The complexity of Domino Tiling

Therese Biedl *

Abstract

In this paper, we study the problem of how to tile a layout with dominoes. For non-coloured dominoes, this can be determined easily by testing whether the layout graph has a perfect matching. We study here tiling with coloured dominoes, where colours of adjacent dominoes must match. It was known that this problem is NP-hard when the layout graph is a tree. We first strengthen this NP-hardness result in two ways: (1) we can use a path instead of a tree, or (2) we can force that exactly all given dominoes are used. However, both these reductions (as well as the existing one) use an unbounded numbers of colours, which is not realistic for domino tiling. As our main interest, we hence study domino tiling with a constant number of colours. We show that this is NP-hard even with 3 colours.

1 Introduction

In this paper, we consider the problem of tiling a given layout with dominoes. More specifically, a *layout* L is an integral orthogonal polygon, i.e., a polygon for which all edges are horizontal or vertical and have integer length. A *domino* is a 1×2-rectangle. We want to test whether there is a placement of dominoes such that every point inside L is covered by exactly one domino, and no point outside L is covered by a domino.

It is folklore that this can be tested in polynomial time as follows. Translate L such that none of its vertices has integer coordinates. Now let V be all grid points (points with integer coordinates) that are inside the layout, and let E be all edges between two vertices of V at unit distance. The pair (V, E) defines the *layout graph* G^L. See also Figure 1(a) and (b).

L can be tiled with dominoes if and only if G^L has a *perfect matching*, i.e., a set of edges M such that every vertex is incident to exactly one edge in M. The existence of such a matching can be tested in $O(n^{3/2})$ time [6], where n is the number of vertices of G^L, since G^L is bipartite with $O(n)$ edges.

In this paper, we study domino tiling while considering colours of dominoes; this appears to first have been studied in [10]. Let $C = \{c_1, \ldots, c_l\}$ be a finite set of l colours. A *coloured domino* is an unordered pair (c_i, c_j). *Uni-coloured dominoes*, i.e., dominoes for which $c_i = c_j$, are not expressively forbidden, but will not be used in our NP-hardness reductions. A *coloured domino tiling* of a layout graph G^L is similar to a domino tiling, except that colours of dominoes must match up. Thus, given a multi-set D of coloured dominoes, a *coloured domino tiling*[1] of G^L with D is a perfect matching M in G^L and a colouring $c : V \to C$ such that

*School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada, e-mail biedl@uwaterloo.ca. Research partially supported by NSERC.

[1] Our definition is different from the one in [10], but describes exactly the same concept.

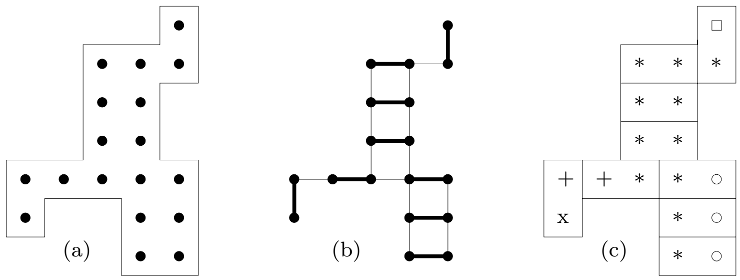

Figure 1: (a) Grid points inside the layout. (b) The layout graph with a perfect matching (bold.) (c) A tiling with coloured dominoes. All vertices marked with the same symbol must be coloured the same.

- if (v, w) is an edge in M, then $(c(v), c(w))$ is a domino in D,
- if (v, w) is an edge not in M, then $c(v) = c(w)$, and
- every domino is used at most once, i.e., there is an injective mapping from $\{(c(v), c(w)) : (v, w) \in M\}$ onto D.

See also Figure 1(d). In what follows, we never consider non-coloured dominoes and/or tilings, and hence drop "coloured" from now on.

We can distinguish domino tilings by whether they are using all given dominoes. In the Exact Domino Tiling problem every domino must be used exactly once in the tiling, so the number of dominoes must equal half the number of vertices of the layout graph. In the Partial Domino Tiling problem we can have arbitrarily many dominoes. Watson and Worman [10] showed the following results:

- Exact Domino Tiling is solvable if the layout graph is a path or a cycle.
- Partial Domino Tiling is NP-hard, even if the layout graph is a tree.

Watson and Worman left as an open question whether Exact Domino Tiling is NP-hard. We prove this, and strengthen their NP-hardness results, as follows:

- We show that Exact Domino Tiling is NP-hard, even if the layout graph is a *caterpillar*, i.e., a tree that consists of a path with degree-1 vertices attached. Alternatively, it is NP-hard even if the layout is a union of paths.
- We show that Partial Domino Tiling is NP-hard, even if the layout graph is a cycle or a path.

Rather than proving these results directly, we relate domino tiling to a graph homomorphism problem, and show its NP-hardness; this graph homomorphism problem may be of independent interest.

Both our NP-hardness proofs and the one in [10] use an unbounded number of colours. This is unrealistic, since normally dominoes are "coloured" with between 0 and 9 dots. We hence next study domino tiling where only a small number of colours are used. As our main result, we show that the problem is NP-hard even if only 3 colours are used.

We are generally only concerned with proving NP-hardness; the domino tiling problems are clearly verifiable in polynomial time, and hence our results really prove NP-completeness.

2 Edge-injective homomorphisms of graphs

Domino tiling can be phrased equivalently in graph-theoretic terms. To do so, we need three graphs related to a domino tiling instance; we review/define these now.

(1) The *layout graph* G^L: This graph was defined earlier and is an induced subgraph of the (2D rectangular) grid, whose vertices are the points inside the layout.

Note that given an induced subgraph of the grid, we can easily construct a layout for which this is the layout graph. Hence we can drop the geometry of the layout and instead consider tiling the layout graph. Note that our definition of domino tiling is purely graph-theoretic as well.

This raises an interesting modification of the problem: We could consider the problem of tiling an arbitrary graph that is not the graph of a layout. This is not needed for our proofs, i.e., the graphs we create are indeed induced subgraphs of the grid.

(2) The *domino graph* G^D: This graph is used to describe the given multi-set of dominoes as graph. Recall that $C = \{c_1, \ldots, c_k\}$ is the set of colours used on dominoes. Define one vertex for every colour. For every domino (c_i, c_j) in the set D of dominoes, add an edge between the vertex of c_i and the vertex of c_j in G^D. We use c_i both for the colour and for the corresponding vertex in G^D, and (c_i, c_j) both for a domino and for its corresponding edge in G^D. Note that G^D is possibly a multi-graph (has multiple edges between a pair of vertices), and it has loops if there are uni-coloured dominoes.

For any graph, we can easily define a multi-set of dominoes for which this is the domino graph; hence the concept of a multi-set of dominoes and of a domino graph are equivalent.

(3) The *contracted graph* $G^C(M)$: Assume we are given a layout graph G^L and a perfect matching M in it. The contracted graph $G^C(M)$ is then defined by contracting every edge that is not in M.[2] Here, *contraction* of an edge (v, w) is the standard graph-theoretic operation defined as follows: Remove edge (v, w). If $v \neq w$, then additionally remove v and w, add a new vertex x, and make it incident to every edge that previously was incident to v or w.

Note that the contracted graph depends on the choice of the perfect matching M. For many simple layouts (paths, cycles, trees) all perfect matchings give the same contracted graph, and we simply write G^C instead of $G^C(M)$.

Domino tilings can be described in terms of mapping the contracted graph into the domino graph. More precisely, a *homomorphism* from graph G to graph H is a mapping σ from $V(G)$ to $V(H)$ such that edges are mapped to edges, i.e., for any edge (v, w) in G, $(\sigma(v), \sigma(w))$ is an edge in H. The homomorphism is *vertex-injective* if it is injective, i.e., no two vertices in G are mapped to the same vertex of H. It is *edge-injective* if the mapping of the edges can be made injective. More formally, for an edge-injective homomorphism we need the homomorphism $\sigma : V(G) \to V(H)$ as before, and an injective edge-mapping $\sigma_E : E(G) \to E(H)$ with $\sigma_E((v, w)) = (\sigma(v), \sigma(w))$.

Graph homomorphisms (also known as *H-colourings*) have been studied extensively before; see for example [3] and the references therein. When adding injectiveness constraints,

[2] This concept was considered only for a path or cycle in the proof of Lemma 1 in [10].

we know of two related topics: Fiala and Kratochvíl [4] study *locally injective homomorphisms*, where for each vertex the incident edges must be mapped injectively. Another related concept are orthogonal graph drawings, which are vertex-injective homomorphisms of a subdivision of G into the rectangular grid (see for example [2]). For domino tiling, we are interested in edge-injective (but not necessarily vertex-injective) homomorphisms. To our knowledge, this topic has not been studied before.

The equivalence of edge-injective homomorphisms and domino tiling follows almost immediately by translating the condition on the colouring function c of a domino tiling to the function σ for the homomorphism. We omit the details here for space reasons.

Theorem 1 *A layout L has a domino tiling if and only if the layout graph G^L has a perfect matching M such that $G^C(M)$ has an edge-injective homomorphism into G^D.*

As a consequence, the results of [10] for Exact Domino Tiling are easily re-proved: An exact domino tiling of a path/cycle corresponds to an edge-bijective homomorphism of a path/cycle into G^D, which exists iff G^D has an Eulerian path/circuit; this can easily be tested in polynomial time.

For our NP-hardness results, we now study the complexity of edge-injective homomorphisms. It is quite straightforward to see that this problem is NP-hard since H-coloring is NP-hard; see also Section 3. We here prove NP-hardness for three special cases of the source graph G.

Theorem 1 *Testing whether G has an edge-injective homomorphism in H is NP-hard, even if*

(a) G is a cycle, or

(b) G is a path, or

(c) G is a caterpillar and G and H have equally many edges.

Proof: The reduction for (a) is done using the Hamiltonian cycle problem defined as follows: Given a 3-regular graph H with n vertices, does H contain a *Hamiltonian cycle*, i.e., a cycle that visits every vertex exactly once? This is known to be NP-hard [5].

Given a 3-regular graph H, let G be a cycle with n vertices. Clearly if H has a Hamiltonian cycle, then G has an edge-injective homomorphism into H. On the other hand, if G has an edge-injective homomorphism into H, then this homomorphism must in fact be vertex-injective, since by 3-regularity of H every vertex in H has only three incident edges. Thus, H contains an n-cycle, which is a Hamiltonian cycle.

The reduction for (b) is similar to (a), except that we use Hamiltonian paths instead of Hamiltonian cycles; the problem of testing whether a 3-regular graph has a Hamiltonian path is also NP-hard [5].

Given a 3-regular graph H, let G be a path with $n+2$ vertices, say $v_0, v_1, \ldots, v_n, v_{n+1}$. Assume we have an edge-injective homomorphism of G into H. This homomorphism need not be vertex-injective, because v_0 and v_{n+1} have only one incident edge in G, and hence one vertex of H could be used both for v_0 (or v_{n+1}) and for some other vertex of G. But for $1 \leq i, j \leq n$, vertices v_i and v_j cannot be mapped to the same vertex in H, since v_i and v_j

have two incident edges each in G, but a vertex in H has degree 3. Therefore, the mapping of $v_1, \ldots, v_n$ forms a path with n vertices in H, which is a Hamiltonian path of H.

Conversely, assume H has a Hamiltonian path $w_1, \ldots, w_n$. Let w_0 be any neighbour other than w_2 of w_1, and let w_{n+1} be any neighbour other than w_{n-1} and w_1 of w_n. Setting $\sigma(v_i) = w_i$ for $i = 0, \ldots, n+1$ then gives an edge-injective homomorphism of G into H.

The reduction for (c) is the most complicated, and again uses Hamiltonian path in 3-regular graphs. (Only here do we use "3-regular" as opposed to "maximum degree 3".) Assume H' is a 3-regular graph with n vertices in which we are searching for a Hamiltonian path. Graph H is obtained from H' by subdividing all edges. More precisely, H contains one vertex u_v for every vertex v of H' and one vertex u_e for every edge e of H', and we add an edge (u_v, u_e) in H if and only if v and e are incident in H'. H' has $\frac{3}{2}n$ edges, so H has $3n$ edges.

Graph G consists of a path $b_1, a_1, b_2, \ldots, a_{n-1}, b_n$ and $n + 2$ more vertices (the *legs*) $l_0, \ldots, l_{n+1}$. For $i = 1, \ldots, n$, leg l_i is adjacent to b_i; leg l_0 is adjacent to b_1 and leg l_{n+1} is adjacent to a_n. See Figure 2. Clearly G has $3n$ edges, so $|E(G)| = |E(H)|$ as desired. Also note that G is a caterpillar.

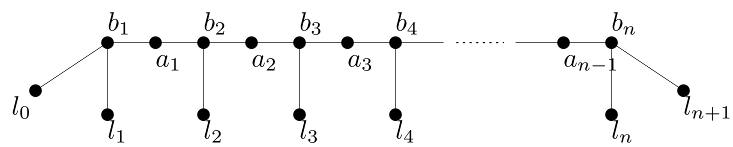

Figure 2: The graph G used for the reduction.

We claim that G has an edge-injective homomorphism into H if and only if H' has a Hamiltonian path. Assume first that H' has a Hamiltonian path $P = v_1 - e_1 - v_2 - e_2 - \ldots - v_n$. Then H contains the path

$$u_{v_1} - u_{e_1} - u_{v_2} - u_{e_2} - \ldots - u_{v_n}.$$

For all i, we map b_i to u_{v_i} and a_i to u_{e_i}. We map the legs to the vertices corresponding to edges not in the Hamiltonian path. More precisely, for $i = 2, \ldots, n-1$, vertex v_i has exactly one incident edge e'_i not in P, and b_i (which maps to u_{v_i}) has one incident leg l_i; we map l_i to $u_{e'_i}$. Similarly v_1 and v_n each have two incident edges not in P, and b_1 and b_n have two incident legs; we map correspondingly. One easily verifies that this is an edge-injective homomorphism of G into H.

For the other direction, assume that G has an edge-injective homomorphism into H. Graph H is bipartite with vertices u_v (for some vertex $v \in H'$) of degree 3, and vertices u_e (for some edge $e \in H'$) of degree 2. Since each b_i has degree 3, it must be mapped to u_{v_i} for some vertex $v_i \in H'$, $i = 1, \ldots, n$. Since H is bipartite, each a_i therefore must be mapped to u_{e_i} for some edge $e_i \in H'$. Since the mapping is a homomorphism, and a_i is adjacent to b_i and b_{i+1}, edges (u_{v_i}, u_{e_i}) and (u_{e_i}, u_{v_i}) must exist in H, which means by construction of H that $e_i = (v_i, v_{i+1})$. So $v_1, \ldots, v_n$ forms a path in H. No vertex can appear twice in the path, since the mapping is edge-injective, each b_i has degree 3, and u_v has degree 3 for any $v \in H'$. So $v_1, \ldots, v_n$ is a Hamiltonian path as desired. □

We now combine Theorem 1 with Observation 1 to achieve our NP-hardness results.

Theorem 2 PARTIAL DOMINO TILING *is NP-hard, even if the layout graph is a path or a cycle.*

Proof: By Theorem 1(a) and (b), testing whether a path/cycle has an edge-injective homomorphism into a graph H is NP-hard. For $n \geq 4$ a path/cycle of length n is the contracted graph of a layout graph (namely, of a path/cycle of length $2n$, which for $n \geq 4$ is an induced subgraph of the grid.) Choosing this layout, and as dominoes a multi-set D with $G^D = H$ gives the desired reduction to domino tiling. □

Using this result, we can now easily prove EXACT DOMINO TILING to be NP-hard, by "padding" the layout with extra 1×2-rectangles. More precisely, assume we have an instance of PARTIAL DOMINO TILING where the layout requires m_1 dominoes, and there are $m_2 > m_1$ dominoes available. Add $m_2 - m_1$ 1×2-rectangles to the layout, which are not adjacent to any other vertices of the layout graph. Then tiling the new layout with the same set of dominoes means an exact domino tiling, and exists if and only if the old layout had a partial domino tiling.

Corollary 1 EXACT DOMINO TILING *is NP-hard, even if the layout is a union of paths.*

However, for this corollary the layout consists of disjoint polygonal regions, not one connected polygon as originally intended for the domino layout problem. We hence give a second NP-hardness proof which overcomes this difficulty.

Theorem 3 EXACT DOMINO TILING *is NP-hard, even if the layout graph is a caterpillar.*

Proof: Theorem 1(c) shows that edge-injective homomorphism is NP-hard even if G is a caterpillar and G and H have equally many edges. Thus similar as in Theorem 2, all that is required to prove is that the graph G used in the proof of Theorem 1(c) is in fact the contracted graph of some layout graph G^L that is a caterpillar. This is indeed the case; see Figure 3. □

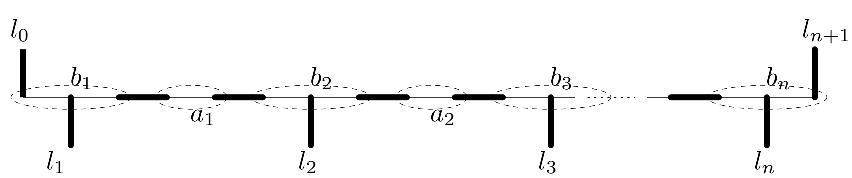

Figure 3: A layout whose graph is a caterpillar, and contracting the unique perfect matching (bold) gives graph G from Figure 2.

Theorem 3 thus solves the open problem posed in [10]: domino tiling is NP-hard even if we require that every given domino is used exactly once. In particular, while EXACT DOMINO TILING is solvable in paths, it becomes NP-hard as soon as we allow multiple paths, or attach legs to a path.

We note here that with a similar proof one can show NP-hardness for EXACT DOMINO TILING for the case of a cycle with legs attached.

3 Few colours

We now study the more realistic case of dominoes which use only a small number of colours. As our main result, we show that the problem then still remains NP-hard, even with only 3 colours.

Note that if we have a constant number of colours, then necessarily there must be many multiple copies of the same type of domino. In fact, our NP-hardness proofs do not use the ability to impose bounds on the number of dominoes. In essence, we therefore revert from edge-injective homomorphisms back to general (not necessarily edge-disjoint) homomorphisms, also known as H-colourings. If we use 3 colours, then the graph H to be mapped to has only 3 vertices.

It is well-known that H-colouring is NP-hard even if H is a 3-cycle. This holds by reduction from 3-colouring, i.e., given a graph G, can the vertices be coloured with 3 colours such that no two endpoints of an edge have the same colour? If H is a 3-cycle, then any homomorphism from G onto H corresponds to a 3-colouring of G, so H-colouring is NP-hard.

However, this does not immediately transfer to NP-hardness for domino tiling for two reasons: (1) The reduction for 3-colouring uses an arbitrary graph G to be mapped to H. Thus, we must find for any graph G a layout graph G^L with $G^C(M) = G$ for some perfect matching M of G^L. To simplify the reduction, we furthermore want that the perfect matching is unique. (2) While it is not too complicated to construct such a graph G^L, even more work must be done to create one that is an induced subgraph of the grid. To ease presentation, we defer this part to later, and first show NP-hardness for graphs that need not be graphs of layouts.

So assume from now on that we are given a graph G and we want to know whether G has a 3-colouring. We may assume that every vertex of G has at least three incident edges. It is known that this problem is NP-hard [5]. We create the layout graph G^L from G in two steps; see also Figure 4.

- Let G' be the graph obtained from G by splitting every vertex into a path. More precisely, if v is a vertex with neighbours $w_1, \ldots, w_k$, then replace v by a path $u_1, \ldots, u_k$ and make u_i adjacent to w_i. Edges in this path are called *path edges*, whereas all other edges (those connecting paths of two different vertices) are called the *original edges.*
- Let G^L be the graph obtained from G' by subdividing every path edge and attaching a vertex of degree 1 to the subdivision vertex. The two edges that replaced the path edge are again called *path edges.* The edges to the vertices of degree 1 are called *leaf edges.*

Since any perfect matching must contain edges incident to a vertex of degree 1, we can easily prove the following:

Claim 1 *G^L has a unique perfect matching consisting of the leaf edges and the original edges.*

Claim 2 *G^C is 3-colourable if and only G is 3-colourable.*

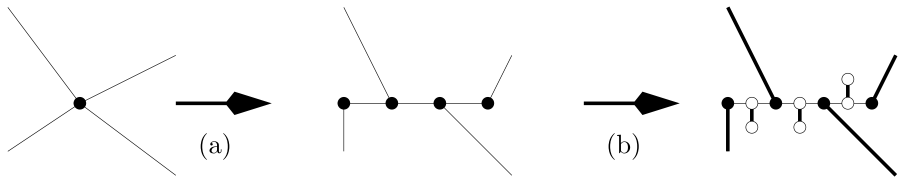

Figure 4: (a) Replacing a vertex by a path. (b) Subdividing the path and attaching leaves (in white), and the unique perfect matching (bold).

Proof: Recall that G^C is obtained from G^L by contracting all edges that are not in the (unique) perfect matching M. In particular, we contract all path edges, and hence undo the splitting of vertices. Thus, G^C is almost the same as G, except that at every vertex v we have a number of leaf edges attached to vertices of degree 1.

If G^C can be 3-coloured, then G (as an induced subgraph) can also be 3-coloured. Conversely, assume G can be 3-coloured. Then for every vertex v, we colour the extra vertices of degree 1 at v with some colour that is not used for v; this gives a 3-colouring of G^C. □

Theorem 4 *Testing whether a graph (not necessarily resulting from a layout) can be tiled with a given set of dominoes is NP-hard, even if only 3 colours are used by the dominoes.*

Proof: Let G be a graph for which we want to test the existence of a 3-colouring. Define the graph G^L as described above. Let D be a set of dominoes with three colours c_1, c_2, c_3, and for each $c_i \neq c_j$, add sufficiently many dominoes (c_i, c_j).[3] Assume G^L has a domino tiling, with colouring function c. Any vertex v in G^C is obtained by contracting some vertices $w_1, \ldots, w_k$ of G^L that are connected by edges not in the perfect matching M. Thus $w_1, \ldots, w_k$ all must have the same colour; set $c(v) = c(w_1)$. Any edge in G^C corresponds to a domino in the tiling; since there are no uni-coloured dominoes therefore this colouring of G^C is a 3-colouring of G^C. By Claim 2 G is 3-colourable. The other direction is similar. If G is 3-colourable, then so is G^C. Assign to each vertex in G^L the colour of the vertex in G^C into which it was contracted. Then all matching edges have differently coloured endpoints, and thus correspond to a domino. This gives a valid domino tiling, since every type of domino exists sufficiently often. □

The rest of this section is devoted to extending Theorem 4 to graphs that result from a layout. The basic idea is the same, but we need to modify our graph further and apply some graph drawing results to find the layout.

Recall that G was the graph which we wanted to 3-colour. We now choose G to be a *planar graph*, i.e., G can be drawn without crossing in the plane. Furthermore, we assume that G has maximum degree 4. It is known that 3-colouring remains NP-hard even for planar graphs with maximum degree 4 [5].

Now we create a *planar orthogonal drawing* of G, i.e., a crossing-free drawing of G on the 2D rectangular grid such that every edge is routed as a sequence of horizontal and vertical

[3] "Sufficiently many" means "enough copies such that we can never run out of dominoes." More precisely, if the layout graph has $2n$ vertices, then n dominoes are needed in the tiling; adding n dominoes of every kind is hence sufficient.

line segments. Such drawings exist for any planar graph with maximum degree 4, see for example [8, 7, 1]. Furthermore, the total edge-length is polynomial.

We obtain the layout by expanding the planar orthogonal drawing of G. To do so, we first scale the drawing, i.e., we replace each row/column of the grid by many rows/columns (20 should be enough), to achieve sufficiently much separation between parallel edge segments. Then we replace each vertex, line segment, and *bend* (place where an edge changes direction) with a gadget. Our gadgets are illustrated in Figure 5.

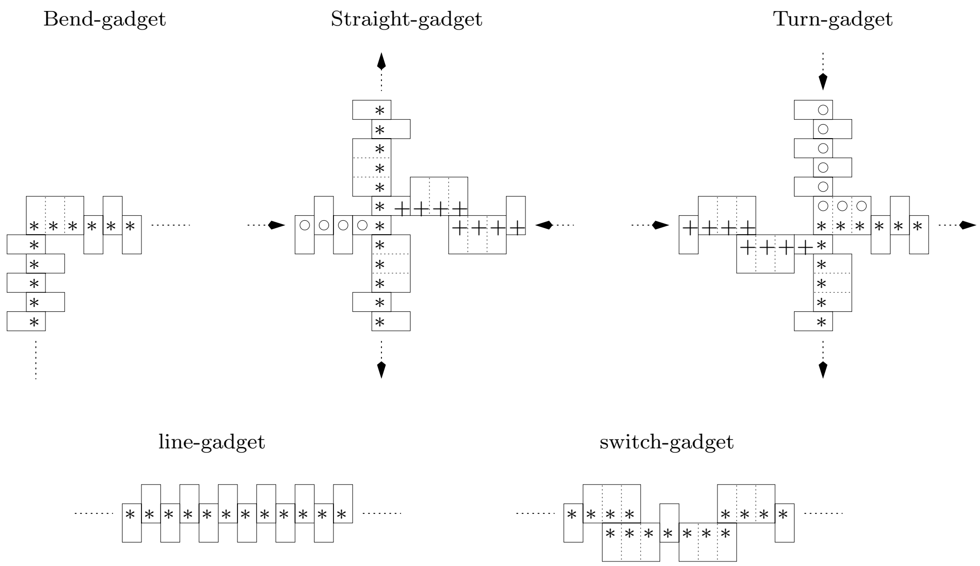

Figure 5: Gadgets used in the reduction.

Before explaining how exactly to do the replacement, we need some properties of the gadgets:

Claim 3 *Assume there are no uni-coloured dominoes. Then any gadget has a unique perfect matching corresponding to the tiling illustrated in Figure 5.*

Proof: (Sketch) For any vertex of degree 1, any perfect matching must contain the incident edge. After accounting for these edges, in each gadgets we are left with some 2×3-rectangles, possibly attached to each other. These subgraphs have more than one perfect matching, but going through cases shows that all but one tiling force one of the dominoes to be uni-coloured. □

Claim 4 *Assume we have a domino tiling of the gadgets without uni-coloured dominoes. Then within any gadget, all squares marked with the same symbol in Figure 5 must have the same colour. Moreover, squares marked $*$ have a different colour than squares marked $+$ or $\circ$. Squares marked $+$ and $\circ$ can, but need not, have the same colour.*

Proof: By Claim 3 there is only one possible perfect matching M. Therefore places marked with the same symbol are connected via edges not in M, and must receive the same colour. Since there exist dominoes $(*, +)$ and $(*, \circ)$, but no domino $(+, \circ)$, the claims about distinctness are immediate. □

Theorem 5 Partial Domino Tiling *is NP-hard, even if only 3 colours are used.*

Proof: Let G be a planar graph with maximum degree 4 for which we want to find a 3-colouring. Create a planar orthogonal drawing of G and scale it as explained before.

Now impose an edge direction onto G such that every vertex has at most two outgoing edges and at most two incoming edges. It is well-known that such an orientation exist (e.g., make G Eulerian by adding edges between vertices of odd degree, direct all edges while walking along an Eulerian circuit, and then delete the added edges.)

Now we replace the drawing by gadgets as follows. Replace each bend with the bend-gadget, rotated if needed such that the lines marked with $*$ coincide with the drawing of the incident edge segments. Each vertex v is replaced with either the straight-gadget or the turn-gadget, depending on the orientations of the outgoing edges of v in the orthogonal drawing. More precisely, assume that vertex v has degree 4. If both outgoing edges of v have the same orientation (both horizontal or both vertical), then we use the straight-gadget; we rotate it if needed such that outgoing edges of v are on lines marked with $*$. Otherwise, we use the turn-gadget, and rotate it such that outgoing edges of v are on lines marked with $*$. Vertices of degree 3 use the same gadgets, but omit one of the four directions as needed.

Finally we complete the drawing by adding line gadgets to replace horizontal or vertical line segments; note that a line gadget can be made arbitrarily long. However, inserting a line gadget may lead to a 2×2-square where it attaches to a bend-gadget or vertex-gadget. This would destroy the uniqueness of the perfect matching, therefore we replace part of the line gadget by the switch gadget. Note that the switch gadget has the same length as the line gadget in Figure 5, but ends in a "down" domino instead of an "up" domino; hence using it avoids creation of a 2×2-square.

This finishes the description of our layout L. The dominoes D have three colours c_1, c_2, c_3, and for each $c_i \neq c_j$, we have sufficiently many dominoes (c_i, c_j). Note that we have no uni-coloured dominoes, so Claims 3 and 4 hold and we have a unique perfect matching and contracted graph G^C.

Assume we have a coloured domino tiling with these dominoes. For each v, let $c(v)$ be the colour used for the places marked $*$ in the gadget replacing v. This is a 3-colouring for G. For if (v, w) is an edge, then it was oriented in some way, say $v \rightarrow w$. Then $c(v)$ and $c(w)$ are the colour $*$ at the gadgets of v and w. But $c(v)$ is transmitted along the drawing of the edge (v, w) via the line gadgets, bend gadgets and switch gadgets (if any) replacing it, and reaches the gadget of w as one of $+$ or $\circ$. Hence $c(v) \neq c(w)$ by Claim 4.

For the other direction, given a 3-coloring of G, we can obtain a domino tiling by letting $*$ at the gadget of v be the colour of v; this requires no uni-coloured dominoes and is a domino tiling since we have sufficiently many dominoes of all types.

Finally, note that since the drawing of G had polynomial total edge length, the construction is polynomial in the size of G, which finishes the NP-hardness proof. □

By modifying the proof slightly, we can extend this result to Exact Domino Tiling. For disconnected layouts, this is straightforward as in Corollary 1, but we can even prove this for connected layouts by using three copies of the previous layout and adding a connection gadget.

Corollary 2 Exact Domino Tiling *is NP-hard, even if only 3 colours are used.*

Proof: Let L be the layout of the previous proof. Attach to it a (sufficiently long) line gadget in such a way that it does not influence whether L has a domino tiling. (For example, we know that the leftmost column of the drawing contains a bend gadget since the graph has minimum degree 3; attach the line gadget at the 2×3-rectangle of the corresponding bend gadget.) See Figure 6(a). Call the new layout L', and assume it contains $2n'$ vertices.

Let L'' be the layout that results from copying L' three times, rotating the copies, and connecting them with a gadget consisting of 9 dominoes as illustrated in Figure 6(b). The set of dominoes is as before, except that each type of domino exists $n' + 3$ times. We have $3n' + 9$ dominoes in total, and L'' has $6n' + 18$ vertices, so if there exists a domino tiling at all, it must be exact.

If L'' has a domino tiling, then L', and therefore L, also has a domino tiling. On the other hand, if L has a domino tiling, then L' has a domino tiling by colouring the attached line gadget arbitrarily; we have sufficiently many dominoes for this since L' needs n' dominoes. Now we can also domino tile the three copies of L', by permuting the colours cyclically for the other two copies. Tiling the three copies thus uses exactly n' dominoes of each kind, and the three "ends" of the line gadgets of the copies of L' are coloured in three different ways. Finally, the connector gadget can be tiled as shown in Figure 6(b); this gives three different colours at the three connection points to the copies of L', and hence a domino tiling of L. □

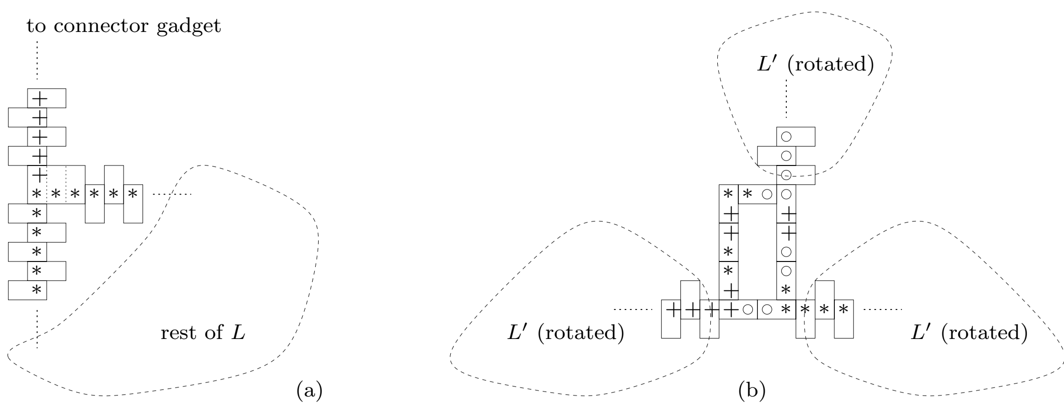

Figure 6: (a) Extending L into L'. (b) Connecting the three copies of L'.

4 Conclusion

In this paper, we studied the domino tiling problem when colours of dominoes are taken into account. We proves NP-hardness results for many variants of this problem. In particular, Partial Domino Tiling is NP-hard even if the layout to be tiled is a path or cycle, and Exact Domino Tiling is NP-hard even if the layout to be tiled is a caterpillar or a union of paths. Of even more interest is domino tiling with few colours; here we established NP-hardness even for three colours.

As for open problems, the main interesting question is domino tiling with two colours. (One colour is the same as not considering colours at all, and hence polynomial.) This would be NP-hard if the following problem were NP-hard: Given a graph G, does G have a perfect matching M such that $G^C(M)$ is bipartite? We do not know the complexity status of this problem.

Another interesting problem is whether domino tiling with a bounded number of colours is polynomial on some special graph classes? In particular, is it polynomial on trees or partial k-trees? (Note that H-colouring is polynomial on these classes [9].) This seems likely with a similar approach as in [9], but we have not been able to give an algorithm that takes into account how many dominoes of each kind are used for each subtree.

Acknowledgments

The author would like to thank Jan Kratochvíl, Mark Watson and Chris Worman for helpful discussions.

References

[1] T. Biedl and G. Kant. A better heuristic for orthogonal graph drawings. *Computational Geometry: Theory and Applications*, 9:159–180, 1998.

[2] G. Di Battista, P. Eades, R. Tamassia, and I. Tollis. *Graph Drawing: Algorithms for Geometric Representations of Graphs*. Prentice-Hall, 1998.

[3] Josep Díaz, Maria J. Serna, and Dimitrios M. Thilikos. The complexity of restrictive H-coloring. In *28th International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 126–137. Springer-Verlag, 2002.

[4] Jiří Fiala and Jan Kratochvíl. Complexity of partial covers of graphs. In *Algorithms and Computation, 12th ISAAC '01, Christchurch, New Zealand*, number 2223 in Lecture Notes in Computer Science, pages 537–549. Springer Verlag, 2001.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[6] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2:225–231, 1973.

[7] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Computing*, 16(3):421–444, 1987.

[8] R. Tamassia and I. Tollis. Planar grid embedding in linear time. *IEEE Transactions on Circuits and Systems*, 36(9):1230–1234, 1989.

[9] J.A. Telle and A. Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM J. Discrete Math*, 10(4):529–550, 1997.

[10] Mark Watson and Chris Worman. Tiling layouts with dominoes. In *16th Conference on Computational Geometry*, pages 86–90, 2004.