simple, undirected graphs: the edges have no orientation (the edge $e_{ij} = (x_i, x_j)$ is equivalent to the edge $e_{ji} = (x_j, x_i)$), and multiple edges and self-loops are not allowed. When counting the degree of a vertex set resulting from the contraction of an edge e_{ij} , the *in-degree* of vertex x_i counts all edges which connect to x_i but excludes the edge e_{ij} , and the *out-degree* of vertex x_j counts all edges which connect to x_j but excludes the edge e_{ij} .

For any graph \mathcal{G} on n vertices, the complement graph \mathcal{G}^c is also defined on the vertex set $V(\mathcal{G})$ and contains an edge \bar{e}_{ij} only if \bar{e}_{ij} does not exist on \mathcal{G} . Thus the union of \mathcal{G} and its complement \mathcal{G}^c form the complete graph on $V(\mathcal{G})$. For example, the complete graph $\mathcal{G} = K_n$ has a complement of order $|V(\mathcal{G}^c)| = |V(\mathcal{G})| = n$ vertices but $|E(\mathcal{G}^c)| = 0$ edges.

A graph \mathcal{G}' is *minor embeddable* in \mathcal{G} if for each vertex v of a graph \mathcal{G}' a mapping $\phi_{\mathcal{G}}(v)$ can be found which takes v to a connected subtree of \mathcal{G} . The individual vertex sets of $\phi_{\mathcal{G}}(v)$ do not overlap, and $\phi_{\mathcal{G}}(x), \phi_{\mathcal{G}}(y)$ are adjacent if there exists vertices $x_i \in \phi_{\mathcal{G}}(x)$ and $y_i \in \phi_{\mathcal{G}}(y)$ which are adjacent on \mathcal{G} . The order of a given subtree is the number of vertices it contains. An isomorphic

embedding maps each vertex v to a single vertex of \mathcal{G} (i.e. $\phi_{\mathcal{G}}(v) = v$), i.e. each vertex is mapped to a subtree of order 1.

Generating a minor (\mathcal{M}) from a given graph \mathcal{G} is done by edge contraction or edge removal, and \mathcal{G} may have a large number of minors. For finite graphs, there exists a set of graph minors $\mathcal{M} \equiv \{\mathcal{M}^{(i)}\}$ which we define as the MSC. The minors $\mathcal{M}^{(i)} \in \mathcal{M}$ are unique in that they are not isomorphically embeddable in the original graph, nor are they contained in any other minor as a subgraph. As a result, the set \mathcal{M} covers the entire set of minors (i.e. for any minor m of \mathcal{G} , m is either a member of the MSC or is contained in a minor of the set cover as a subgraph). By definition any minor formed by edge deletion cannot be included in the MSC. When searching for possible members of the MSC only minors formed by edge contraction are considered.

This work focuses on the MSC for bipartite graphs ($K_{N,N'}$). A bipartite graph is a graph with a vertex set which can be partitioned into two non-overlapping subsets: $v_a \subset V(\mathcal{G}), v_b \subset V(\mathcal{G}), v_a \cup v_b = V(\mathcal{G}), v_a \cap v_b = \emptyset$. Complete bipartite graphs $K_{N,N'}$ are those graphs of size $|E(K_{N,N'})| = NN'$, with all edges existing between vertices in different partitions. Incomplete bipartite graphs $\tilde{K}_{N,N'}$ are graphs with missing edges between partitions, $|E(\tilde{K}_{N,N'})| < NN'$. On a complete bipartite graph, all vertices in a partition have the same degree, which is equal to the order of the other partition. For the case of incomplete bipartite graphs, we define vertices which are not fully connected to the opposite partition as *incomplete vertices*.

A subset of the edge set of a graph, is called an *edge matching* if all edges are non-adjacent (do not share a vertex). A perfect matching is an edge set which leaves no vertex of the graph uncoupled. The size of the perfect matching set for a complete bipartite graph $K_{N,N}$ is N and is an upper bound for the size of the perfect matching on an incomplete bipartite graph $\tilde{K}_{N,N}$. For a complete bipartite graph with unequal partition orders, $K_{N,N'}$, the size of the perfect matching set is $\min(N, N')$.

3 Minor embedding in the ideal Chimera hardware graph

The quantum annealer from D-Wave Systems, Inc. uses a lattice of coupled superconducting flux qubits. The topology for the connections and interactions between the qubits is represented by a hardware graph referred to as the Chimera(n, m, c) graph [41, 3, 42, 43]. This graph has a fixed topology: it is an $n \times m$ square lattice of $K_{c,c}$ unit cells with intercell connections column-wise between left bipartite partitions and row-wise between right bipartite partitions (see Chimera(3,3,4) in Fig. 1). The Chimera(n, m, c) is bipartite, and the largest biclique is the unit cell $K_{c,c}$. Few problems can be isomorphically embedded into the D-Wave processor. In general, optimizing a given logical Hamiltonian requires the use of minor embedding into the hardware graph, which creates a significant bottleneck in the quantum annealing workflow.

Rather than enumerate the entire MSC of the Chimera graph, we use an intermediate embedding step to construct a virtual hardware graph of the

original hardware graph. We assume the Chimera graph is ideal, with all qubits operational (no hard faults). Contracting all intercell connections on a Chimera(n, m, c) hardware results in a virtual hardware² which is a complete bipartite graph $K_{nc,mc}$ (see Fig. 1). For this virtual hardware, we then construct the MSC. The remainder of this paper studies the construction of the MSC and establishes that the largest complete graph embeddable in a $K_{nc,mc}$ virtual hardware is $K_{\min(n,m)c+1}$. These results agree with lemmas presented in Ref. [11] which showed the treewidths of $K_{c,c}$ and K_c coincide, $\tau(K_{c,c}) = \tau(K_{c+1}) = c$. The approach of Klymko et al. was to use the maximal

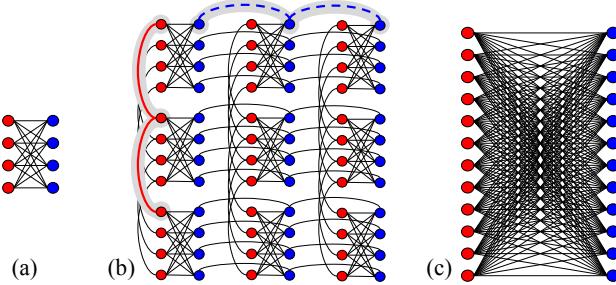


Fig. 1 [Color online] The Chimera graph and virtual hardware construction: (a) the unit cell of a Chimera graph is the $K_{4,4}$ graph, (b) the 3×3 grid of unit cells for Chimera(3,3,4), vertex bags are defined along each row or column of intercell connections (highlighted in grey), (c) contraction along all edges in a bag results in the $K_{12,12}$ virtual hardware. The left partition consists of vertex bags with 3 physical qubits, formed by contracting vertical intercell connections (red), the right partition consists of vertex bags with 3 physical qubits, formed by contracting all horizontal intercell connections (blue, dashed)

minor set of the Chimera unit cell to define an iterative embedding process. In this work we consider a two-step embedding procedure which first generates a complete bipartite virtual hardware then identifies the MSC of the virtual hardware. An embeddable complete graph K_n is embedded into the virtual hardware through an isomorphic mapping into (at least) one of the minors in the minor set.

For an ideal Chimera(n, n, c) the two-step embedding and the Klymko embedding identify the largest embeddable complete graph as K_{nc+1} . Generalizing to a Chimera(n, m, c) hardware, the largest embeddable graph is $K_{\min(n,m)c+1}$. Using the two-step embedding results in a final embedding for $K_{\min(n,m)c+1}$ which embeds each logical qubit into a chain of length m, n or length $m + n$. With the MSC construction, the embedding of K_{nc+1} into a Chimera hardware graph with $n \times n$ square grid of $K_{c,c}$ unit cells will require

² See footnote 1

2 chains of order n and $cn - 1$ chains of order $2n$. The distribution of chain lengths for the embedding of K_{13} on a 3×3 square Chimera is: 2 chains of order $\ell = m = 3$ and 11 chains of order $\ell = 2m = 6$. Comparison to the embeddings shown in Ref. [11], where the maximal minor set of the Chimera unit cell was used to define an iterative embedding process, shows that the two-step embedding finds the same embeddings for K_{13} and K_{17} as the iterative minor extension embeddings.

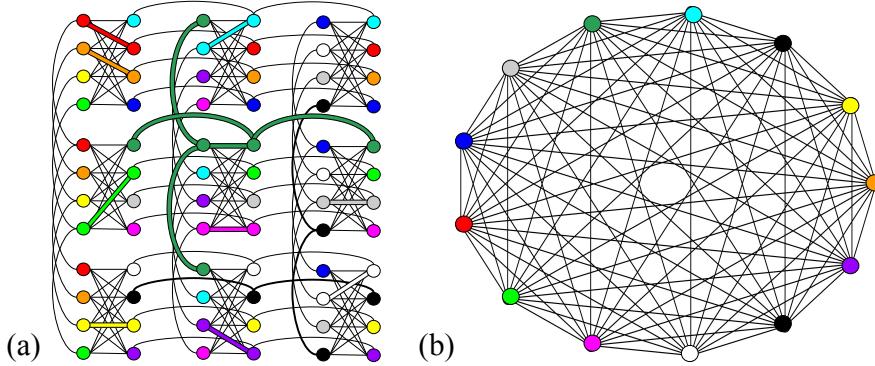


Fig. 2 [Color online] Example of embedding K_{13} in the Chimera(3,3,4) virtual hardware construction: (a) the embedding of a logical qubit into 6 physical qubits is highlighted (dark green), (b) of the 13 vertices on K_{13} , 11 are chains of order 6 (various colors) while 2 are chains of order 3 (black)

In Ref. [7], Choi gave a lower bound of $\lceil \frac{n-3}{d-2} \rceil$ on the minimum number of qubits needed to embed each logical qubit, and a lower bound on the total number of qubits needed to embed a graph of order n : $\Omega(n^2/d)$ qubits, where $d = c + 2$ is the number of couplings per physical qubit. Our embedding on Chimera($n, n, 4$) for K_{4n+1} , by embedding each logical qubit into either n or $2n$ physical qubits, saturates the lower bound $\lceil \frac{nc-3}{d-2} \rceil = \lceil \frac{4n-3}{4} \rceil = n$. The ideal Chimera(12, 12, 4) hardware contains 1152 qubits, while the minimum number of qubits needed to embed K_{49} is 400 qubits. In this work we show that by embedding into the $K_{48,48}$ virtual hardware, the largest embeddable clique is K_{49} .

The remainder of this section is centered around proving the following theorem:

Theorem 1 *On the ideal Chimera(n, m, c) hardware ($n \times m$ grid of $K_{c,c}$ unit cells) the largest complete graph which can be embedded through the two-step process is $K_{d_{min}c+1}$, where $d_{min} = \min(n, m)$.*

Proof Through the construction of a bipartite virtual hardware, each physical qubit is contained in a virtual qubit, which is a chain of length n . It will be shown in the construction of the MSC that each logical qubit eventually is

embedded into a single virtual qubit (final embedding into a subtree of order n) or at most a pair of virtual qubits (final embedding into a subtree of order $2n$).

Lemma 1 *On the ideal Chimera(n, n, c) hardware, embedding the complete graph K_{nc+1} will have $nc - 1$ subtrees of order $2n$ and 2 subtrees of order n .*

The dimensions of the quantum hardware can be used to minimize the order of the vertex subtrees. A graph K_{N+1} can be embedded in Chimera(n', n', c') hardware if $n'c' \geq N$ but the order of each subtree is only dependent on n' . For example, K_{41} can be embedded in a Chimera(10, 10, 4) hardware with maximum embedding subtree of order 20, or can be embedded in Chimera(2, 2, 20) hardware with maximum embedding subtree of order of 4.

4 MSC of complete bipartite graphs

The MSC of a complete bipartite graph $K_{N,N}$ contains exactly N minors, and creates a graph sequence which converges to a complete graph K'_N , ($N' < 2N$). The edges which one must contract in order to form minors of the MSC are a set of $(N - 1)$ non-adjacent edges. We prove our construction of the MSC, and the identification of the set of edges to contract in terms of an edge matching set and the graph complement of the complete bipartite graph.

Based on our definition of the MSC as the minors which covers all possible minors which can be constructed from a graph, we expect each set member to be the densest connected graph on N' vertices. As each MSC minor is denser than the previous minor, the corresponding sequence of graph complements will become sparsely connected. The final minor in the set cover of a complete bipartite graph $K_{c,c}$ is found to be the complete graph K_{c+1} whose complement graph is of order $(c + 1)$ but size 0.

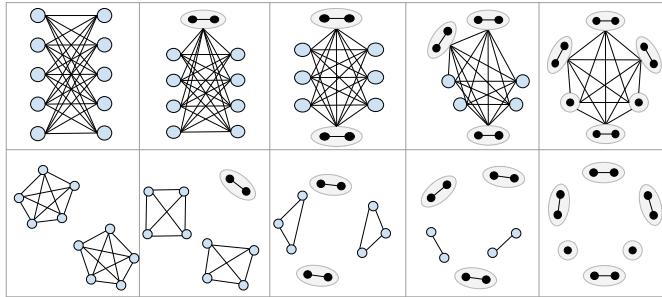


Fig. 3 Top row: the construction of the MSC for a complete bipartite graph $K_{5,5}$. Bottom row: the evolution of the complement for each minor. The final MSC minor is the complete graph K_6 which has a corresponding complement graph which has an empty edge set

Theorem 2 *For a complete bipartite graph $\mathcal{G} = K_{c,c}$, contracting a single edge will result in a minor $\mathcal{M}_{\mathcal{G}}^{(1)}$ which is a member of the MSC.*

Proof By definition, a bipartite graph does not have connections between vertices contained in the same vertex partition. Contracting any edge on a complete bipartite graph will result in a vertex bag which is connected to both bipartite partitions. Consider an edge on the bipartite graph e_{ij} connecting vertex v_i in the left partition and vertex v_j in the right partition. We also define the sets of $(N-1)$ neighboring vertices $\{v'_i\}, \{v'_j\}$. All vertices in $\{v'_i\}$ are in the right partition, and vice versa for $\{v'_j\}$. The contraction of e_{ij} creates the vertex bag $\phi(x_{ij})$ which is fully connected: having connections to all vertices in $\{v'_i\} \cup \{v'_j\}$. Consequently, such a minor cannot exist as a subgraph of the original graph because of its degree $2(N-1)$ and its connections between both partitions.

It is also seen that the contraction of an edge on the original bipartite graph and the creation of a minor with a fully connected vertex results in a disconnected vertex in the minor's complement. Next we consider the action of contracting two edges on a graph. First, considering the case of edges which are non-adjacent on the original graph (do not share any vertices).

Corollary 1 *The contraction of a pair of non-adjacent edges (e_1, e_2) on a complete bipartite graph $K_{c,c}$ will result in a MSC minor.*

Proof The contraction of the first edge e_1 will result in a MSC minor $\mathcal{M}_{\mathcal{G}}^{(1)}$ by creating a fully connected vertex. Contracting the second edge e_2 will also create a fully connected vertex. As a result, the minor created by contracting two non-adjacent edges $\mathcal{M}_{\mathcal{G}}^{(2)}$ is identified as a member of the MSC: it cannot be contained in $\mathcal{M}_{\mathcal{G}}^{(1)}$ as a subgraph due to the additional fully connected vertex, nor is it possible for it to be a subgraph of $\mathcal{M}^{(0)}$.

When non-adjacent edges are contracted we see that another disconnected vertex exists on the minor's complement. This effect is not observed for the minor formed when two adjacent edges are chosen on the original graph and then contracted.

Corollary 2 *The contraction of a pair of adjacent edges on a complete bipartite graph will not result in a MSC minor. Contracting adjacent edges will result in a graph minor which is a subgraph of the MSC minor generated by contracting non-adjacent edges. Likewise the complement of such a minor will contain the MSC minor complement as a subgraph.*

Proof Consider two sets edges on a complete bipartite graph. Set 1 contains a pair of non-adjacent edges ($S_1 = \{(v_1^a \leftrightarrow v_1^b), (v_2^a \leftrightarrow v_2^b)\}$); the edges connect four distinct vertices on the graph with 2 vertices in each bipartite partition. Set 2 contains a pair of adjacent edges; they connect 3 distinct vertices on the graph ($S_2 = \{(v_1^a \leftrightarrow v_1^b), (v_2^a \leftrightarrow v_1^b)\}$). The first contraction in either set will

add connections between vertices in the same partition: $v_1^a \leftrightarrow v_2^a, v_3^a, \dots, v_c^a$, $v_1^b \leftrightarrow v_2^b, v_3^b, \dots, v_c^b$. Contracting a non-adjacent edge adds the (unique) connections: $v_2^a \leftrightarrow v_3^a, \dots, v_c^a$, $v_2^b \leftrightarrow v_3^b, \dots, v_c^b$, but contracting an adjacent edge only adds the connections: $v_2^a \leftrightarrow v_3^a, \dots, v_c^a$.

The results for non-adjacent edges are extended further, establishing that there is a finite size edge matching which will generate all minors of the MSC for a complete bipartite graph.

Theorem 3 *For a complete bipartite graph $K_{N,N}$ the cardinality of the MSC is equal to n and is formed by the contraction of a set of $(n-1)$ non-adjacent edges where n is the size of the maximal edge matching set.*

Proof By our definition, a MSC minor of a graph cannot be contained within any other minor as a subgraph. Any graph has at least one minor in its MSC, $\mathcal{M}^{(0)} = \mathcal{G}$ the original graph. All remaining minors in the MSC must be non-isomorphic to each other, any other minor, or the original graph.

From Theorem (2), we established that contraction of a single edge on a bipartite graph creates the first minor $\mathcal{M}^{(1)}$ of order $2N-1$, with 1 vertex set of degree $2N-2$ and the remaining vertices with degree N . It was shown in Corollary (1) that a pair of non-adjacent edge contractions will create a second minor $\mathcal{M}^{(2)}$ of order $2N-2$ with 2 vertex set of degree $2N-3$ and the remaining vertices with degree N .

This argument is extended to postulate that N non-adjacent edges could create N minors in the MSC where the value of N is determined by enforcing the condition that all minors in the MSC are non-isomorphic. As edges are contracted, the order of each subsequent minor is reduced by 1, while the number of fully connected vertices is increased by 1. After contracting $(N-1)$ edges the minor $\mathcal{M}^{(N-1)}$ is of order $N+1$ and has $(N-1)$ vertex sets of degree $2N-N=N$ and 2 vertices of degree N (the complete graph K_{N+1}). While additional non-adjacent edges may exist, any further contractions will result in minors which can be contained in the $\mathcal{M}^{(N-1)}$ minor. For the complete bipartite graph, $n=N$ the size of the maximal edge matching set.

From choosing $(N-1)$ edges to contract and including the first minor defined by the original graph, it follows that any complete bipartite graph $K_{N,N}$ will have a MSC of cardinality N .

Corollary 3 *The complete bipartite graph $K_{1,1}$ has MSC of cardinality 1.*

Corollary 4 *A complete bipartite graph which is also a star graph $K_{N,1} = \mathcal{S}_N$ has MSC of cardinality 1.*

From the above theorems we present our final result for complete graphs.

Theorem 4 *The largest clique minor of a $K_{N,N}$ graph is K_{N+1} .*

Proof The original bipartite graph $\mathcal{G} = K_{N,N}$ has $2N$ total vertices of degree $d = N$. The MSC of such a graph is the ordered sequence formed by contracting

$(N - 1)$ non-adjacent edges: $\{\mathcal{M}^{(0)}, \mathcal{M}^{(1)}, \mathcal{M}^{(2)}, \dots, \mathcal{M}^{(N-1)}\}$, where $\mathcal{M}^{(0)} = \mathcal{G}$. The order of minor $\mathcal{M}^{(i)}$ is determined by the previous minor:

$$|V(\mathcal{M}^{(i)})| = |V(\mathcal{M}^{(i-1)})| - 1 = |V(\mathcal{M}^{(0)})| - (i). \quad (1)$$

The number of vertices of degree N is reduced by 2 on each minor while the number of fully connected vertices is increased by 1. As a result the final MSC minor $\mathcal{M}^{(N-1)}$ is of order $2N - (N - 1) = N + 1$ and N -degree regular, it is the graph K_{N+1} . By definition the graph K_{N+2} cannot exist or it would be in the MSC and the graph K_N is contained within K_{N+1} as a subgraph.

Corollary 5 *The largest clique minor of a $K_{N,N'}$ graph is $K_{\min(N,N')+1}$.*

Corollary 6 *The clique number of each minor is strictly increasing over the MSC of a complete bipartite graphs.*

Proof The first minor in the MSC is the bipartite graph, which has a clique number of 2. As non-adjacent edges are contracted, each subsequent minor has an increasing number of vertices which are connected to both bipartite partitions. As a result the order of the largest clique in each minor increases by 1 vertex and the clique number increases by 1. The maximum clique number of $K_{N,N}$ is $N + 1$ and is reached on the final minor in the MSC.

Theorem 5 *The treewidth of any minor in the MSC of a complete bipartite graph $K_{N,N}$ is N .*

Proof The proof follows from a simple treewidth argument. Each minor of the $K_{N,N}$ MSC is formed sequentially, so each minor in the MSC is a minor of the previous member. By definition, a graph \mathcal{H} is a minor of \mathcal{G} if the treewidth of \mathcal{H} is bounded from above by the treewidth of \mathcal{G} : $\text{tw}(\mathcal{H}) \leq \text{tw}(\mathcal{G})$. The treewidth of the first minor, a complete bipartite graph $K_{N,N}$, is known $\text{tw}(K_{N,N}) = N$. The treewidth of the final minor K_{N+1} is known $\text{tw}(K_{N+1}) = N$ and thus each minor in the remaining sequence must have treewidth $N = \text{tw}(K_{N,N}) \geq \text{tw}(\mathcal{M}^{(i)}) \geq \text{tw}(K_{N+1}) = N$.

The construction of the MSC for a complete bipartite graph $K_{N,N}$ can be implemented using a simple greedy algorithm. To add an edge to the matching set, one is chosen at random from the existing edges of the graph. The first edge is chosen from all edges of the graph. To ensure subsequently chosen edges are not adjacent to any edge which already exists in the set $E_{\mathcal{M}}$, after adding an edge to the minor cover edge set, all edges adjacent the head and tail of an edge e_{ij} are removed from the graph. This process is repeated, with edges added to the matching set until a stopping condition is met. Once all edges in the non-adjacent edge set are contracted, and vertex sets formed, the remaining vertices of the original graph are isomorphically mapped to vertex sets of size 1.

As stated in Corollary 3, the $K_{1,1}$ graph minor has $|\mathcal{M}| = 1$. This is used to define the stopping condition: once the removal of the head/tail vertices of a contracted edge results in a subgraph which is $K_{1,1}$ the procedure ends. It is seen that this procedure constructs a set of only $(N - 1)$ non-adjacent edges, from which all possible members of the MSC are formed.

5 MSC of nearly complete bipartite graphs

Most bipartite graphs encountered in real world applications are not complete, it is likely that connections are absent between the partitions. For example, a quantum processor may have faulty (inoperable) qubits, and thus the Chimera(n, m, c) hardware graph may be missing vertices. In this section we look at the MSC for the class of incomplete bipartite graphs $\tilde{K}_{N,N}$, created by removing a sparse subset of edges from a complete bipartite graph. The approach outlined in Sec. 4 may have limited applicability for such graphs. Randomly choosing connections on a graph may not result in a set of $(N - 1)$ non-adjacent edges; collisions between the missing edge sets and the non-adjacent edge set are probable and the completion of the non-adjacent edge set may require the addition of an edge which does not exist on the graph. Additionally, even if a set of $(N - 1)$ non-adjacent edges is found, there is no guarantee the contractions will yield the clique minor K_{N+1} . Since random bipartite graphs, and incomplete bipartite graphs are very common in quantum annealing (discussed in Sec. 3) this last point is very important to investigate.

The MSC of $K_{N,N}$ establishes that the largest embeddable clique is K_{N+1} . On an incomplete bipartite graph, the edges which contract and form members of the MSC cannot be chosen completely at random. For $\tilde{K}_{5,5}$ missing a single edge, we show in Fig. 4 it still contains the clique minor K_6 if the non-adjacent edge set is chosen appropriately. However this result is not guaranteed for any arbitrary non-adjacent edge set (as shown in Fig. 5). The robustness of the K_{N+1} minor as edges are removed from $K_{N,N}$ is of importance in quantum annealing applications (discussed in Sec. 3) where hard faults can dramatically affect the connectivity of the hardware graph.

We show how the results in Sec. 4 can be modified and applied to the simple case of $K_{N,N}$ missing a single edge. From there we define three conditions which identify graphs $\tilde{K}_{N,N}$ that cannot have a clique minor of order $N + 1$. Only graphs with equal partition orders are discussed, but the results can be extended to $\tilde{K}_{N,N'}$. However, a full discussion of how to mitigate the effects of faults is left as an open question.

Most notably there exist a class of edges which will never assist in the creation of a MSC minor. These are identified as those edges connecting to leaves (also known as terminal vertices).

Theorem 6 *Contraction of an edge which connects to a terminal vertex will never yield a minor in the MSC.*

Proof In our definition of the MSC, we noted that any member of the MSC cannot be contained in another member as a subgraph. Consider a graph \mathcal{G} with a leaf vertex. The edge connecting to the leaf has a head with in-degree $(d_i - 1)$ while the tail has out-degree 0. Contraction of this edge will result in a vertex set $\phi(x)$ with degree $(d_i - 1)$ and the resulting minor will be a subgraph of \mathcal{G} .

However, the K_{N+1} clique minor can only be generated after $(N - 1)$ edge contractions for either a complete bipartite graph $K_{N,N}$ or incomplete bipar-

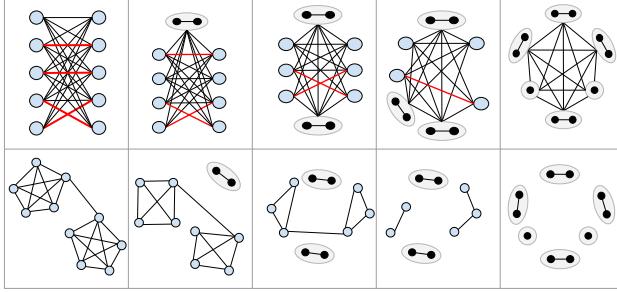


Fig. 4 [Color online] Top row: evolution of a minor set generated by a non-adjacent edge set for $\tilde{K}_{5,5}$ (missing a single edge) which covers the set of incomplete vertices (heavy red lines). Bottom row: evolution of the complement for each minor

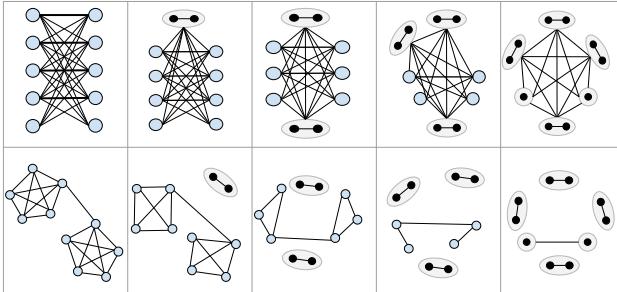


Fig. 5 Top row: evolution of a minor set generated by an arbitrary non-adjacent edge set (heavy black lines) for $\tilde{K}_{5,5}$ (missing a single edge) . Bottom row: evolution of the complement for each minor

ite graph $\tilde{K}_{N,N}$. The robustness of the K_{nc+1} graph minor is given by the following result:

Theorem 7 *On the faulty Chimera(n, n, c) hardware, embedding the complete graph K_{nc+1} is not possible if the virtual hardware is missing more than $m' = (nc)(nc+1)/2 + (nc-1)$ edges and the set of n' incomplete vertices cannot be covered by $(nc-1)$ edges.*

Our proof of the clique minor robustness is framed in terms of the complement graph. For a complete bipartite graph $K_{N,N}$, its complement graph will consist of two disconnected complete graphs K_N . As $K_{N,N} \rightarrow \tilde{K}_{N,N}$ through the removal of edges while maintaining partition orders, the complement graph will gain edges which connect the two disconnected K_N graphs.

The first condition for a faulty Chimera graph is that it's bipartite virtual hardware cannot be missing more than $(N+1)N/2 + (c-1)$ edges. This is a resource argument, that the original graph must have a sufficient number of edges to contract and still support the complete graph K_{N+1} . The second

requirement is that the set of incomplete vertices on the bipartite graph must be coverable by at most $(N - 1)$ edges. If an incomplete vertex is not contained in a larger vertex embedding, then the resulting minor will not have the maximum degree possible. The third requirement is that throughout the sequence of edge contractions, a disconnected dimer consisting of two vertex embeddings joined by an edge must not be created in the minor complement. If such a dimer exists it cannot be resolved into two disconnected vertex sets.

Example 1 Consider the bipartite graph formed by removing a single edge from $K_{2,2}$. The clique minor K_3 cannot be formed from the resulting incomplete bipartite graph $\tilde{K}_{2,2}$ because the original graph has 3 edges, any contraction would result in a minor with 3 vertices and 2 edges.

Example 2 The crown graph is an incomplete bipartite graph which is formed from $K_{N,N}$ by removing the N edges of a perfect matching and is an example of a graph which does not contain a K_{N+1} minor. There are $2N$ incomplete vertices, this set cannot be covered by any combination of $(N - 1)$ edges. The lack of a K_{N+1} minor is further verified by treewidth argument (see [44],[45]): a graph \mathcal{G} contains \mathcal{H} as a minor if the treewidth of \mathcal{H} is bounded above by the treewidth of \mathcal{G} , $\text{tw}(\mathcal{H}) \leq \text{tw}(\mathcal{G})$. The crown graph has treewidth $N - 1$ and thus cannot embed K_{N+1} ($\text{tw}(K_{N+1}) = N$).

6 Open problems

We close with a brief discussion of two open problems: the more full treatment of a faulty hardware graph and the full implementation of MSC embedding.

Hard faults (inoperable qubits) can occur in a quantum annealer, which result in missing vertices and edges on the hardware graph. There are many approaches to mitigating the effects of hard faults: one can choose to consider only the largest $n' \times n'$ square portion of the hardware graph which lacks any faults (see [11]) or distort the shape of vertex bags to accommodate faults (see [12]). In our preliminary discussion, we consider simply removing any hard faults prior to constructing a virtual hardware. The resulting virtual hardware of a Chimera(n, n, c) with faults will remain a bipartite graph, however it will be missing edges, the partition orders may not be equal, nor equal to nc , and its associated MSC may be larger than that of $K_{nc,nc}$. The robustness of the K_{N+1} clique minor was discussed in Sec. 5 for graphs missing few edges and the construction of edge matchings needs careful consideration to maximize the order of embeddable clique minors.

While we have identified the MSC for a virtual representation of the Chimera lattice, we have not discussed methods for solving the subsequent subgraph isomorphism problem. The latter step is necessary for choosing which member(s) of the MSC contains the input graph as a subgraph. Subgraph isomorphism is believed to be NP-complete and we may expect that solving this problem also poses a significant computational task. However, the instance of interest is strictly smaller than the original minor embedding problem. Since

members of the MSC are known beforehand, unique properties of those graphs (which may be used to easily reject potential matches) can be precomputed. These properties include: order, size, degree distribution, clique number, and treewidth. Moreover, recent work from Babai [46] has shown that graph isomorphism may be computable in quasipolynomial time, a result that would have a profound implications. We defer a more detailed analysis of this step to a subsequent publication.

7 Conclusions

Minor embedding of the logical graph describing an input Hamiltonian presents a significant bottleneck in adiabatic quantum programming. Our aim in this work has been to reduce the difficulty of finding an embedding for a known input graph by exploring what graphs can be embedded into a complete bipartite virtual hardware. By defining the MSC, we identify K_{nc+1} as the largest clique which is minor embeddable into the $K_{nc,nc}$ virtual hardware of the Chimera(n, n, c).

We have developed a general method for constructing the MSC of a fully connected bipartite graph $K_{N,N}$. It was seen that the contraction of edges belonging to a set of $(N-1)$ non-adjacent edges constructs all members of the MSC. This edge set could be found using a simple greedy algorithm and the method is also applicable to complete bipartite graphs $K_{N,N'}$ with $N \neq N'$.

For the case of an incomplete bipartite graph, the simple greedy algorithm is of limited use as the number of minors in the MSC can be very large. We focus on the largest clique minor, and determine two criteria that identify graphs which does not have a K_{N+1} clique minor: first, any incomplete bipartite graph which does not meet a minimum size $|E| < (N+1)^2 + (N-1)$ and second, any incomplete bipartite graph on which the set of incomplete vertices cannot be covered by $(N-1)$ edges. These two criteria are enough to determine that a graph does not have a K_{N+1} clique minor, but they are insufficient to determine if a graph does have a K_{N+1} minor.

By identifying the MSC of an ideal Chimera hardware, we have a solution to the problem of complete graph embedding. If the graph K_{nc+1} can be embedded, then any graph of order $n' < nc + 1$ can be embedded, but the actual vertex map is not known. Determining the robustness of the K_{nc+1} clique minor on a Chimera hardware with faulty qubits is still an open ended question.

8 Acknowledgements

This work was supported by the United States Department of Defense and used resources of the Computational Research and Development Programs at Oak Ridge National Laboratory. This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S.

Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan.

References

1. Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science*, 292(5516):472–475, 2001.
2. William M. Kaminsky and Seth Lloyd. Scalable architecture for adiabatic quantum computing of NP-hard problems. In *Quantum computing and quantum bits in mesoscopic systems*, pages 229–236. Springer, 2004.
3. M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198, 05 2011.
4. Travis S. Humble, Alex J. McCaskey, Ryan S. Bennink, Jay Jay Billings, EF. D’Azevedo, Blair D. Sullivan, Christine F. Klymko, and Hadayat Seddiqui. An integrated programming and development environment for adiabatic quantum optimization. *Computational Science & Discovery*, 7(1):015006, 2014.
5. Andrew Lucas. Ising formulations of many NP problems. *Frontiers in Physics*, 2:5, 2014.
6. Vicky Choi. Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quantum Information Processing*, 7(5):193–209, 2008.
7. Vicky Choi. Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design. *Quantum Information Processing*, 10(3):343–353, 2011.
8. Alexander Grigoriev and Hans L. Bodlaender. Algorithms for graphs embeddable with few crossings per edge. *Algorithmica*, 49(1):1–11, 2007.
9. Michael D. Coury. Embedding graphs into the extended grid. *arXiv preprint cs/0703001*, 2007.
10. Fedor V. Fomin, Bart M. P. Jansen, and Michał Pilipczuk. Preprocessing subgraph and minor problems: When does a small vertex cover help? *Journal of Computer and System Sciences*, 80(2):468–495, 2014.

11. Christine Klymko, Blair D. Sullivan, and Travis S. Humble. Adiabatic quantum programming: minor embedding with hard faults. *Quantum information processing*, 13(3):709–729, 2014.
12. Arman Zaribafyan, Dominic J. J. Marchand, and Seyed Saeed Changiz Rezaei. Systematic and deterministic graph-minor embedding for cartesian products of graphs. *arXiv preprint arXiv:1602.04274*, 2016.
13. Jun Cai, William G. Macready, and Aidan Roy. A practical heuristic for finding graph minors. *arXiv preprint arXiv:1406.2741*, 2014.
14. Tomas Boothby, Andrew D. King, and Aidan Roy. Fast clique minor generation in chimera qubit connectivity graphs. *Quantum Information Processing*, 15(1):495–508, 2016.
15. Travis S. Humble, Alexander McCaskey, Jonathon Schrock, Keith Britt, Hadayat Seddqi, and Neena Imam. Performance models for split-execution computing systems. 18th Workshop on Advances in Parallel and Distributed Computational Models, 2016.
16. Davide Venturelli, Dominic J. J. Marchand, and Galo Rojo. Quantum annealing implementation of job-shop scheduling. *arXiv preprint arXiv:1506.08479*, 2015.
17. Eleanor G. Rieffel, Davide Venturelli, Bryan O’Gorman, Minh B. Do, Elicia M. Prystay, and Vadim N. Smelyanskiy. A case study in programming a quantum annealer for hard operational planning problems. *Quantum Information Processing*, 14(1):1–36, 2015.
18. Alejandro Perdomo-Ortiz, Joseph Fluegemann, Sriram Narasimhan, Rupak Biswas, and Vadim N. Smelyanskiy. A quantum annealing approach for fault detection and diagnosis of graph-based systems. *The European Physical Journal Special Topics*, 224(1):131–148, 2015.
19. Keith A. Britt and Travis S. Humble. High-performance Computing with Quantum Processing Units. *ACM Journal on Emerging Technologies in Computing*, in press, 2016.
20. Hadayat Seddqi and Travis S. Humble. Adiabatic quantum optimization for associative memory recall. *Frontiers in Physics*, 2:79, 2014.
21. Pavol Hell and Jaroslav Nešetřil. On the complexity of h-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92 – 110, 1990.
22. Marek Cygan, Fedor V. Fomin, Alexander Golovnev, Alexander S. Kulikov, Ivan Mihajlin, Jakub Pachocki, and Arkadiusz Socała. Tight lower bounds on graph embedding problems. *arXiv preprint arXiv:1602.05016*, 2016.
23. Martín Cera, Ana Diánez, Pedro García-Vázquez, and Juan Carlos Valenzuela. Graphs without minor complete subgraphs. *Discrete Mathematics*, 307(11):1276–1284, 2007.
24. K. Wagner. Über eine eigenschaft der ebenen komplexe. *Mathematische Annalen*, 114(1):570–590, 1937.
25. Neil Robertson and P. D. Seymour. Graph minors. I. Excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39 – 61, 1983.
26. Neil Robertson and P. D. Seymour. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92 – 114, 1986.

27. Neil Robertson and P. D. Seymour. Graph minors. VIII. A Kuratowski theorem for general surfaces. *Journal of Combinatorial Theory, Series B*, 48(2):255 – 288, 1990.
28. Neil Robertson and P. D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43 – 76, 2003.
29. Neil Robertson and P. D. Seymour. Graph minors. XX. Wagner’s conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325 – 357, 2004. Special Issue Dedicated to Professor W.T. Tutte.
30. Ken-ichi Kawarabayashi and Bojan Mohar. Some recent progress and applications in graph minor theory. *Graphs and Combinatorics*, 23(1):1–46, 2007.
31. Zi-Xia Song and Robin Thomas. The extremal function for K9 minors. *Journal of Combinatorial Theory, Series B*, 96(2):240–252, 2006.
32. Nikolaos Fountoulakis, Daniela Kühn, and Deryk Osthus. The order of the largest complete minor in a random graph. *Random Structures & Algorithms*, 33(2):127–141, 2008.
33. Nikolaos Fountoulakis, Daniela Kühn, and Deryk Osthus. Minors in random regular graphs. *Random Structures & Algorithms*, 35(4):444–463, 2009.
34. Gwenaël Joret and David R Wood. Complete graph minors and the graph minor structure theorem. *Journal of Combinatorial Theory, Series B*, 103(1):61–74, 2013.
35. David Eppstein. Grid minors in damaged grids. *arXiv preprint arXiv:1303.1136*, 2013.
36. A. V. Kostochka. Lower bound of the Hadwiger number of graphs by their average degree. *Combinatorica*, 4(4):307–316, 1984.
37. Andrew Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory, Series B*, 81(2):318–338, 2001.
38. Reinhard Diestel and Christof Rempel. Dense minors in graphs of large girth. *Combinatorica*, 25(1):111–116, 2004.
39. Andrew Thomason. Disjoint unions of complete minors. *Discrete Mathematics*, 308(19):4370–4377, 2008.
40. Thomas Böhme, Ken-ichi Kawarabayashi, John Maharry, and Bojan Mohar. Linear connectivity forces large complete bipartite minors. *Journal of Combinatorial Theory, Series B*, 99(3):557–582, 2009.
41. D-wave publications. <http://www.dwavesys.com/resources/publications>. Accessed: 2016-05-12.
42. N. G. Dickson, M. W. Johnson, M. H. Amin, R. Harris, F. Altomare, A. J. Berkley, P. Bunyk, J. Cai, E. M. Chapple, P. Chavez, F. Cioata, T. Cirip, P. deBuen, M. Drew-Brook, C. Enderud, S. Gildert, F. Hamze, J. P. Hilton, E. Hoskinson, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Lanting, T. Mahon, R. Neufeld, T. Oh, I. Perminov, C. Petroff, A. Przybysz, C. Rich, P. Spear, A. Tcaciu, M. C. Thom, E. Tolkacheva, S. Uchaikin, J. Wang, A. B. Wilson, Z. Merali, and G. Rose. Thermally assisted quantum annealing of a 16-qubit problem. *Nat Commun*, 4:1903,

05 2013.

43. Helmut G. Katzgraber, Firas Hamze, and Ruben S. Andrist. Glassy chimeras could be blind to quantum speedup: Designing better benchmarks for quantum annealing machines. *Phys. Rev. X*, 4:021008, Apr 2014.
44. Hans L. Bodlaender. A tourist guide through treewidth. *Acta cybernetica*, 11(1-2):1, 1994.
45. Fedor V. Fomin and Dimitrios M. Thilikos. Dominating sets and local treewidth. In Giuseppe Di Battista and Uri Zwick, editors, *Algorithms - ESA 2003: 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003. Proceedings*, pages 221–229. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
46. László Babai. Graph isomorphism in quasipolynomial time. *arXiv preprint arXiv:1512.03547*, 2015.