Theoretical Computer Science 989 (2024) 114369

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Minor embedding in broken chimera and derived graphs is NP-complete [☆]

Elisabeth Lobe [a,*], Annette Lutz [b]

[a] Institute for Software Technology, German Aerospace Center (DLR), Lilienthalplatz 7, 38108, Brunswick, Germany
[b] Institute for Software Technology, German Aerospace Center (DLR), Linder Höhe, 51147, Cologne, Germany

ARTICLE INFO

Communicated by M. Hirvensalo

Keywords:
Graph minor embedding
Hamiltonian cycle problem
Quantum annealing
Chimera graph
Pegasus graph
Zephyr graph

ABSTRACT

The embedding is an essential step when calculating on the D-Wave machine. In this work, we show the hardness of the embedding problem for all types of currently existing hardware, represented by the Chimera and the derived Pegasus and Zephyr graphs, containing unavailable qubits. We construct certain broken Chimera graphs, where it is hard to find a Hamiltonian cycle. As the Hamiltonian cycle problem is a special case of the embedding problem, this proves the general complexity result for the Chimera graphs. By exploiting the subgraph relation between the Chimera and the derived graphs, the proof is then further extended to the Pegasus graphs and to the Zephyr graphs.

1. Introduction

1.1. Background

Quantum annealing is an emergent technology mainly driven by the developments of the company D-Wave systems. Their machines attract a growing number of users from diverse fields by promising to solve their optimization problems faster than classical computers could ever do. The problem is encoded in a system of quantum bits (*qubits*) with adjustable degrees of pairwise interactions [1]. By slowly evolving the quantum system several times, the quantum annealers provide a sample of solutions in each run with objective values close to the optimum [2].

The D-Wave machines optimize a so-called *Ising model* with a specific structure, which was shown to be NP-complete [3]. Thus, in theory, a huge variety of problems can be transferred to them in polynomial time [4]. However, several transformation steps are required to bring the actual problem into the specific Ising format. In particular, the limited size of the hardware and the restricted qubit connectivity need to be overcome. We are not aware of any application of practical interest matching the so-called *Chimera*, *Pegasus* or *Zephyr* graphs directly, which represent the original, the currently operating and the recently announced new hardware structure of the D-Wave machines, respectively. See, e.g., [5–8] for some example applications.

This requires to solve the so-called *embedding* problem, where a set of qubits shall be found for every original vertex to represent it in the hardware. By applying a strong coupling to the edges between the qubits in such a set, they can act as a single logical vertex

[☆] This article belongs to Section C: Theory of natural computing, Edited by Lila Kari.
[*] Corresponding author.
E-mail address: elisabeth.lobe@dlr.de (E. Lobe).

https://doi.org/10.1016/j.tcs.2023.114369
Received 28 October 2021; Received in revised form 24 November 2023; Accepted 20 December 2023

Available online 28 December 2023

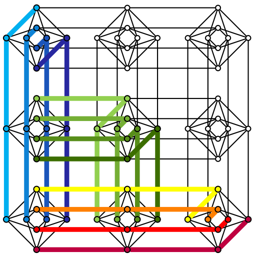


Fig. 1. Standard triangular complete graph embedding in the ideal Chimera graph where the logical vertices of the complete graph are represented by different chains [11]. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

and the union of all neighbors of these qubits yields the desired connectivity [9]. We give a more precise definition of an embedding and its relation to the term *minor* in Section 2.1.

The embedding problem has a straightforward solution for certain well-structured graphs transferred into the ideal Chimera graph. For example in Fig. 1, the standard embedding of the complete graph with 12 vertices is shown, derived from the TRIAD layout introduced in [9]. It can easily be extended for larger Chimera graphs, where the number of embeddable vertices grows linearly with the size of the Chimera graph. This however means, that the number of vertices used in the hardware graph grows quadratically. In an ideal Chimera graph in the size of the currently operating `DW_2000Q` solver of D-Wave, we can therefore embed only up to 65 completely connected vertices, although it contains 2048 qubits [10].

Further easily embeddable graphs can be found for instance in [12]. In general, those graphs follow the underlying grid structure of the Chimera. For arbitrary graphs without an obvious relation to such a structure, being too large to simply use the complete graph scheme but small enough that they might fit, we always need to find a suitable embedding before we are able to calculate on the machine. This still holds for the new hardware architectures with the Pegasus or the Zephyr graphs, although they yield higher connectivities [13,14]. Both types of graphs have a strong relation to the Chimera graph due to the underlying superconducting loop structure and therefore can be termed as *derived* graphs.

Above we use the term *ideal* as it refers to the graph provided from the built-in hardware structure of overlapping superconducting loops forming the qubits, which is for instance described in [13]. However, some qubits, or in rare cases also couplings, are switched off by the programming interface of the currently operating hardware, because they do not show the expected behavior [15]. As a single such qubit can already break the standard embedding schemes, it is not clear anymore how to find an embedding for the complete and other well structured graphs, not to speak of arbitrary graphs. Furthermore, after each calibration of the machine, the location of the inaccessible qubits changes.

Considering the ongoing development of the quantum annealing devices, we expect the number of qubits to grow further. But it seems unlikely that a fully connected graph structure can soon be realized due to physical restrictions. At the same time, we assume to see a decreasing number of broken qubits compared to the total number, however, they will not vanish completely. For calculations on D-Wave's annealing machines, the embedding problem will therefore stay relevant in the long term.

1.2. Related work

In general, for two arbitrary graphs G and H, it is NP-hard to decide whether G can be embedded in H. We go into more detail about this in Section 2.3. In their extensive research around such graph relations, Robertson and Seymour could show among other things that there exists a polynomial algorithm to decide this question when G is fixed to a single graph and only H is part of the input [16]. As it includes the knowledge about so-called forbidden minors of G, this result hides constants depending exponentially on the size of G. A concrete algorithm following Robertson and Seymour's idea is given for hardware graphs with fixed branchwidth in [17], which is further improved by [18], and for planar hardware graphs in [19]. Although the bounds were improved, the dependence on G is still exponential and thus the algorithm practically infeasible for larger graphs.

The embedding problem in the quantum annealing context deals with a different case as G is an arbitrary graph derived from the specific application while H is fixed to be in the class of all Chimera, Pegasus or Zephyr graphs with broken vertices. Even though the hardware graphs are very well structured, no comparable results to Robertson and Seymour's are known in that case. In the contrary, we show in this work that the problem remains hard. This is stated more formally in Section 2.3 by Theorem 11.

This means in general, the embedding problem is as hard as the actual problem that shall be solved on the D-Wave machine. This has some implications in practice. Usually, several possibilities exist to encode an arbitrary optimization problem as an Ising model, e.g. by introducing more vertices, the connectivity between them could be reduced. However, in view of our hardness result, we cannot expect to identify an easy to recognize set of graph properties which the chosen formulation should fulfill to be better embeddable than others or to be embeddable at all in the current hardware graph.

Heuristics, as for example presented in [20], are intended to embed arbitrary graphs 'on-the-fly'. Although they work well in practice at the moment, especially for sparse input graphs, with growing hardware sizes, they will most likely not be able to scale as well. To overcome such a possible bottleneck, the idea of precalculated templates was already introduced in [21]. Virtual hardware graphs form an intermediate layer between the actual hardware and the problem graph and shift the expensive computation away from the user. For the most general template, the complete graph, an algorithm was found in [11] whose runtime is exponential only in the number of broken vertices. Other templates are for instance discussed in [22].

Our construction to prove the stated complexity result was mainly inspired by the ideas of [23]. There, Itai et al. showed that the Hamiltonian Cycle Problem for grid graphs is NP-complete. As the Hamiltonian Cycle Problem reduces to the embedding problem, we adopt several of their definitions and results for the grid graphs and, in the first place, transfer them to the Chimera graphs. For this, we introduce the basic concepts in Section 2. Afterwards we show the construction of specific Chimera graphs in Section 3, for which we establish several results about Hamiltonian paths and cycles in Section 4. With this, we can conclude the proof of the NP-completeness of the Hamiltonian Cycle Problem and thus of the embedding problem for Chimera graphs. The extension to the derived Pegasus and Zephyr graphs is then shown in Section 5. Finally, we give a brief outlook about further research directions in Section 6.

2. Preliminaries

In this section, we introduce the main terms and notations and provide a collection of existing concepts, recaptured and matched up to give a full picture of the overall proof. We start with minors and the related embeddings. After introducing the specific hardware graph of the D-Wave machine, the Chimera graph, we formulate the main question and reduction steps. Our construction is based on the ideas of [23], whose basic grid definitions are provided at the end of this section.

In our construction we always refer to simple undirected graphs. For the basic graph definitions, we generally follow [24]. For shortness, we identify the tuple $(x, y) \in X \times Y$ for two sets X, Y with the non-commutative product xy. We will also use vw to abbreviate the subset $\{v, w\} \subseteq X$, although v and w commute. However, it is clear from the context whether reverting the product ordering is feasible. In general, we use xy for integer coordinates in the two-dimensional space, while vw refers to an edge of a graph.

2.1. Minors and embeddings

Graph minors are a basic concept in graph theory. Please see [24,25] for more details. In the following, we introduce graph minors and their relation to the embedding as well as some hardness results in a more descriptive way. Although several different definitions for graph minors exist, most of them are equivalent. We use the following one from [24]:

Definition 1. Given two arbitrary graphs G and H, G is called a *minor* of H if G is isomorphic to a graph that can be formed from H by a series of the following operations: edge contraction, edge or vertex deletion.

The term ‘edge contraction’ means here that the endpoints of a single edge are replaced by a single new vertex whose neighbors are formed by the union of the neighbors of these endpoints. By this, no multiple edges or loops appear and we always keep the resulting graphs simple. Furthermore, note that, if we reasonably assume that no vertex is deleted that is formed by an edge contraction, the ordering of the operations is arbitrary. It is also easy to see that the minor relation is transitive, meaning a minor of a minor of some graph H is itself a minor of H.

Several results concerning minor relations exist. The most straightforward question is a classical problem in graph theory:

Definition 2 *(*Minor Containment Problem*).* Given two arbitrary graphs G and H, does H contain G as a minor?

It is a generalization of the Subgraph Homeomorphism Problem mentioned in [26]. We recall the analogously applicable proof for the following hardness result as we use the same argumentation later on:

Lemma 3. *The* Minor Containment Problem *is NP-complete.*

Proof. The problem is in NP since the edge contractions and edge and vertex deletions that need to be applied to H provide a polynomial certificate. Moreover, it includes the NP-complete Hamiltonian Cycle Problem as a special case by G being a cycle graph with the same number of vertices as H. □

Although our paper mainly works with the above minor definition, here we briefly want to connect it with a concept that is widely used in the quantum annealing context. There graph minors usually appear in terms of an *embedding*: To overcome the very restricted hardware architecture and simulate an arbitrary problem connectivity, several hardware vertices need to be combined to form a logical vertex. This is, e.g., described in [9]. In [25] we find the more formal definition, where 2^X denotes the set of all subsets of a set X:

Definition 4. For two graphs G and H, an *embedding* of G in H is a map $\varphi : V(G) \to 2^{V(H)}$ fulfilling the following properties

a) all $\varphi(v)$ for $v \in V(G)$ induce disjoint connected subgraphs in H,
b) for all edges $vw \in E(G)$, there exists an edge in H connecting the sets $\varphi(v)$ and $\varphi(w)$.

We call G *embeddable* into H if such an embedding function for G and H exists.

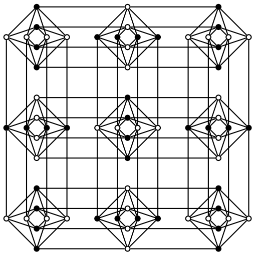


Fig. 2. Bipartition in the Chimera graph $C_{(3,3)}$.

With this, we can formulate an analogous question to above:

Definition 5 *(*Embedding Problem*).* Given two arbitrary graphs G and H, is G embeddable into H?

We can easily see that Minor Containment Problem and the Embedding Problem are equivalent in the following sense:

Lemma 6. *Given two arbitrary graphs G and H, G is a minor of H if and only if G is embeddable into H.*

Proof. We start with the implication, if G is embeddable into H, it is a minor of H. Given the map φ, the graph G can be obtained from H by successively contracting all edges in $H[\varphi(v)]$, labeling the resulting vertex with v for all $v \in V(G)$ and deleting all surplus vertices and edges that are not in $V(G)$ and $E(G)$, respectively.

Reversely, if G is a minor of H, we can always find an embedding: All vertices concerned by edge contractions that finally form a single vertex v of G are included in its embedding $\varphi(v)$, while all deleted vertices and edges can be ignored. □

Due to this relation, also the term *minor embedding* is used and we refer to both problems equivalently as Minor Embedding Problem in the following.

2.2. Chimera graphs

In this section, we introduce the *Chimera graph* representing the connectivity of D-Wave's quantum annealing hardware, which was first described in [27].

Definition 7. Let C_{cr} be the Chimera graph with the number of columns $c \in \mathbb{N}$ and the number of rows $r \in \mathbb{N}$ of unit cells of complete bipartite subgraphs with four vertices in each partition. The $K_{(4,4)}$-unit cells are arranged in a grid pattern and connected as shown in Fig. 2. In accordance with the figure, we also define an *orientation* of the vertices, where we call a vertex *horizontal*, if it lies in the horizontal partition of a unit cell, thus is connected vertically to vertices in other unit cells above and/or below, *vertical* otherwise.

Let C_∞ be the Chimera graph with an infinite number of unit cells in all four directions. Meaning, we also allow negative row and column unit cell coordinates. Thus, there is no boundary and all vertices have a degree of 6. We call this graph *infinite Chimera graph* in the following.

The Chimera graph is bipartite itself as it can be seen by the vertex coloring in Fig. 2. Wherever necessary, we use the terms *ideal* or *non-broken* Chimera graphs for C_{rc} or C_∞ to clearly distinguish them from the following:

Definition 8. We call a Chimera graph *broken* if it is a vertex-induced subgraph of an ideal Chimera graph, meaning $C_\infty[W]$ for some $W \subseteq V\left(C_\infty\right)$. Let $\mathcal{C}$ denote the set of all finite broken Chimera graphs, thus derived from $C_{rc} \subset C_\infty$ with $c, r < \infty$.

Remark. By additionally allowing edges to be broken, the construction in Section 3 still works and could even be simplified while it would require less broken vertices. However, in currently operating hardware, the probability of a broken edge is pretty small. To address the more restricted case, we focus on vertex-induced subgraphs in this work. If the hardness result holds for this subset it also holds for all subgraphs of hardware graphs.

We use a *depth*, the size of the partitions in a unit cell, of $d = 4$ according to existing hardware. However, our approach can be extended to arbitrary $d \geq 4$, because a Chimera graph with depth d_1 is a vertex-induced subgraph of all Chimera graphs with depth $d_2 \geq d_1$. Thus, it is contained in the corresponding set of broken Chimera graphs.

2.3. Reduction of the Hamiltonian cycle problem

With the definitions of the former sections, we can now formulate the overall question this paper is dealing with:

Definition 9 *(*BROKEN CHIMERA MINOR EMBEDDING PROBLEM*).* Given an arbitrary graph G and some $C \in \mathcal{C}$, is G a minor of C, i.e., is G embeddable into H?

In this section, we show how the BROKEN CHIMERA MINOR EMBEDDING PROBLEM reduces to the HAMILTONIAN CYCLE PROBLEM to adopt the known complexity results for this problem. Note that until now there is no result about the complexity status of the related question whether a given graph is actually itself a broken Chimera graph, meaning more formally

Definition 10 *(*BROKEN CHIMERA RECOGNITION PROBLEM*).* Given an arbitrary graph G, do we have $G \in \mathcal{C}$?

This means to check whether G is a vertex-induced subgraph of an ideal Chimera graph. The number of rows and the number of columns of unit cells of a corresponding Chimera graph can be bounded by the number of vertices in G. Clearly, this problem is not harder than the arbitrary INDUCED SUBGRAPH ISOMORPHISM PROBLEM, thus is in NP, but it remains open, whether it is easier. We do not examine this problem further because, by the practical conditions, we know we are given a broken Chimera graph.

We proceed with the main complexity result, which we prove in the course of the paper:

Theorem 11. *The* BROKEN CHIMERA MINOR EMBEDDING PROBLEM *is NP-complete.*

As the BROKEN CHIMERA MINOR EMBEDDING PROBLEM is a special case of the MINOR EMBEDDING PROBLEM, it is in NP. But does the restriction to Chimera graphs as hardware graphs result in better solvability?

The HAMILTONIAN CYCLE PROBLEM was shown not to be hard for the ideal Chimera graph [12]. Thus, the argument in the proof of Lemma 3 does not apply for those graphs. However, the presented construction of a Hamiltonian cycle cannot be transferred easily if the Chimera graph is broken. In fact, we show in general

Lemma 12. *The* HAMILTONIAN CYCLE PROBLEM *for graphs $C \in \mathcal{C}$ is NP-complete.*

By this, Theorem 11 is proven straightforwardly with the same arguments as for the general MINOR EMBEDDING PROBLEM in the proof of Lemma 3. To prove in turn Lemma 12, we use a special set of graphs $\mathcal{B}$ where we already know that

Lemma 13 *([23] Lemma 2.1). The* HAMILTONIAN CYCLE PROBLEM *for graphs $B \in \mathcal{B}$ is NP-complete.*

Here, $\mathcal{B}$ is the set of all planar bipartite graphs with maximum degree 3. Note that two necessary conditions for the existence of a Hamiltonian cycle are an equal number of vertices in each partition and that no vertex has a degree of 1. Thus, in the following we further restrict $\mathcal{B}$ to such graphs.

In the following course of this article, we show how to construct a broken Chimera graph, denoted by $C(B)$, from a graph $B \in \mathcal{B}$ such that it fulfills the following two properties:

Lemma 14. *For all $B \in \mathcal{B}$, $C(B)$ can be constructed in polynomial time.*

Lemma 15. *For all $B \in \mathcal{B}$, $C(B)$ has a Hamiltonian cycle if and only if there exists a Hamiltonian cycle in B.*

This means that the HAMILTONIAN CYCLE PROBLEM is hard for the constructed broken Chimera graphs with a size polynomial in the size of the graph B and it thus is hard for $C \in \mathcal{C}$ in general, holding Lemma 12. We have summarized the reduction steps in Fig. 3, where the highlighted step is the one we prove here.

2.4. Basics of grid graphs

Itai et al. showed an analogous result to Lemma 12 for grid graphs in [23]. We adopt the proof to the Chimera graph and reuse some of their constructions. For this, we recall and rephrase some of their definitions and results about grid graphs in this section.

Definition 16. Let

$$G_\infty := \left(\mathbb{Z}^2, \left\{\{ab, xy\} \subset \mathbb{Z}^2 : |a-x| + |b-y| = 1\right\}\right)$$

be the *infinite (non-broken) grid graph* with vertices defined by integer *coordinates* in two-dimensional space. A *grid graph* is an arbitrary subgraph of the infinite grid graph.

A *(non-broken) rectangular (grid) graph* with one corner at coordinate $ab \in \mathbb{Z}^2$ and the opposite at $xy \in \mathbb{Z}^2$ with $xy \neq ab$ is a special grid graph and defined by

$$R^{ab}_{xy} := G_\infty\left[\left\{\min\{a,x\}, ..., \max\{a,x\}\right\} \times \left\{\min\{b,y\}, ..., \max\{b,y\}\right\}\right],$$

where we abbreviate $R_{cr} := R^{(1,1)}_{cr}$ for a rectangular graph of size $c \times r$ for a number of columns $c \in \mathbb{N}$ and rows $r \in \mathbb{N}$ placed at $(1,1)$.

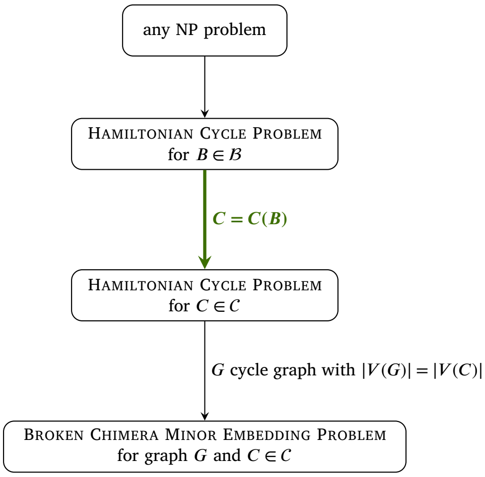

any NP problem
Hamiltonian Cycle Problem for $B \in \mathcal{B}$
$C = C(B)$
Hamiltonian Cycle Problem for $C \in \mathcal{C}$
G cycle graph with $|V(G)| = |V(C)|$
Broken Chimera Minor Embedding Problem for graph G and $C \in \mathcal{C}$

Fig. 3. Polynomial reduction steps of the proof.

Remarks.

- With the grid vertices ab, xy, ay, xb as corners of the strip, we have $R^{ab}_{xy} = R^{xy}_{ab} = R^{xb}_{ay} = R^{ay}_{xb}$.
- R^{ab}_{xy} is of size $(|a-x|+1) \times (|b-y|+1)$ and, with $c = |a-x|+1$ and $r = |b-y|+1$, we have $R^{ab}_{xy} \cong R_{cr}$.

Analogously to the broken Chimera graph, we call a rectangular graph *broken* if it is a vertex-induced subgraph of a rectangular graph, the graphs [23] is dealing with. A *rectangular subgraph* denotes an arbitrary subgraph of a rectangular graph, thus a grid graph that can be bounded by a rectangular shape. We further need the following special rectangular graphs:

Definition 17. A *grid strip* is a rectangular graph R^{ab}_{xy} for $ab, xy \in \mathbb{Z}^2$ with

a) $|a-x| = 1$ and $|b-y| \geq 1$ or
b) $|b-y| = 1$ and $|a-x| \geq 1$,

thus a vertex width of 2 in one of the dimensions and arbitrary in the other. To the latter we refer as the *length* of the strip. We say a strip is *vertically oriented* in case a) and *horizontally oriented* in case b). In short, we use the term *vertical*, respectively, *horizontal strip*.

Remark. Only $R_{(2,2)}$ is both vertically and horizontally oriented.

The main concepts presented here are still taken from [23], but especially the following is defined slightly differently:

Definition 18. A *grid tentacle* is the union of a series of alternately oriented grid strips $(S_i)_{i=1,\dots,n} = \left(R^{a_ib_i}_{x_iy_i}\right)_{i=1,\dots,n}$, where we have for all $i = 1, \dots, n-1$:

a) $V(S_i) \cap V(S_{i+1}) = \{a_{i+1}b_{i+1}, a_{i+1}y_{i+1}\}, V(S_i) \cap V(S_{i+1}) \cap \{a_iy_i, x_iy_i\} \neq \emptyset$ or
b) $V(S_i) \cap V(S_{i+1}) = \{a_{i+1}b_{i+1}, x_{i+1}b_{i+1}\}, V(S_i) \cap V(S_{i+1}) \cap \{x_ib_i, x_iy_i\} \neq \emptyset$.

This means, in a tentacle, we always attach the two ‘starting points’ of the next strip to the side of the strip before, by which always one of the ‘end points’ of the latter is reused. We have the first option of the above disjunction when a vertical grid strip is followed by a horizontal one and the second option for the reverse case. This is shown in Fig. 4.

Definition 19. Let the *parity* of a grid vertex $xy \in \mathbb{Z}^2$ given by: If we have $x + y \equiv 0 \mod 2$ we call the vertex *even*, otherwise *odd*. Let the vertex sets be correspondingly denoted by the disjoint sets $V^{\text{even}}(G_\infty)$ and $V^{\text{odd}}(G_\infty)$.

It is easy to see that the even and odd vertices define a bipartition of the grid graph and with this also of grid strips and tentacles. By calling one arbitrary partition of $B \in \mathcal{B}$ even and the other odd, thus $V(B) = V^{\text{even}}(B) \cup V^{\text{odd}}(B)$, we can define analogously to [23]:

Definition 20. A *parity-preserving embedding* of a graph $B \in \mathcal{B}$ in the grid graph is a injective function $\psi : V(B) \to V(G_\infty)$, where

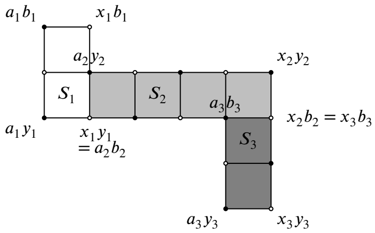
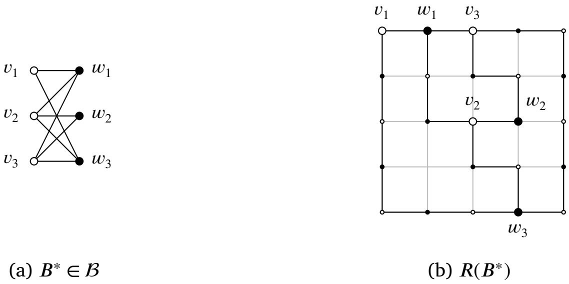


Fig. 4. Grid tentacle example composed of three grid strips with even (white) and odd vertices (black).



(a) $B^* \in \mathcal{B}$

(b) $R(B^*)$

Fig. 5. Example graph and its rectangular representation from [23] with even (white) and odd vertices (black).

a) the parities of v and $\psi(v)$ are the same for all $v \in V(B)$,
b) for all $vw \in E(B)$, there exists a path from $\psi(v)$ to $\psi(w)$ in G_∞ such that the inner vertices of this path do not intersect with any other full path.

Note that this is a different type of embedding than in Definition 4 because it defines a one-to-one mapping of the vertices from the problem graph, here B, and the hardware graph, the infinite grid, while the representation of edges is generalized to paths. Although both embedding concepts are related, it is not necessary for our construction to go into details about this here.

The above definition would also be valid for arbitrary bipartite graphs, but we are especially interested in graphs $B \in \mathcal{B}$, as in [23] the authors could show that the size of the grid graph needed for a parity-preserving embedding can be bounded:

Lemma 21 *([23] Lemma 2.2). For $B \in \mathcal{B}$, we can construct in polynomial time a parity-preserving embedding of B in a rectangular graph R_{nn} with $n = k\,|B|$ (for some constant $k \in \mathbb{N}$).*

For a fixed $B \in \mathcal{B}$ let $\psi : V(B) \to V(R_{nn})$ be such an embedding, which we assume to be fixed in the following, too. Let $R(B) \subseteq R_{nn}$ denote the grid graph resulting from this embedding, where we have $\psi(V(B)) \subseteq V(R(B))$. We call $R(B)$ the *rectangular representation* of B. Note that $R(B)$ is a rectangular subgraph but not a broken rectangular graph in our sense.

An adopted example from [23] is shown in Fig. 5. Although there might be a much simpler rectangular representation, on the one hand, it is not guaranteed to be found by the last lemma and, on the other hand, the shown example includes several different cases, which we handle in the following sections. Thus, we reuse the same example throughout the whole article.

3. Broken chimera graph construction

In this section, we show the construction of $C(B)$ for an arbitrary $B \in \mathcal{B}$. To ensure that the existence of a Hamiltonian cycle is mutually induced between the original graph B and the constructed broken Chimera graph $C(B)$, we use a very restricted version of the broken Chimera graph. Although this might not be necessary for the hardness of the problem, meaning most likely we could reduce the number of broken vertices, we want to keep the Hamiltonian cycle construction mostly unambiguous.

We start by giving an overview about the overall concept by evolving the underlying grid graph. Afterwards, we introduce the representations of the single elements, the vertices and edges. Finally, we combine the elements to the full broken Chimera graph corresponding to B.

3.1. Underlying rectangular subgraph

The rectangular representation $R(B)$ is the base line for our construction. However, it is not sufficient to be used directly. In this section, we therefore evolve the full underlying grid structure of the final Chimera graph. Later, the grid vertices are replaced with unit cells, where the coordinates of the grid vertices correspond to the coordinates, columns and rows, of the unit cells.

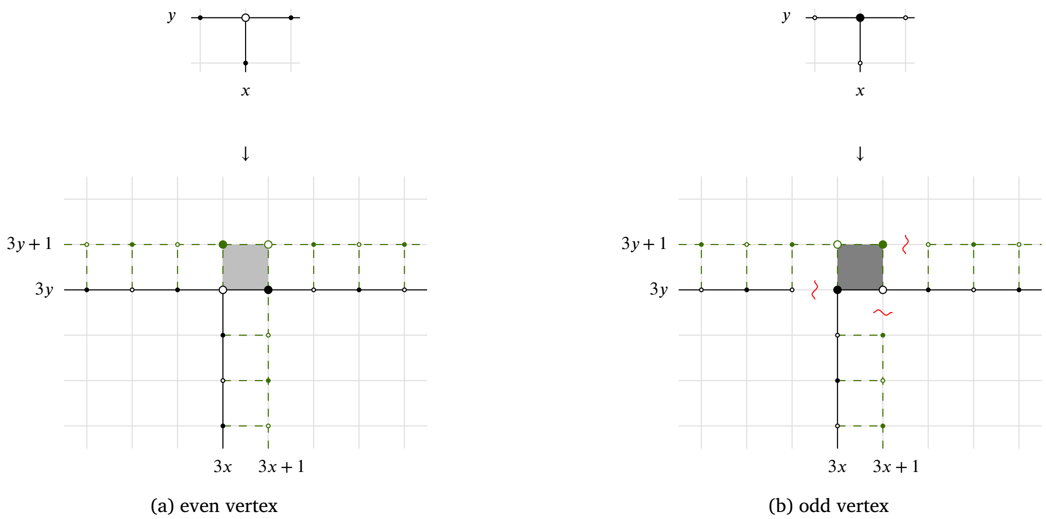


Fig. 6. Manipulation of grid vertices and incident edges of the rectangular representation to obtain the tripled rectangular representation (solid, black) with its extension (dashed, green online) illustrated in the lower part, where the tilde (red online) indicates missing edges for odd vertices.

Definition 22. Let $L(B)$ be the *enlarged rectangular representation* of B, a rectangular subgraph obtained from $R(B)$ by the following manipulations: We start with tripling the size of the rectangular representation $R(B)$ in each direction by replacing each edge by a straight path of length 3. This means that an original vertex v of B, represented in $R(B)$ by $xy = \psi(v)$ for a parity-preserving embedding ψ as found by Lemma 21, is now represented by the coordinate $3x\,3y$. Due to the odd factor 3, the parity of the vertices is preserved.

Afterwards, we extend the produced graph by adding the vertices $\tilde{x}(\tilde{y}+1)$, $(\tilde{x}+1)\tilde{y}$ and $(\tilde{x}+1)(\tilde{y}+1)$ and the edges $\{\tilde{x}\tilde{y}, \tilde{x}(\tilde{y}+1)\}$, $\{\tilde{x}\tilde{y}, (\tilde{x}+1)\tilde{y}\}$, $\{\tilde{x}(\tilde{y}+1), (\tilde{x}+1)(\tilde{y}+1)\}$ and $\{(\tilde{x}+1)\tilde{y}, (\tilde{x}+1)(\tilde{y}+1)\}$ (if not present already) for each vertex $\tilde{x}\tilde{y}$ in the tripled graph. This manipulation is shown in Fig. 6a. However, for the odd vertices $v \in V^{\text{odd}}(B)$ with $\psi(v) = xy$ we do not use the edges $\{(3x-1)\,3y, 3x\,3y\}$, $\{(3x+1)\,3y, (3x+1)(3y-1)\}$, $\{(3x+1)(3y+1), (3x+2)(3y+1)\}$ and $\{3x\,(3y+1), 3x\,(3y+2)\}$ (if they were added before at all). After removing them, we obtain the final enlarged rectangular subgraph for our construction $L(B)$. This is illustrated in Fig. 6b for a single odd vertex.

As it is shown exemplarily in Fig. 7, the resulting graph is now a combination of grid strips and tentacles connected by one or two edges with 'space in-between'. In the following sections, we describe the Chimera elements that replace the specific grid elements of $L(B)$. Thus, we specify:

Definition 23. Let $S_v := R^{(3x+1)(3y+1)}_{3x\,3y}$ denote the 2×2 grid strip subgraph that represents the original vertex $v \in V(B)$ in $L(B)$ for $\psi(v) = xy$. Let T_e be the grid tentacle between S_v and S_w representing edge $e = vw \in E(B)$ in $L(B)$. W.l.o.g. for v being even and w being odd, tentacle T_e is connected to S_v by two edges and to S_w by a single edge.

Concerning the overall graph, it is easy to see that

Corollary 24. *For all $B \in \mathcal{B}$, $L(B)$ can be constructed in polynomial time.*

Proof. Because the number of manipulations is bounded by the number of vertices and the number of edges of $R(B)$, the claim follows directly from Lemma 21. □

Remark. The enlarged rectangular representation $L(B)$ is a subgraph of a rectangular graph of size at most $(3n-1) \times (3n-1)$ for constant n from Lemma 21.

3.2. Vertices as broken chimera subgraphs

In this section, we explain the broken Chimera graphs that represent the single vertices. Remember each vertex $v \in V(B)$ is represented in $L(B)$ by the grid strip $S_v = R^{(3x+1)(3y+1)}_{3x\,3y}$ with $\psi(v) = xy$ for the parity-preserving embedding ψ. Now we replace the corresponding 2×2 grid vertices by a specific broken Chimera graph of size 2×2 as illustrated in black in Fig. 8.

We restrict our construction to graphs $B \in \mathcal{B}$, where each vertex has a degree of 2 or 3. No matter where the missing edge in the rectangular representation $R(B)$ is placed, the incident edges of a vertex with degree 3 form a T-like structure, which is possibly simply rotated. All vertices of degree 2 can be represented by just dropping one of the surplus connections.

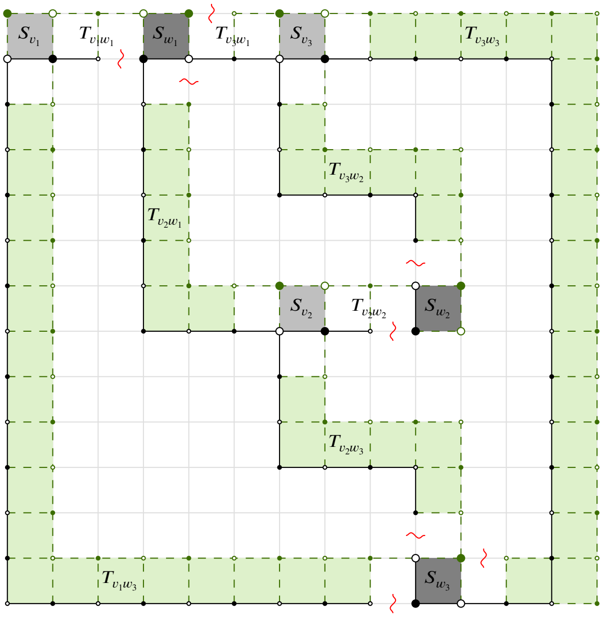


Fig. 7. Tripled (solid, black) and extended (dashed, green online) rectangular representation forming the enlarged rectangular representation $L(B^*)$ corresponding to the example of Fig. 5 with the sets of 2×2 vertices (underlaid in gray) corresponding to the original even (light) and odd vertices (dark) and the connecting tentacles (underlaid, light green online).

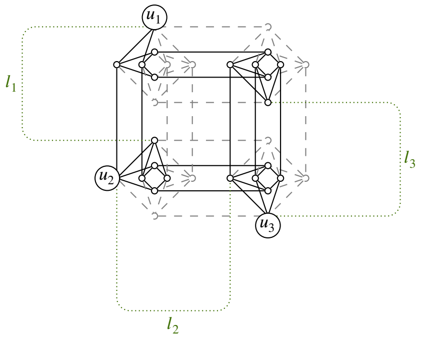


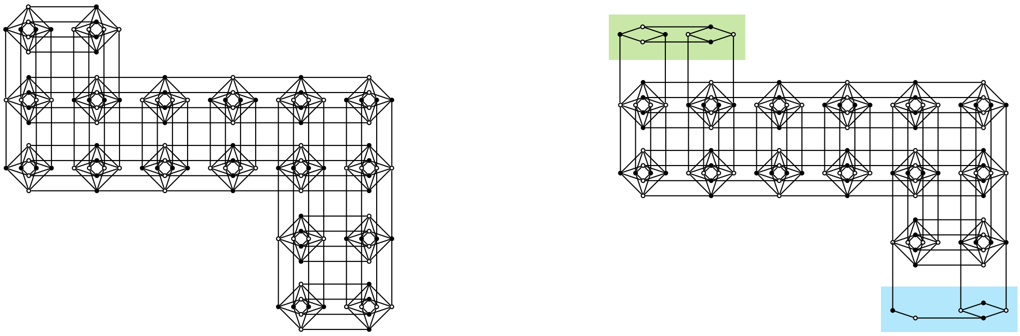
Fig. 8. Broken Chimera graph representing a grid vertex with broken vertices (dashed) and additional edges (dotted, green online) missing the edge to the north neighbor.

The same principle is kept for the representation of v in $L(B)$ and also in our broken Chimera graph: The dotted (green online) edges added in Fig. 8, l_1, l_2 and l_3, analogously form a ‘T’. Although those edges do not exist in a Chimera graph as shown, they illustrate the later connection to the Chimera elements representing the original edges. Thus, we define

Definition 25. Let $\tilde{C}_{(2,2)} \in C$ be the broken Chimera graph with the vertex set, and thus the connectivity, as shown in black in Fig. 8. For $v \in V(B)$, let $\tilde{C}(S_v) \subset C_\infty$ be the subgraph isomorphic to $\tilde{C}_{(2,2)}$, possibly rotated according to the representation of the incident edges in $R(B)$ and placed at coordinate $3x3y$ (with the unit cell in the lower left corner after rotation), that represents v.

As it can be seen in the definition, we use the same broken Chimera graph for all vertices in $V(B)$ regardless of their parity. The distinction of even and odd vertices only becomes apparent when connecting the vertex elements with the edge elements as already indicated by the construction of $L(B)$. We go into detail about the differences in the next sections.

However, note that we did not color the Chimera subgraph’s partitions in Fig. 8 by intention. Their assignment changes depending on the parity of the grid coordinates. Furthermore, the vertices u_1, u_2 and u_3 are labeled because they mark the main entry points of the Hamiltonian cycle constructed later on.



(a) unmodified version

(b) modified version with broken unit cells connecting to the even and the odd vertex representation (underlaid, on the top, green online, resp., on the bottom, blue online)

Fig. 9. Chimera tentacles corresponding to the example in Fig. 4.

3.3. Edges as chimera tentacles

In contrast to the vertices, the construction of the edges is more straightforward because nearly all vertices in the grid tentacles of $L(B)$ are replaced by non-broken Chimera unit cells. With the grid definitions of [23] from the previous section, we define analogously

Definition 26. For T being a grid tentacle, let the corresponding *Chimera tentacle* be

$$C(T) := C_\infty \left[\bigcup_{xy \in V(T)} V^{xy} \right],$$

where V^{xy} is the set of vertices in the unit cell at the coordinates $xy \in \mathbb{Z}^2$ in the infinite Chimera graph.

An example of a Chimera tentacle is illustrated in Fig. 9a. As a strip is also a tentacle with just one element, the same definition holds for *Chimera strips*.

As mentioned in the section before, the distinction between odd and even vertices is done by the way the tentacles are attached to the $C_{(2,2)}$-subgraphs representing the vertices. The necessity of this distinction becomes clear in the next section when we assemble the Hamiltonian cycle in the constructed Chimera graph. According to the enlarged rectangular representation $L(B)$, we restrict odd vertices to just a single edge in each direction, while even vertices get two edges. As we focus on vertex-induced subgraphs of the Chimera graph, this is realized by additional broken vertices in the joined tentacles, leading to the removal of edges. Thus, let us define slightly different Chimera tentacles to represent the edges:

Definition 27. For $e = vw \in E(B)$ and, w.l.o.g., v even and w odd, let $\tilde{C}(T_e)$ be the *modified Chimera tentacle* representing edge e, where the first two unit cells, connecting to S_v, are replaced by the element highlighted at the top (green online) in the example in Fig. 9b and the last two unit cells, connecting to S_w, by the one highlighted at the bottom (blue online).

Remarks.

- As we triple the rectangular representation, we ensure that we have a Chimera strip of length at least one between S_v and the next vertex or a possible corner in the tentacle in each direction for all vertices $v \in V(B)$. Thus, the above replacement is always possible in straight elements.
- If the tentacle T_e for some $e \in E(B)$ only consists of a single strip of length 1, we only use the part for odd vertices. Note that this case could be circumvented by scaling the underlying rectangular graph by a factor of 5 instead of 3. This however would also lead to larger Chimera graphs.
- By the definitions, $C(T)$ and $\tilde{C}(T)$ are broken Chimera graphs for all grid tentacles T.
- If T is finite, meaning the series of strips defining T only consists of a finite number of strips of finite lengths, $C(T)$ and $\tilde{C}(T)$ are finite and we have $C(T), \tilde{C}(T) \in \mathcal{C}$.

3.4. Composition of all elements

In this section, we show how the single elements are combined to form the broken Chimera graph. In Fig. 10, we show how the subgraph for the vertices and the tentacles are finally connected by an example of two neighboring vertices. Now we have all elements to combine the full broken Chimera graph. Thus, we can now formally define:

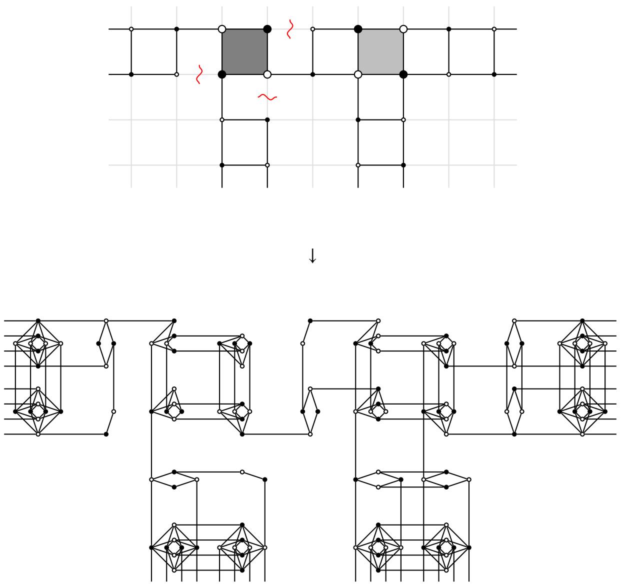


Fig. 10. Combined Chimera graph elements derived from $L(B)$ representing neighboring odd and even vertices in $C(B)$ with three incident edges each.

Definition 28. Let

$$C(B) := C_\infty\left[\bigcup_{v\in V(B)} V\left(\tilde{C}(S_v)\right) \cup \bigcup_{e\in E(B)} V\left(\tilde{C}(T_e)\right)\right]$$

be the broken Chimera graph corresponding to $B \in \mathcal{B}$.

In other words, $C(B)$ is the broken Chimera graph derived from $L(B)$ by replacing S_v with $\tilde{C}(S_v)$ for all $v \in V(B)$ and T_e with $\tilde{C}(T_e)$ for all $e \in E(B)$ and connecting those elements with the corresponding edges. By this, we can already conclude Lemma 14 with

Proof of Lemma 14. Analogously to Corollary 24, the replacements of grid vertices and paths of $L(B)$ by Chimera elements can be done in linear time. □

Remember, this also means that $C(B)$ is finite (if B is finite) and we have $C(B) \in \mathcal{C}$.

4. Hamiltonicity

In this section, we establish some results about Hamiltonian paths and cycles, first in the single elements and finally in the full constructed broken Chimera graph $C(B)$. After introducing the overall concept, we evaluate Chimera tentacles first as we build on previous results on Hamiltonian paths in rectangular Chimera graphs there. After analyzing the vertex elements, we can finally combine the full Hamiltonian cycle.

4.1. *Concept of return and cross paths*

While a Hamiltonian cycle visits every single vertex, just a subset of edges is covered. In our construction, we extend single edges to paths, meaning we add further vertices. Even if the edge is not used in a Hamiltonian cycle of B, those vertices need to be covered by a Hamiltonian cycle of $C(B)$.

To handle this issue in their construction for grid graphs, the authors of [23] introduced the concept of *cross* and *return paths*. The principle is illustrated in Fig. 11. While paths corresponding to edges used in the Hamiltonian cycle are simply passed over, thus form a cross path, the cycle is extended by loops to cover the vertices in the remaining paths. These loops are called return paths.

Clearly, the resulting cycle in the rectangular representation is not a Hamiltonian cycle anymore as some vertices are visited twice. However, we can use this as a base line for our Chimera graph construction where the grid vertices are replaced with unit cells. As unit cells can be crossed more than once without using a vertex twice, the loops form a valid extension of the cycle finally resulting in a Hamiltonian cycle.

Due to the parity-preserving embedding, every path in the rectangular representation still connects vertices of different parity. Thus, we can decide that all the return paths start in even grid vertices representing even original vertices. In this case, the loops

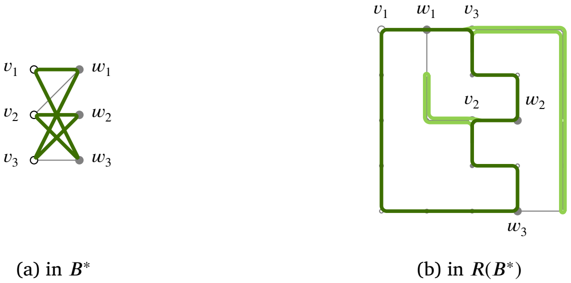


Fig. 11. Hamiltonian cycles in the example graph and its rectangular representation of Fig. 5 with additional loops (lighter).

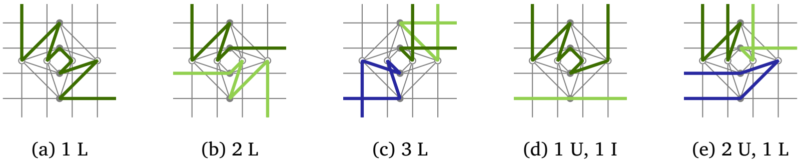


Fig. 12. Different possibilities for parts of a Hamiltonian path to cross a unit cell where the letters L, U and I stand for L-turn, U-turn and crossing straight, respectively.

cross the vertices along the path until they reach the last before the odd vertex, corresponding to the original odd neighbor, and return from there.

4.2. Chimera tentacles

In the following, we establish some results about Hamiltonian cycles and paths in the Chimera tentacles. This can be used later on to construct the desired return and cross paths in the full broken Chimera graph $C(B)$. Note that we only handle finite strips and tentacles.

The Chimera graph consists of unit cells of complete bipartite subgraphs with an equal number of vertices in the partitions in the non-broken case. For broken graphs, note that induced subgraphs of $K_{(4,4)}$ are K_{nm}, for $n, m \leq 4$, and in K_{nn} there exist Hamiltonian paths between every pair of vertices that are in different partitions, while in $K_{(n,n-1)}$, $n \geq 2$, we only have Hamiltonian paths between every pair of vertices from the larger partition.

Thus, paths completely covering Chimera unit cells need to fulfill a certain condition: Each unit cell needs to be entered and left by such a path over an equal number of horizontal and vertical edges. This can be realized in different ways, for which we show some combinations in Fig. 12. One or more vertex pairs, consisting of one horizontal and one vertical vertex each, can be traversed after each other to cross the unit cell in an L-shape. Alternatively, by splitting up these pairs, the unit cell can be crossed straight or in a U-turn. Note that the latter parts can only appear in pairs when there are no broken vertices.

For the Chimera tentacles handled in the following, we only use L-turns. Those can still be realized even if pairs of horizontal and vertical vertices are removed, like in unit cells at the beginning and the end of the modified Chimera tentacles. After establishing the general results for the non-modified Chimera tentacles, we show how to transfer them to the modified versions required for $C(B)$.

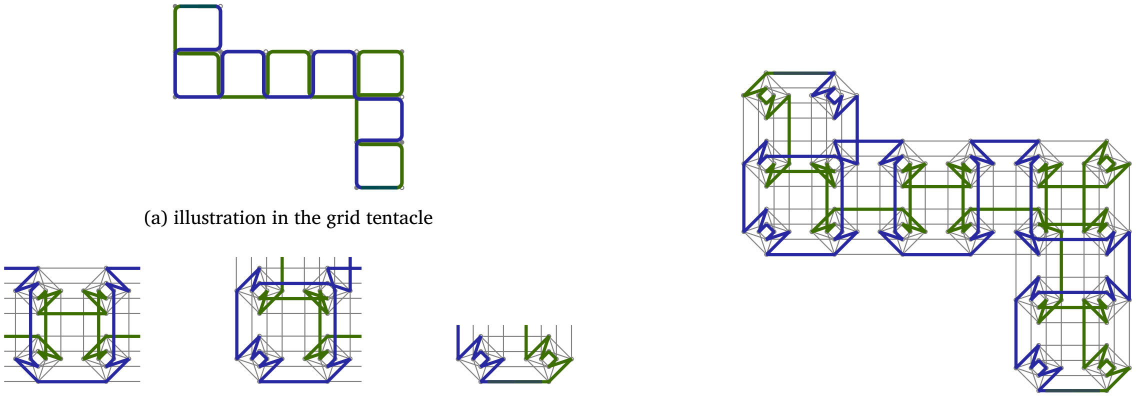
4.2.1. Realizing return paths

In our Chimera construction, we need to make sure that the Chimera tentacles are suitable to realize return paths. By [12], we know every Chimera strip has a Hamiltonian cycle. There, the concept of alternating ‘snake paths’ was shown to cover all unit cells of a rectangular shaped Chimera graph crossing them only using L-turns. Here, we briefly recall the construction and extend the statement to Chimera tentacles:

Lemma 29. *There exists a Hamiltonian cycle for all (finite) Chimera tentacles.*

Proof. We provide the single elements that can be combined to form a cycle covering all vertices in all unit cells of the Chimera tentacle. The cycle consists of two parallel parts along the long sides of the strips: Like in the construction of [12], both parts alternately loop to the other side of the strips by doing two L-turns. The principle is illustrated in the underlying grid tentacle in Fig. 13a. The parts are finally connected in the beginning and the end of the tentacle to close the cycle.

On the Chimera level, the L-turns of the alternating parts are realized by at least one vertex pair, consisting of a horizontal and a vertical vertex, in each unit cell. The remaining vertex pairs inside the unit cell could be used in either of the parts. For symmetry reasons, we use two pairs in each straight part, however, this is not required. An example is shown in Fig. 13b. In the unit cells in the corners and at the ends of the tentacle we can do the turn by simply covering all vertices, which is shown in Figs. 13c and d. In the inner unit cell of the corner, the lower left in Fig. 13c, the parts meet over three L-turns. This way, we can assemble the full cycle as shown in the example in Fig. 13e.



(a) illustration in the grid tentacle

(b) straight elements

(c) corner

(d) closing loop

(e) Chimera tentacle with a possible Hamiltonian cycle

Fig. 13. Hamiltonian cycle construction in the Chimera tentacle example with the two different alternating parts.

Finally, in Figs. 13b-d, we show the possible set of components that is sufficient, apart from rotation of the elements and permuting the ordering of horizontal or vertical vertices in a unit cell due to the symmetry, to construct a Hamiltonian cycle in a tentacle. □

From the constructed Hamiltonian cycle, we can now easily get a Hamiltonian path that allows to cover the whole tentacle by a return path.

Corollary 30. *Let $C(T)$ be a Chimera tentacle derived from a grid tentacle T beginning with strip $S = R^{ab}_{xy}$. There exists a Hamiltonian path in $C(T)$*

a) from a horizontal vertex in V_{ab} to a horizontal vertex in V_{xb} if S is oriented vertically or
b) from a vertical vertex in V_{ab} to a vertical vertex in V_{ay} if S is oriented horizontally.

Proof. We can use the same construction as before in the proof of Lemma 29 just with an open straight element rather than a closing loop at the beginning of the tentacle. □

Remark. By cutting the closing loop between the final unit cells, we can also find a Hamiltonian path from a vertical vertex in V_{ab} to a vertical vertex in V_{xb} if S is oriented vertically or from a horizontal vertex in V_{ab} to a horizontal vertex in V_{ay} if S is oriented horizontally. However, this is not necessary for our construction.

4.2.2. *Realizing cross paths*

Similarly to the paragraph before, we show the suitability of Chimera tentacles to realize cross paths here. First of all, we establish a relation between the partitions of the Chimera graph and the parity of the grid vertices. W.l.o.g., we can specify the parity of the Chimera graph vertices in the following way: The vertices in the horizontal partition of the unit cell at coordinate $(1, 1)$ are called *even* and in the vertical partition *odd*.

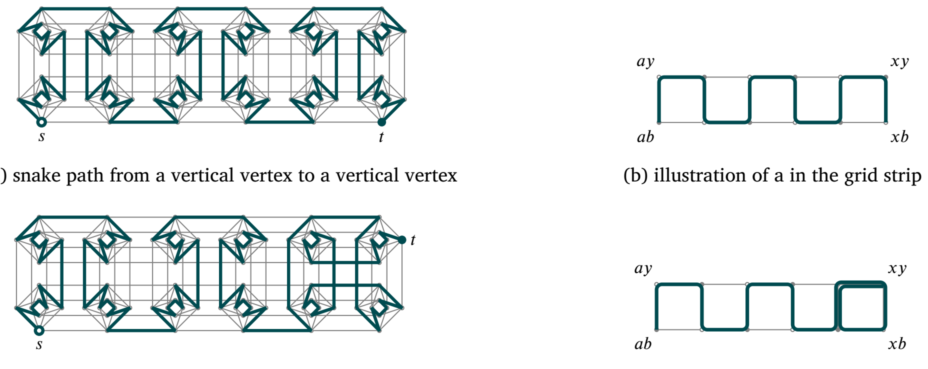
Due to the alternating orientation of the partitions of the Chimera graph over the unit cells, remember Fig. 2, the parity of the other vertices can directly be deduced from the parity of the unit cell coordinates: The above relation is repeated in every unit cell at an even grid coordinate, while it is reversed if the grid coordinate is odd. In other words, for two different unit cells with coordinates of the same parity their horizontal vertices belong to the same partition, so do the vertical vertices. Now we can simply deduce

Corollary 31. *Let $v \in V_{ab}$ and $w \in V_{xy}$ with $ab, xy \in \mathbb{Z}^2$ be two vertices in C_∞. The vertices v and w are of different parity if either*

a) ab and xy have different parity and v and w are oriented equally or
b) ab and xy have the same parity and v and w have different orientation.

With these parity relations, we can now build the base for Hamiltonian paths from 'the beginning' to 'the end' of a Chimera tentacle, thus forming a cross path. We start with simple strips and show the following claim exemplarily for horizontal strips, but it symmetrically holds for vertical ones (by replacing xb with ay in the last sentence).

Lemma 32. *Let $S = R^{ab}_{xy}$ be a horizontal grid strip. There exists a Hamiltonian path from $s \in V_{ab}$ to $t \in V_{xy}$ in the Chimera strip $C(S)$ if s and t have different parity. The same holds if we replace xy with xb (the other corner point at the end of the strip).*



(a) snake path from a vertical vertex to a vertical vertex

(b) illustration of a in the grid strip

(c) snake path from a vertical vertex to a horizontal vertex

(d) illustration of c in the grid strip

Fig. 14. Hamiltonian path construction in a Chimera strip example.

Proof. The partitions in a Chimera strip are of the same size due to the equal number of horizontal and vertical vertices in the full unit cells. Thus, a Hamiltonian path needs to start in one partition and end in the other. It remains to show that there always exists a Hamiltonian path in this case.

As already mentioned before, we need to enter, respectively, leave a unit cell horizontally in the same number as vertically. The easiest way to realize a Hamiltonian path respecting this is a single snake path only consisting of L-turns, analogously to one of the parts as in the proof of Lemma 29.

Assume the starting vertex s belongs to the vertical partition. By iteratively covering all vertices of a unit cell once entered, the path snakes along the strip as shown in Fig. 14a. The principle can be illustrated more simply in the grid, see Fig. 14b. Such a snake path has the following properties:

- The snake path can be extended analogously to cover a strip of arbitrary length.
- Due to the symmetry of the unit cells, we can interchange the ordering of the horizontal and the vertical vertices along the path in a unit cell arbitrarily.
- The final unit cell is either at xy or xb depending on the length of the strip.
- We always enter the final unit cell vertically and thus end up in a vertical vertex.
- As the path also respects the bipartition of the grid graph, the final unit cell has a different parity than the first one.

All in all this means, if t is a vertical vertex like s, it needs to be in the final unit cell according to Corollary 31 and we are done.

If t is a horizontal vertex instead, it can only be in the unit cell in the other corner as in the case before. By slightly modifying the snake path using an additional loop, we can also finish the path there. This is demonstrated in Fig. 14c and d. We can observe that the final unit cell is now of the same parity as the starting one. Furthermore, it is entered horizontally, meaning the path finishes in a horizontal vertex.

In the reverse case, where s is a horizontal and t is a vertical vertex, we can simply use the whole construction in the reverse direction, meaning we either need to rotate it, in case of $t \in V_{xy}$, or mirror it, if $t \in V_{xb}$. Note that, by both transformations, the way the path 'snakes' is inverted. Finally, if both s and t are horizontal, we need to apply the additional loop at both ends of the strip. □

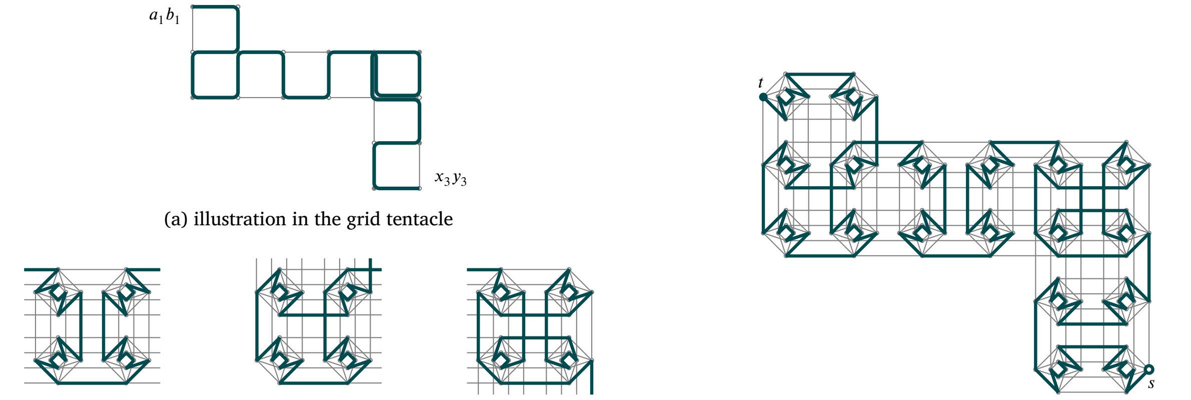
Now we can generalize the result to Chimera tentacles. Note that, with a vertex being *oriented orthogonally* to a strip it is contained in, we mean that it is oriented vertically if the strip is oriented horizontally and vice versa.

Lemma 33. *Let $T = \bigcup_{i=1}^{n} S_i$ be a grid tentacle of a series of strips $(S_i)_{i=1,\dots,n}$ for $n \in \mathbb{N}$ with $S_i = R_{x_i y_i}^{a_i b_i}$. There exists a Hamiltonian path from $s \in V_{a_1 b_1}$ to $t \in V_{x_n y_n}$ in the Chimera tentacle $C(T)$, if s and t have different parity.*

Proof. The necessity of the parity condition follows the same arguments as given in the proof of Lemma 32. We still need to prove the existence of a Hamiltonian path. Analogously to the proof of Lemma 29, we provide one sufficient set of components which can be assembled to a Hamiltonian path. Those components are derived from the paths in the single strips as shown in Lemma 32, which can easily be combined to form a Hamiltonian path through the whole Chimera tentacle.

We explain the concept using the underlying rectangular tentacle as illustrated in Fig. 15a. Assume the starting vertex s lies in a partition that is oriented orthogonally to the starting strip S_1. Then, we can use a simple snake path through the first strip, like in the proof of Lemma 32, finishing in one of the two 'end points'. When crossing over to the next strip, we have two cases: Either the next strip is attached to the side of the strip where the snake path ends or it is attached to the other side.

In the first case, the path can be extended along the edge that both strips share back to the already covered other starting point of the next strip. By this, an inner corner is formed and the snake path can continue in the next strip. This is shown in Fig. 15a in the left corner. In the latter case, we can use the additional loop as explained in the proof of Lemma 32 to move to the other side of the strip and continue from there. This forms an outer corner, like in Fig. 15a on the right at the transition from S_2 to S_3.



(a) illustration in the grid tentacle

(b) straight elements

(c) inner corner

(d) outer corner

(e) Chimera tentacle with a possible Hamiltonian path

Fig. 15. Hamiltonian path construction in the Chimera tentacle example.

By successively adding the next strip according to the specific case, we can assemble the whole path ending up again in a vertex being oriented orthogonally to the final strip. Using the additional loop again, we can change the final vertex to be in a partition oriented equally as the strip. The remaining cases can be achieved by adding the loop in the beginning of the tentacle. However, this again switches the route of the snake path and thus the cases for all corners as well.

Because the whole construction only uses L-turns and does not enter a grid vertex representing a unit cell more than twice, it can easily be realized on the Chimera level like shown in Fig. 15e with the components of Figs. 15b-d. With all rotated and reordered variants of these components we can assemble Hamiltonian paths in all Chimera tentacles. □

4.2.3. Modified chimera tentacles

The previous statements about the Hamiltonicity always refer to arbitrary Chimera tentacles. However, we use the modified version in the constructed broken Chimera graph $C(B)$. But we observe the results are easily transferable:

Corollary 34. *Lemma 29 and Corollary 30 still hold for modified Chimera tentacles.*

Proof. In the proof of both claims, the unit cells corresponding to the start and end points are fully covered by forming an L-turn in the underlying rectangular representation. Pairs consisting of one horizontal and one vertical vertex can be removed from these closing elements until a single pair is left without disturbing the construction. □

Unfortunately, we cannot state the same for Lemmas 32 and 33: The possibly necessary additional loops in the beginning or the end of the tentacle might require more vertices than the modified Chimera tentacles provide as the corresponding unit cells need to be entered more than once. However, by excluding these cases, we can analogously deduce the following result.

Corollary 35. *Let $T = \bigcup_{i=1}^{n} S_i$ be a grid tentacle of a series of grid strips $(S_i)_{i=1,\dots,n}$ for $n \in \mathbb{N}$ with $S_i = R_{x_i y_i}^{a_i b_i}$. There exists a Hamiltonian path from $s \in V_{a_1 b_1}$ to $t \in V_{x_n y_n}$ in the modified Chimera tentacle $\tilde{C}(T)$ if s and t have different parity and s and t are oriented orthogonally to the strips S_1 and S_n, respectively.*

Thus, return and cross paths can analogously be realized in modified Chimera tentacles representing edges in the constructed Chimera graph. We can now continue with connecting these paths in the corresponding vertex elements.

4.3. Vertex elements

In contrast to the Chimera tentacles representing the edges, we have certain broken vertices placed at specific positions in $\tilde{C}_{(2,2)}$, respectively, in $\tilde{C}(S_v)$ for all $v \in V(B)$. Remember Fig. 8 and Definition 25 of Section 3.2. In particular, the unit cell in the upper left corner (without rotation) has an unequal number of vertices in the different partitions. This means we cannot pair all vertices anymore, like in Figs. 12a-c, and a Hamiltonian path in $\tilde{C}(S_v)$, as a part of the overall Hamiltonian cycle, cannot cover all vertices in this unit cell by simple L-turns. It rather needs to follow a more complicated path by using an uneven number of U-turns or straight paths.

Recapturing the notation introduced with Fig. 8, we can establish the following statement about paths in these graphs starting and finishing in the labeled vertices u_1 to u_3.

Lemma 36. *There exists a Hamiltonian path in $\tilde{C}_{(2,2)}$ from*

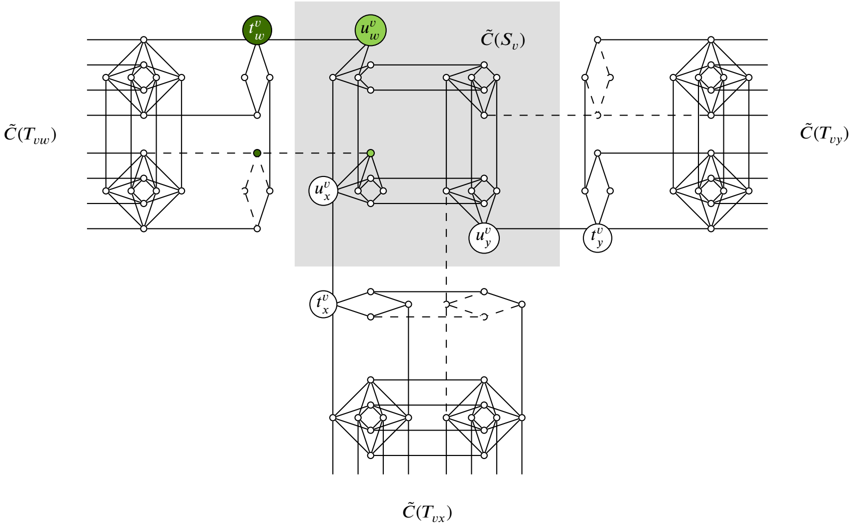


Fig. 16. Vertices joining the vertex Chimera element $\tilde{C}(S_v)$ with the edge Chimera elements $\tilde{C}(T_{vn})$ for $n \in N(v) = \{w, x, y\}$, with an exemplary assignment of the neighboring vertices, for v being an even vertex (including the dashed vertices and edges) or odd (excluded).

a) u_1 *to* u_2 *using none of the additional edges,*
b) u_1 *to* u_2 *using additional edge* l_3 *(and not* l_1*,* l_2*),*
c) u_1 *to* u_3 *using none of the additional edges,*
d) u_1 *to* u_3 *using additional edge* l_2 *(and not* l_1*,* l_3*),*
e) u_2 *to* u_3 *using none of the additional edges,*
f) u_2 *to* u_3 *using additional edge* l_1 *(and not* l_2*,* l_3*).*

Proof. By inspection, compare Fig. 17 on page 18. □

4.4. Assembly

Now we have collected all the single elements that are necessary to assemble a full Hamiltonian cycle in the broken Chimera graph $C(B)$. In this section, we show that all these pieces indeed fit together.

For this, let us specify the vertices forming the junctions between the elements: We denote the vertices in $\tilde{C}(S_v)$ isomorphic to u_1, u_2 and u_3 of $\tilde{C}_{(2,2)}$ by u^v_w, u^v_x and u^v_y, respectively, according to the corresponding neighboring vertices $w, x, y \in N(v)$. In other words, if the Chimera tentacle $\tilde{C}(T_e)$ for edge $e = vw \in E(B)$ replaces l_i for some $i \in \{1, 2, 3\}$ in the depiction of $\tilde{C}(S_v)$ according to Fig. 8, the vertex isomorphic to u_i is now denoted by u^v_w. The vertex corresponding to u_j, $j \in \{1, 2, 3\}$, of $\tilde{C}(S_w)$ is now denoted by u^w_v. Let $t^v_w \in V(\tilde{C}(T_e))$ further be the unique Chimera tentacle vertex which is connected to u^v_w with $t^v_w u^v_w \in E(C(B))$. This notation is also illustrated in Fig. 16.

As we do not know the trace of the Hamiltonian cycle in B in advance, and therefore also not in $C(B)$, we need to ensure that all possible ways of traversing B can equivalently be realized in $C(B)$. At first, this means wherever we have an edge between two vertices in B, we need to be able to cover the corresponding Chimera tentacle by a cross path and with this, in particular, connect the corresponding vertex representations in $C(B)$.

Although this seems trivial by the snake paths introduced before, we need to pay attention to the following: The way such a possible snake path covers a tentacle is predetermined by the single edge connecting the representation of the odd vertex with the tentacle. It is not straightforward to see that the snake path reaches the other end of the tentacle, such that we can always enter the representation of the neighboring even vertex v as expected. That is in the vertex t^v_w, from where we can enter u^v_w and continue according to Lemma 36. We can show that this is indeed true due to the kept properties of the parity-preserving embedding.

Lemma 37. *For all edges* $e = vw \in E(B)$*, a Hamiltonian path in* $\tilde{C}(T_e)$ *from* t^w_v *to* t^v_w *exists.*

Proof. In the Chimera graph, the parity of the unit cell coordinates influences which vertices of which partition of the unit cell are even or odd: In particular, for w.l.o.g. v being an even original vertex, the corresponding coordinate $3x\,3y$ in $L(B)$ is even and we thus observe u^v_n are even for all $n \in N(v)$. Reversely, for w as the neighbor of v being odd, all u^w_m, $m \in N(w)$, are odd, too. Note that this remains valid even if we rotate the $\tilde{C}_{(2,2)}$-subgraph structure. In turn, this means that their neighboring vertices t^v_n and t^w_m are odd and even, respectively. Moreover, t^v_w and t^w_v are oriented orthogonally to their corresponding Chimera strip of $\tilde{C}(T_e)$ and therefore the conditions of Corollary 35 are met and we find a Hamiltonian path accordingly. □

As the next step, we show that crossing a vertex in B, using two arbitrary edges out of three possible, can be analogously realized in $C(B)$. Because a Hamiltonian cycle always uses one edge to enter and one edge to leave a vertex, for a vertex of degree 3, we have one outgoing edge left that is not covered by a Hamiltonian cycle. The corresponding Chimera tentacle needs to be covered by a return path. Thus, we need to make sure that we are able to loop from the representation of an even vertex in all three directions. This means, in turn, that all odd vertices can be touched by a return path. More precisely, we show

Lemma 38. *For all edges $e = vw \in E(B)$, with w.l.o.g. v even and w odd, exists a Hamiltonian path in $\tilde{C}(S_v) \cup \tilde{C}(T_e)$ from u^v_x to u^v_y with $w \neq x, y \in N(v)$.*

Proof. By Lemma 36, we know how to traverse the vertex element from u^v_x to u^v_y analogously including the additional edge in the direction of the attached tentacle. This means that the vertex element is covered partly, left over vertex u^v_w and is entered again in the neighboring unit cell of the vertex element in a vertex whose partition is equally oriented to u^v_w, highlighted (light, green online) in the example of Fig. 16. Equivalently, the loop in the Chimera tentacle starts and ends in the first strip of the tentacle in equally oriented vertices, more precisely, in t^v_w and the corresponding vertex in the neighboring strip unit cell, respectively, both highlighted (dark, green online) in Fig. 16. Thus, these vertices have different parity and there exists a Hamiltonian path between these vertices by Corollary 30 forming the desired return path in the Chimera tentacle. By finally covering the remaining vertices in the vertex element, we can build the whole desired Hamiltonian path. □

Remark. In case vertex v has a degree of only 2, where w.l.o.g. y is not an element of $N(v)$, the above lemma still holds, when we keep the notation u^v_y to identify the corresponding vertex. However, these cases are not relevant when constructing the overall cycle as v can only be crossed over the edges vw and vx. Thus, the element $\tilde{C}(S_v)$ can only be entered over the edges $t^v_w u^v_w$ and $t^v_x u^v_x$.

For the sake of completeness, we provide the parts of all possible pathways in the elements of a vertex of degree 3 in the Chimera graph in Fig. 17. Additionally, we show in Fig. 18 how the parts of the Hamiltonian cycle are connected in the extracted example of Fig. 10. The illustrations of the pathways of Fig. 17 in the grid are shown in Fig. 19 on page 19. Finally, we can now conclude our example with the illustration of the full Hamiltonian cycle in the grid in Fig. 20.

4.5. Mutual induction

In this section, we bring together the results of all the previous sections and conclude the overall complexity result with the proof of Lemma 15. It consists of two different parts that need to be shown for all $B \in \mathcal{B}$:

a) If there exists a Hamiltonian cycle in B, there exists a Hamiltonian cycle in $C(B)$.
b) If there exists a Hamiltonian cycle in $C(B)$, there exists a Hamiltonian cycle in B.

We actually prove two even stronger statements, where the two cycles do have a clear correspondence to each other and can be converted into another by applying specific transformation steps. This is what we mean in the following with the term ‘induced’. As $C(B)$ is specifically designed to fulfill the following property, we can easily show at first:

Lemma 39. *A Hamiltonian cycle in $B \in \mathcal{B}$ induces a Hamiltonian cycle in $C(B)$.*

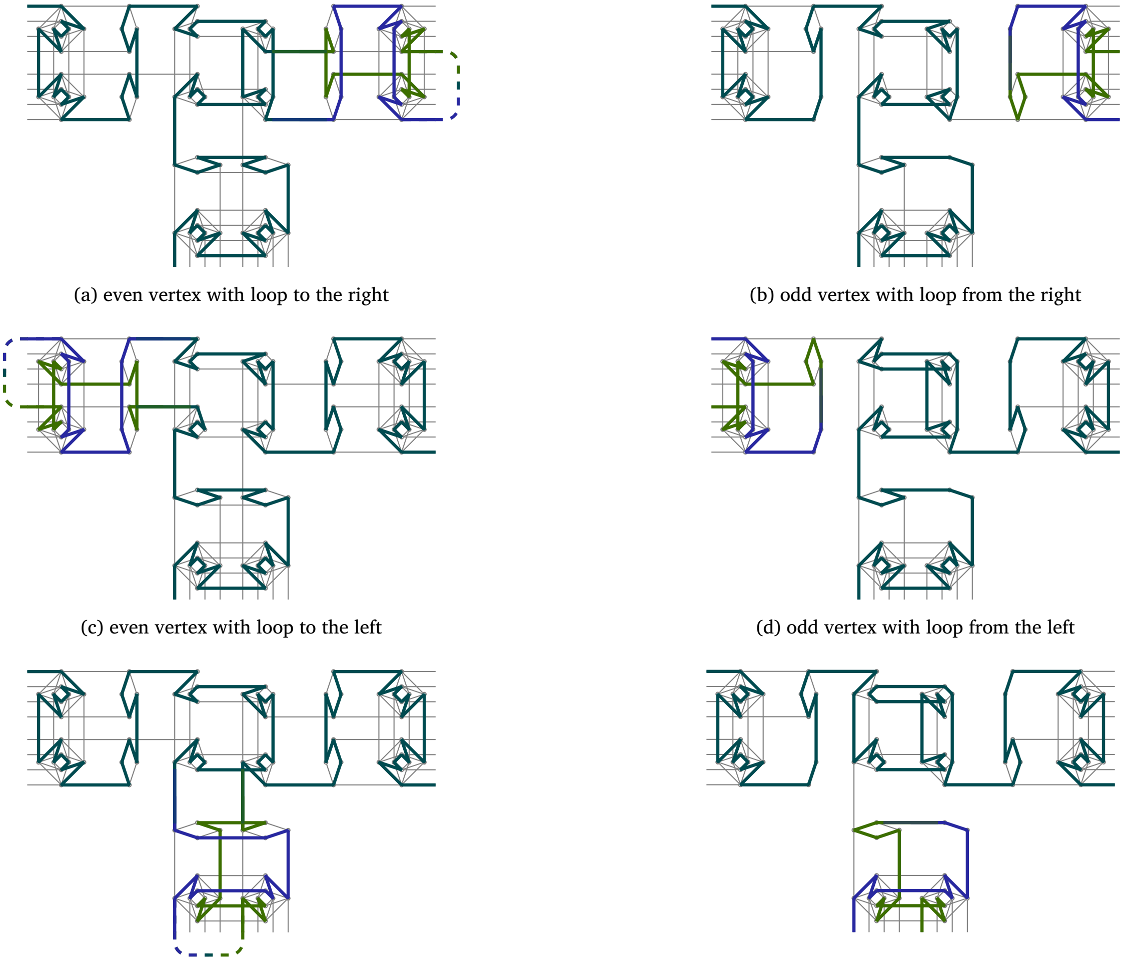
Proof. Let H be a Hamiltonian cycle in B. We have two cases for the edges in B: For all edges in the cycle $e = vw \in E(H)$, we can find a Hamiltonian path from t^v_w to t^w_v in $\tilde{C}(T_e)$ according to Lemma 37. For all remaining edges $e = vw \in E(B) \setminus E(H)$, let v denote the even incident vertex. As v is also covered by the Hamiltonian cycle, it needs to have two further neighbors $x, y \neq w \in N(v)$ with $vx, vy \in E(H)$. By Lemma 38, we can find a Hamiltonian path in $\tilde{C}(S_v) \cup \tilde{C}(T_e)$ from u^v_x to u^v_y. Thus, all vertices in the Chimera tentacles representing the edges are covered.

By the latter edge case, we have indeed handled all even vertices of degree 3 already. For the remaining vertices, $v \in V(B)$ being odd or having degree 2, we can find a Hamiltonian path from u^v_w to u^v_x equivalently to Lemma 36 for $w, x \in N(v)$ being the neighboring vertices where we have $vw, vx \in E(H)$. Hence, all vertices in the Chimera vertex elements are covered, too.

By finally connecting all Hamiltonian path elements for the vertices and edges with the bridging Chimera edges $t^v_w u^v_w$ and $t^w_v u^w_v$ for all edges in the original Hamiltonian cycle $vw \in E(H)$, we have assembled the corresponding Hamiltonian cycle in $C(B)$. □

For the reverse direction, we cannot assume that a possible Hamiltonian cycle in $C(B)$ exactly follows our construction. Nevertheless, we can show the analogous result to above using the following simple result about the Hamiltonicity of graph minors formed by contracting an edge of a Hamiltonian cycle:

Lemma 40. *Let G be an arbitrary simple graph with more than three vertices and Hamiltonian cycle H. Furthermore, let $e \in E(H)$ be an edge of G that is covered by the Hamiltonian cycle H, $\tilde{G}$ be the graph that is formed by contracting edge e and $\tilde{H}$ be the cycle in $\tilde{G}$ that is analogously formed by contracting edge e. Then $\tilde{H}$ is a Hamiltonian cycle of $\tilde{G}$.*



(a) even vertex with loop to the right

(b) odd vertex with loop from the right

(c) even vertex with loop to the left

(d) odd vertex with loop from the left

(e) even vertex with loop to below

(f) odd vertex with loop from below

Fig. 17. Parts of the Hamiltonian cycle through the broken Chimera graph representing a vertex with three incident edges.

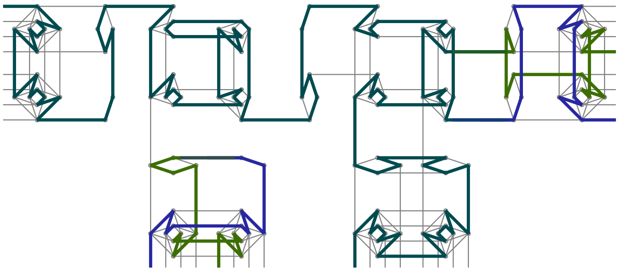
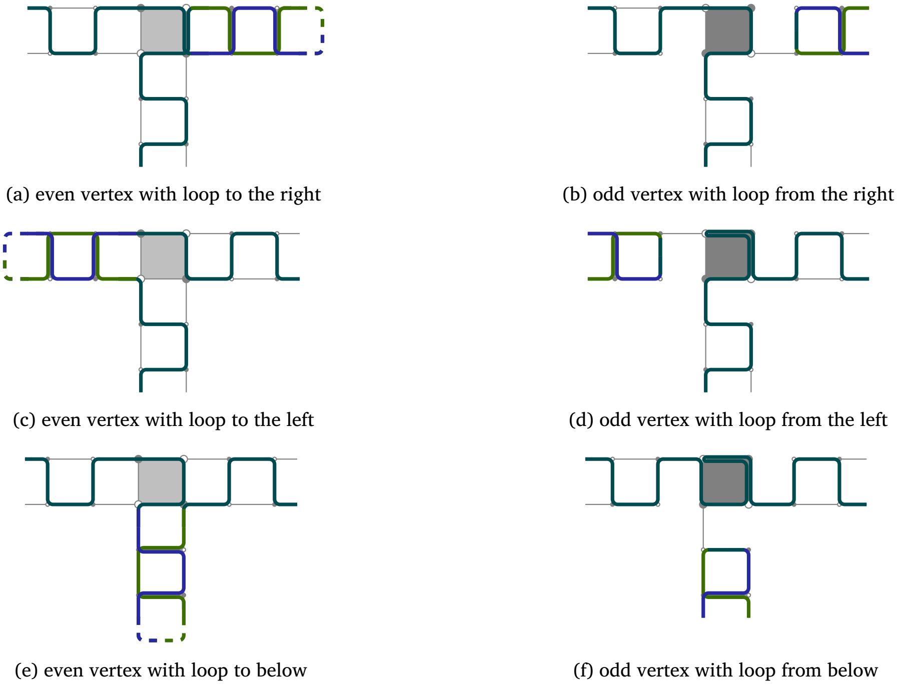


Fig. 18. Parts of the Hamiltonian cycle in neighboring vertices of the example in Fig. 10.

Proof. By contracting an edge, the cycle remains a cycle. As it still covers all vertices of $\tilde{G}$ including the newly formed vertex from the contraction, it also remains Hamiltonian. □

Remark. In case of G having only three vertices, the graph and thus also the Hamiltonian cycle is a triangle. Only in case we allow multiple edges, the graph resulting from contracting an edge still has a Hamiltonian cycle. However, by the formerly mentioned additional removal of multiple edges, the resulting simple graph collapses to a single edge and no Hamiltonian cycle exists anymore.

By *contracting along* a set of edges P forming a path, we mean in the following that each edge in P is successively contracted until just a single vertex is left. Consequently a Hamiltonian cycle containing P in the original graph induces a Hamiltonian cycle in the graph resulting from the contraction of P. Now we can prove our final lemma



(a) even vertex with loop to the right

(b) odd vertex with loop from the right

(c) even vertex with loop to the left

(d) odd vertex with loop from the left

(e) even vertex with loop to below

(f) odd vertex with loop from below

Fig. 19. Parts of the Hamiltonian cycle through the broken Chimera graph representing a vertex with three incident edges illustrated in the underlying grid graph.

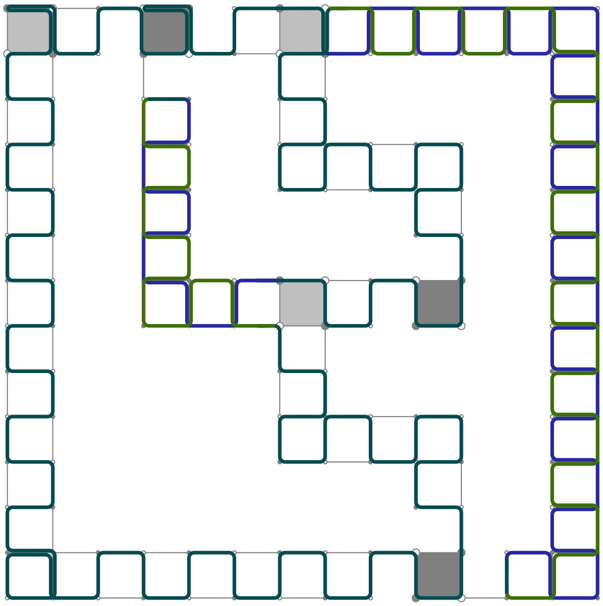


Fig. 20. Cycle in $L(B^*)$ illustrating the Hamiltonian cycle in $C(B^*)$.

Lemma 41. *A Hamiltonian cycle in $C(B)$ induces a Hamiltonian cycle in $B \in \mathcal{B}$.*

Proof. Let H be a Hamiltonian cycle in $C(B)$. We show in the following that it is possible to contract $C(B)$ along paths in H, such that the resulting graph corresponds to B and the analogously contracted cycle derived from H is a Hamiltonian cycle in B according to Lemma 40.

For all edges $e = vw \in E(B)$ with w.l.o.g. v even and w odd, the corresponding Chimera tentacle $\tilde{C}(T_e)$ is connected to the vertex element $\tilde{C}(S_v)$ by two edges and to $\tilde{C}(S_w)$ by a single edge. Thus, H can only enter and leave the tentacle a single time. The corresponding part of the cycle forms a Hamiltonian path in the tentacle, always either from and to the even vertex representation or from the even to the odd vertex representation. How these paths are realized in detail is not of interest, but they form a kind of return or cross paths, like in our construction of a Hamiltonian cycle in $C(B)$.

We can analogously revert the steps of the construction by contracting along the Hamiltonian paths in the single elements: Each Chimera tentacle is contracted, such that only a single edge remains connecting the different vertex representations. While cross paths result in an edge covered by the resulting Hamiltonian cycle, return paths are contracted into the even vertex representation. The single edge between the representation of the odd vertex and the Chimera tentacle is kept. It is not covered by the Hamiltonian cycle in $C(B)$ and also not in the contraction. By further contracting the paths in the vertex elements until a single vertex remains, we can obtain the original graph B together with a Hamiltonian cycle in it. □

Finally, by the last two lemmata, the proof of Lemma 15 follows directly and we have completed the reductions of Section 2.3.

5. Transfer to derived graphs

In this section, we briefly transfer our findings about the Chimera graph to the latest currently operating hardware structure, the *Pegasus* graph, and the recently announced new architecture *Zephyr*. Both are derived from the Chimera graph by stretching and shifting the underlying superconducting loops. By this, the grid structure of unit cells is preserved, but a larger connectivity between the vertices can be realized, in particular, bridging different unit cells. As the Pegasus graph and the Zephyr graph are less accessible than the Chimera, we do not go into more details here. Please see [13] for a complete formal description of the Pegasus graph P_n for $n \in \mathbb{N}$ and its different derived versions and [14] for the Zephyr graph Z_n, $n \in \mathbb{N}$. Both are also described in the D-Wave documentation [15].

5.1. Pegasus

Here, we concentrate on the 'standard' Pegasus graph, meaning with the default shift, as currently available in hardware by the D-Wave Advantage system. We use the same term 'broken' as for the Chimera graph when considering a vertex-induced subgraph of a Pegasus graph. Analogously to $\mathcal{C}$, let $\mathcal{P}$ be the set of all finite broken Pegasus graphs. We can now state a similar question as before, where we can show that the hardness of this question relates to the one for the Chimera graph.

Definition 42 *(*Broken Pegasus Minor Embedding Problem*).* Given an arbitrary graph G and some $P \in \mathcal{P}$, is G a minor of P, i.e., is G embeddable into P?

Fig. 21 illustrates the relation between the Chimera and the Pegasus graph. The highlighted subgraph of the depicted Pegasus graph does nearly have the same connectivity as the Chimera graph apart from the additional edges inside the unit cells (marked in red online). Let C^+_{cr} be the Chimera graph derived from C_{cr} by adding these edges. This means an edge from the first to the second vertex and an edge from the third to the fourth vertex in each partition in each unit cell of C_{cr}, when enumerating the vertices along the depicted ordering. Now we can easily see from the figure that C^+_{cr} is isomorphic to a vertex-induced subgraph of a (standard) Pegasus graph. In other words, it is a broken Pegasus graph.

By enclosing the additional edges, wherever both vertices are non-broken, in each partition of each unit cell in all broken Chimera elements of Section 3, we can construct analogous elements: For $\tilde{C}(S_v)$ of Definition 25 and $C(T_e)$ of Definition 26, respectively, $\tilde{C}(T_e)$ of Definition 27 we get $\tilde{C}^+(S_v)$ for all $v \in V(B)$ and $C^+(T_e)$, respectively, $\tilde{C}^+(T_e)$ for all $e \in E(B)$ and thus finally $C^+(B)$ for $B \in \mathcal{B}$. This can still be done in polynomial time and we have $C^+(B) \in \mathcal{P}$.

Although $C^+(B)$ is not bipartite anymore, we also keep the chequered pattern of even and odd vertices. It is easy to see that all the statements, lemmas and corollaries, in Section 4 still hold when replacing all occurrences of Chimera elements with the corresponding ones from above. In particular, the proof of Lemma 41 remains valid because the additional edges just concern vertices inside a unit cell and the Chimera tentacles are thus still connected to the vertex elements by only three edges in total.

This way, we can deduce the following:

Lemma 43. *The* Hamiltonian Cycle Problem *for graphs $P \in \mathcal{P}$ is NP-complete.*

Similarly to the treatment of Chimera graphs, this implies

Theorem 44. *The* Broken Pegasus Minor Embedding Problem *is NP-complete.*

Note that we could replace the set of edges added between the vertices in each unit cell in $C^+(B)$ by a different one. This would not interfere with our Hamiltonian cycle construction nor with the argument for the reverse induction of a Hamiltonian cycle in the original graph. Even if we consider a Chimera graph structure but with unit cells forming a complete subgraph, the above argumentation also holds. Thus, the decisive pattern here is having a vertex-induced subgraph that consists of unit cells with a $K_{(4,4)}$-subgraph being arranged and connected in a grid pattern. This means, for all new hardware architectures, if they fulfill this property, the embedding problem remains hard in general.

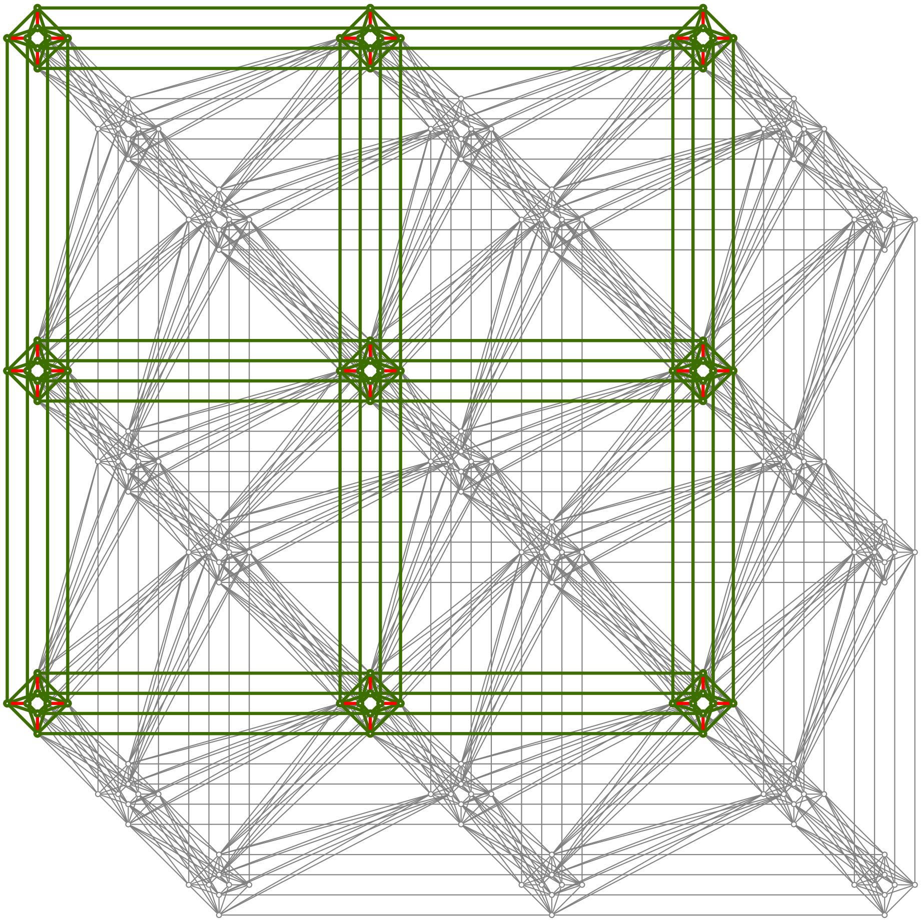


Fig. 21. Inner of the Pegasus graph P_5 without the incomplete unit cells at the boundary with the Chimera subgraph $C^+_{(3,3)}$ (bold, green online) and its additional edges (red online).

5.2. Zephyr

Again, let analogously $\mathcal{Z}$ be the set of all finite broken Zephyr graphs. We have the same question as before.

Definition 45 *(*Broken Zephyr Minor Embedding Problem*).* Given an arbitrary graph G and some $Z \in \mathcal{Z}$, is G a minor of Z, i.e., is G embeddable into Z?

In Fig. 22, we can easily see that we can directly find an induced subgraph in the Zephyr graph that is isomorphic to a Chimera graph. Without any further reduction steps, we can therefore analogously, and for the sake of completeness, state

Lemma 46. *The* Hamiltonian Cycle Problem *for graphs $Z \in \mathcal{Z}$ is NP-complete.*

This finally implies also

Theorem 47. *The* Broken Zephyr Minor Embedding Problem *is NP-complete.*

6. Outlook

In the given proof we use a very restrictive construction for the broken Chimera graph. While certain broken vertices are indispensable, for instance those in the odd vertex representation, some vertices could be added again and the results would still hold. In particular, the 'holes' between the Chimera tentacles yield a large number of broken vertices: From Lemma 21 we see that the area covered by the broken Chimera graph grows quadratically, while the number of unit cells that are actually used in the construction might only depend linearly on the size of original graph. Thus, the holes would occupy a quadratic number of unit cells, meaning with growing graph size, the ratio of non-broken vertices would tend to zero.

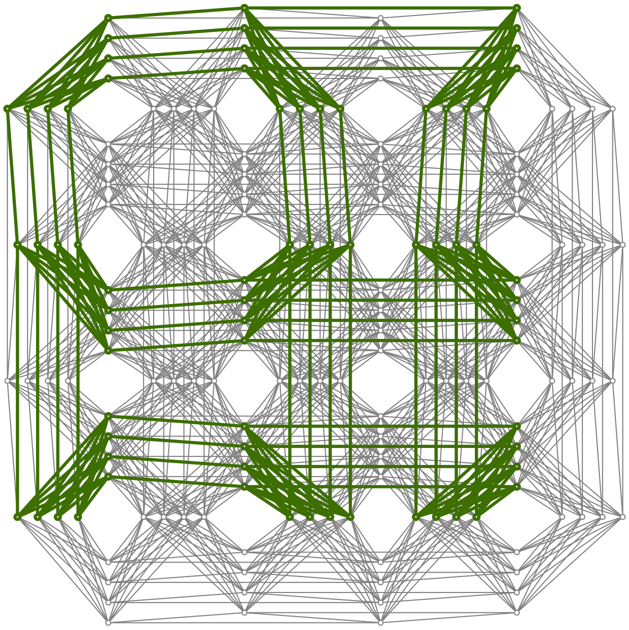


Fig. 22. Zephyr graph Z_2 with the Chimera subgraph $C_{3,3}$ (bold, green online).

Currently operating annealing hardware has in turn a very small ratio of broken vertices. One possibility to overcome the gap could be to fill up the holes with non-broken unit cells connected to the tentacles, such that they can be covered by return or cross paths, too. A first step would be therefore to estimate the best broken vertex ratio which we can achieve with such a construction within the bounds given by the rectangular representation. Another open research question is whether we can restrict the broken vertex ratio, for example to a certain linear function in the size of the Chimera graph, and still show that the problem is hard.

Another open problem is if the construction also holds for Pegasus graphs with an arbitrary shift different than the standard one, as explained in [13]. Can we always find the Chimera graph as a subgraph in these Pegasus graphs like in Section 5.1? As the reduction to the Chimera includes to declare two thirds of the Pegasus vertices to be broken, there is also room for further improvement concerning the broken vertex ratio.

CRediT authorship contribution statement

Elisabeth Lobe: Conceptualization, Formal analysis, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Annette Lutz:** Investigation, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] D-Wave Systems Inc., D-wave systems documentation – QPU solver datasheet, https://docs.dwavesys.com/docs/latest/doc_qpu.html. (Accessed 14 September 2023).

[2] M. Jünger, E. Lobe, P. Mutzel, G. Reinelt, F. Rendl, G. Rinaldi, T. Stollenwerk, Quantum annealing versus digital computing: an experimental comparison, J. Exp. Algorithmics (JEA) 26 (2021) 1–30, https://doi.org/10.1145/3459606.

[3] V. Choi, Minor-embedding in adiabatic quantum computation: I. The parameter setting problem, Quantum Inf. Process. 7 (5) (2008) 193–209, https://doi.org/10.1007/s11128-008-0082-9.

[4] A. Lucas, Ising formulations of many NP problems, Front. Phys. 2 (2014) 5, https://doi.org/10.3389/fphy.2014.00005.

[5] D. Venturelli, D.J. Marchand, G. Rojo, Quantum annealing implementation of job-shop scheduling, preprint, arXiv:1506.08479, 2015.

[6] E.G. Rieffel, D. Venturelli, B. O'Gorman, M.B. Do, E.M. Prystay, V.N. Smelyanskiy, A case study in programming a quantum annealer for hard operational planning problems, Quantum Inf. Process. 14 (1) (2015) 1–36, https://doi.org/10.1007/s11128-014-0892-x.

[7] T. Stollenwerk, B. O'Gorman, D. Venturelli, S. Mandrà, O. Rodionova, H. Ng, B. Sridhar, E.G. Rieffel, R. Biswas, Quantum annealing applied to de-conflicting optimal trajectories for air traffic management, IEEE Trans. Intell. Transp. Syst. 21 (1) (2019) 285–297, https://doi.org/10.1109/TITS.2019.2891235.

[8] T. Stollenwerk, E. Lobe, M. Jung, Flight gate assignment with a quantum annealer, in: International Workshop on Quantum Technology and Optimization Problems, Springer, 2019, pp. 99–110.

[9] V. Choi, Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design, Quantum Inf. Process. 10 (3) (2011) 343–353, https://doi.org/10.1007/s11128-010-0200-3.

[10] C. Klymko, B.D. Sullivan, T.S. Humble, Adiabatic quantum programming: minor embedding with hard faults, Quantum Inf. Process. 13 (3) (2014) 709–729, https://doi.org/10.1007/s11128-013-0683-9.

[11] E. Lobe, L. Schürmann, T. Stollenwerk, Embedding of complete graphs in broken chimera graphs, Quantum Inf. Process. 20 (7) (2021) 1–27, https://doi.org/10.1007/s11128-021-03168-z.

[12] E. Lobe, Quadratische binäre Optimierung ohne Nebenbedingungen auf Chimera-Graphen (Quadratic binary optimization without constraints on Chimera graphs) (in German), Master's thesis, Otto-von-Guericke-Universität Magdeburg, 2016, elib.dlr.de/112063/.

[13] K. Boothby, P. Bunyk, J. Raymond, A. Roy, Next-generation topology of D-Wave quantum processors, preprint, arXiv:2003.00133, 2020.

[14] K. Boothby, A.D. King, J. Raymond, Zephyr topology of D-Wave quantum processors, D-Wave Technical Report Series, 2021, https://www.dwavesys.com/media/2uznec4s/14-1056a-a_zephyr_topology_of_d-wave_quantum_processors.pdf.

[15] D-Wave Systems Inc., D-Wave systems documentation – Getting started with D-Wave solvers – D-Wave QPU architecture: topologies, https://docs.dwavesys.com/docs/latest/c_gs_4.html. (Accessed 14 September 2023).

[16] N. Robertson, P.D. Seymour, Graph minors. XIII. The disjoint paths problem, J. Comb. Theory, Ser. B 63 (1) (1995) 65–110, https://doi.org/10.1006/jctb.1995.1006.

[17] I.V. Hicks, Branch decompositions and minor containment, Networks, Int. J. 43 (1) (2004) 1–9, https://doi.org/10.1002/net.10099.

[18] I. Adler, F. Dorn, F.V. Fomin, I. Sau, D.M. Thilikos, Faster parameterized algorithms for minor containment, Theor. Comput. Sci. 412 (50) (2011) 7018–7028, https://doi.org/10.1016/j.tcs.2011.09.015.

[19] I. Adler, F. Dorn, F.V. Fomin, I. Sau, D.M. Thilikos, Fast minor testing in planar graphs, Algorithmica 64 (1) (2012) 69–84, https://doi.org/10.1007/s00453-011-9563-9.

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

[21] T.D. Goodrich, B.D. Sullivan, T.S. Humble, Optimizing adiabatic quantum program compilation using a graph-theoretic framework, Quantum Inf. Process. 17 (5) (2018) 118, https://doi.org/10.1007/s11128-018-1863-4.

[22] T. Serra, T. Huang, A. Raghunathan, D. Bergman, Template-based minor embedding for adiabatic quantum optimization, preprint, arXiv:1910.02179, 2019.

[23] A. Itai, C.H. Papadimitriou, J.L. Szwarcfiter, Hamilton paths in grid graphs, SIAM J. Comput. 11 (4) (1982) 676–686, https://doi.org/10.1137/0211056.

[24] B. Korte, J. Vygen, Combinatorial Optimization: Theory and Algorithms, 6th edition, Algorithms and Combinatorics, vol. 21, Springer, 2018.

[25] R. Diestel, Graph Theory, 5th edition, Graduate Texts in Mathematics, vol. 173, Springer, 2017.

[26] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Series of Books in the Mathematical Sciences, W. H. Freeman and Company, 1979.

[27] H. Neven, V.S. Denchev, M. Drew-Brook, J. Zhang, W.G. Macready, G. Rose, Nips 2009 demonstration: binary classification using hardware implementation of quantum annealing, Quantum (2009) 1–17, http://www.google.com/googleblogs/pdfs/nips_demoreport_120709_research.pdf.