*Discrete Mathematics and Theoretical Computer Science* DMTCS vol. **9**, 2007, 107–126

# *On the Tileability of Polygons with Colored Dominoes*

Chris Worman[1] and Boting Yang[1†]

[1]*Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada, S4S 0A2*

*received September 12$^{th}$, 2006, accepted February 21$^{st}$, 2007.*

We consider questions concerning the tileability of orthogonal polygons with colored dominoes. A colored domino is a rotatable $2 \times 1$ rectangle that is partitioned into two unit squares, which are called faces, each of which is assigned a color. In a colored domino tiling of an orthogonal polygon $P$, a set of dominoes completely covers $P$ such that no dominoes overlap and so that adjacent faces have the same color. We demonstrated that for simple orthogonal polygons that can be tiled with colored dominoes, two colors are always sufficient. We also show that for tileable non-simple orthogonal polygons, four colors are always sufficient and sometimes necessary. We describe an $O(n)$ time algorithm for computing a colored domino tiling of a simple orthogonal polygon, if such a tiling exists, where $n$ is the number of dominoes used in the tiling. We also show that deciding whether or not a non-simple orthogonal polygon can be tiled with colored dominoes is NP-complete.

**Keywords:** Domino tiling, Colored domino tiling, Wang tiles, 2-SAT, NP-complete, Orthogonal polygons

## 1 Introduction

We study a computational tiling problem where the tiles model the commonly used domino game piece. We consider tiling orthogonal polygons with so-called *colored dominoes*, which are rotateable $2 \times 1$ rectangles that are partitioned into two colored faces. The colors on the domino faces are used to restrict how the dominoes can be positioned: when the faces of two dominoes are positioned next to each other in a tiling, their colors must coincide. Colored dominoes are similar to *Wang tiles* (1; 13; 9; 20; 4), which are unit squares with colored sides. Wang tiles find application in DNA computing (1; 13) and image processing (7; 11).

Many variants of domino tiling problems have been studied, but most of the results have focused on "colorless" dominoes, which are simply $2 \times 1$ rotateable rectangles (see, for example, (14; 8; 18; 16; 17; 10; 19)). Results concerning colored domino tilings have only arisen relatively recently (21; 5; 22; 23). In the colored domino tiling problems studied in (21) and (5), a multiset of dominoes is provided, and in tilings the multiplicity of the dominoes cannot exceed those in the multiset. In (21), the multiplicity of the provided dominoes equals that of the multiplicity of the dominoes used in the tiling. An algorithm

†Research was supported in part by NSERC under Grant 261290-03.

is described for computing colored domino tilings of so-called "paths" or "cycles". This algorithm runs in time linear in the number of dominoes used in the tiling. The authors of (21) also consider a colored domino tiling problem where some dominoes have already been positioned on the polygon, and we are asked to decide if the tiling can be completed. This problem is shown to be NP-complete. In (5), Biedl studies two variants of a domino tiling problem. In the first problem, known as *EXACT DOMINO TILING*, the multiplicity of dominoes in the provided set is equal to multiplicity of the dominoes used in the tiling. Biedl shows that *EXACT DOMINO TILING* is NP-complete, even for a very restricted class of polygons known as "caterpillars". It is also shown that *EXACT DOMINO TILING* remains NP-complete when the dominoes are restricted to three colors. In the second domino tiling problem studied in (5), which is called *PARTIAL DOMINO TILING*, not all the dominoes are used in the tiling. It is shown that *PARTIAL DOMINO TILING* is NP-complete, and remains so even for so-called "paths" or "cycles". It is also shown that *PARTIAL DOMINO TILING* is NP-complete when the dominoes are restricted to three colors. We also note the result by Beauquier et. al. (3), which proves the NP-completeness for deciding the tileability of a polygon with holes for non-colored dominoes.

In the problems that we study, the set of dominoes used is not a multiset, and each domino has 2 colors and can be used an unlimited number of times. Thus we are concerned with questions related to the following: "Does there exist a set of colored dominoes that can tile a given orthogonal polygon $P$?"

We demonstrate that for simple orthogonal polygons that can be tiled with colored dominoes, two colors are always sufficient. We also show that for tileable non-simple orthogonal polygons, four colors are always sufficient and sometimes necessary. We also describe an $O(n)$ time algorithm for computing a colored domino tiling of a simple orthogonal polygon, where $n$ is the number of dominoes used in the tiling. We also show that deciding whether or not a non-simple orthogonal polygon can be tiled with colored dominoes is NP-complete.

## 2 Definitions

A *colored domino* is a rotatable $2 \times 1$ rectangle that has been partitioned into two colored unit squares, which we refer to as the *faces* of the colored domino. A colored domino is deemed a *twin* if it is monochromatic. Since all dominoes considered herein will be colored, we will often refer to "colored dominoes" simply as "dominoes". All polygons that we consider are orthogonal polygons with integer coordinates. We adopt terminology similar to that found in (21; 5), and refer to such polygons as *layout polygons*. A polygon is deemed simple if it does not contain holes. We refer to the set of unit squares induced by the integer grid within a layout polygon $P$ as the *squares* in $P$, which are denoted by $\rho(P)$. We deem two squares $p, q \in \rho(P)$ to be *adjacent* if $p$ shares an edge with $q$. A *tiling* is a placement of dominoes from a set $D$ of dominoes onto a layout polygon $P$ such that each square in $\rho(P)$ is covered by exactly one domino face and adjacent faces of dominoes have the same color. We say that $p, q \in \rho(P)$ are *matched* under a tiling if $p$ and $q$ are covered by the same domino in the tiling. A tiling of a layout polygon $P$ therefore induces a perfect matching of $\rho(P)$.

Suppose there exists a colored domino tiling $\tau$ of a layout polygon $P$ that uses $k$ colors. We can immediately see that there must also exist a colored domino tiling of $P$ that uses 1 color since we can simply re-color the faces of all the dominoes in $\tau$ to be black. This observation is unsatisfying since the colors on the dominoes becomes irrelevant. Since we are interested in tilings that use colored dominoes, we must disallow twin dominoes. Thus we are only concerned with tilings that do not use twin dominoes. Motivated by this restriction, we define $D_k$ for $k \geq 2$ as follows: $D_k$ is the set of $\binom{k}{2}$ non-twin dominoes over $k$ colors.

**Definition 2.1** ***$k$-tileable:*** *A layout polygon $P$ is $k$-*tileable *if and only if there exists a tiling of $P$ using dominoes from $D_k$.*

Extending definition 2.1 slightly, we define a $k\text{-}tiling$ to be a tiling that uses dominoes from $D_k$. We study the following tiling decision problem:

**Definition 2.2** ***TWINLESS TILING:***
***INSTANCE:*** *A layout polygon $P$.*
***DECISION:*** *Is $P$ $k$-tileable for $k \geq 2$?*

Next we introduce the main tools that we use to analyze colored domino tilings. The following concept will be used extensively throughout the remainder of the paper:

**Definition 2.3** ***Twin-Forcing Arrangement:*** *Let $P$ be a layout polygon with a perfect matching $M$ of $\rho(P)$. A* twin-forcing arrangement *is a set of four squares $p, q, r, s \in \rho(P)$ contained in a $2 \times 2$ subrectangle of $P$, such that $p$ and $q$ are adjacent and matched with each other, while $r$ and $s$ are adjacent, but $r$ is not matched with $s$.*

One can easily show that a twin-forcing arrangement is forbidden in layout polygons that are $k$-tileable for $k \geq 2$ since $p$ and $q$ must be covered with a twin domino. Next we introduce the concepts of *leaves* and *corners* in layout polygons.

**Definition 2.4** ***Leaf:*** *Let $L$ be a $2 \times 1$ subrectangle of $P$. Then $L$ is a* leaf *of $P$ if $L$ contains a square $p$ such that $p$ is adjacent to exactly one square in $\rho(P)$.*

**Definition 2.5** ***Corner:*** *Let $C$ be a $2 \times 2$ subrectangle of $P$. Then $C$ is a* corner *of $P$ if $C$ contains a square $p$ such that $p$ is adjacent to only two other squares in $\rho(P)$.*

The following two Lemmas about leaves and corners will be essential to our algorithm, and will also be useful when we obtain lower bounds on the number of colors needed to tile a non-simple layout polygon.

**Lemma 2.6** *Let $P$ be a $k$-tileable layout polygon for $k \geq 2$ that contains a leaf $L$. In any $k$-tiling of $P$ there is only one domino that covers $L$.*

**Proof:** Let $L$ be a leaf from a layout polygon $P$ that contains the squares $p$ and $q$. W.l.o.g, suppose $p$ is only adjacent to $q$. In any tiling $p$ is matched to an adjacent square, and since $p$ is only adjacent to $q$, the domino that covers $p$ must also cover $q$, as required. □

**Lemma 2.7** *Let $P$ be a $k$-tileable layout polygon for $k \geq 2$ that contains a corner $C$. In any $k$-tiling of $P$ there are only two dominoes that covers $C$.*

**Proof:** Let $C$ be any corner from a layout polygon $P$. Since $C$ is a $2 \times 2$ rectangle, it must contain exactly 4 squares. We will refer to these squares as $p_i$, $p_j$, $p_k$, $p_l$, where $p_i$ is only adjacent to $p_j$ and $p_k$. Since $p_i$ is only adjacent to $p_j$ and $p_k$, $p_i$ must be matched with either $p_j$ or $p_k$ in a tiling of $P$. W.l.o.g, we can suppose that $p_i$ is matched with $p_j$. If $p_k$ is matched with $p_l$ then we have our result, so for the sake of contradiction suppose that $p_k$ is not matched with $p_l$. Then $p_i, p_j, p_k$, and $p_l$ form a twin-forcing arrangement, which is a contradiction. □

A key observation about leaves and corners is their existence in simple layout polygons.

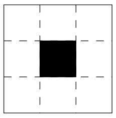


**Fig. 1:** A non-simple layout polygon that does not contain a leaf or a corner.

**Lemma 2.8** *If $P$ is a simple layout polygon such that $|\rho(P)| \geq 2$, then $P$ contains a leaf or a corner.*

**Proof:** Let $p$ be a square from $\rho(P)$ that is incident with a convex vertex $v$ of $P$. Then $p$ is adjacent to at most two other squares from $\rho(P)$, and $p$ must be adjacent to at least one other square since $|\rho(P)| \geq 2$. If $p$ is only adjacent to one other square from $\rho(P)$, then $p$ is part of a leaf, and the Lemma follows.

So suppose $p$ is adjacent to two other squares, say $p_i$ and $p_j$. If $p_i$ and $p_j$ are both adjacent to a common neighbor that is not $p$, then $p$ is part of a corner, and the Lemma follows.

So assume that $p$ is the only common neighbor of $p_i$ and $p_j$. Thus one vertex of $p$ is a convex vertex of $P$, and another vertex of $p$ is a concave vertex of $P$. Now assume there is no leaf or corner. For each convex vertex of $P$ we can associate a unique concave vertex. Let $c$ and $r$ be the number of convex and concave (i.e. reflex) vertices in $P$, respectively. Then we have that $c \leq r$. Since it can easily shown that for any simple orthogonal polygon $c > r$, we arrive at a contradiction and the Lemma follows. $\square$

Notice that Lemma 2.8 only holds for simple layout polygons. Figure 1 illustrates a non-simple layout polygon that does not contain a leaf or a corner.

The next set of definitions provide a connection between graph coloring and colored domino tiling. If $P$ is a layout polygon that has been tiled with colored dominoes, then we refer to the color of the domino face that is covering the square $p \in \rho(P)$ as $color(p)$.

**Definition 2.9** ***Color Partition:*** *Let $P$ be a $k$-tileable layout polygon, and let $\tau$ be a $k$-tiling of $P$. We define the* color partition *of $P$ with respect to $\tau$ to be a partition $P_1, P_2, ..., P_m$ of $P$, such that $P_i$ is a maximal connected set of like-colored squares, for $i = 1, 2, ..., m$.*

The following Lemma will be useful when we consider tiling simple layout polygons.

**Lemma 2.10** *For any $k$-tileable simple layout polygon $P$ with a $k$-tiling $\tau$, the color partition $P_1, P_2, ..., P_m$ implied by $\tau$ is such that $P_i$ is a simple polygon for each $1 \leq i \leq m$.*

**Proof:** For the sake of contradiction, suppose $P_i$ is not a simple polygon, i.e. suppose that $P_i$ contains a hole $H$. Let $u_1, u_2, u_3$ be three consecutive vertices along the boundary of $H$ such that $u_2$ is a reflex vertex with respect to $P_i$. Let $p_1, p_2 \in \rho(P_i)$ be such that $p_1$ has the edge $u_1u_2$, while $p_2$ has the edge $u_2u_3$. Let $p$ be a square in $H$ such that $p$ has edges $u_1u_2$ and $u_2u_3$. Since $P$ is a simple polygon, we know that $p \in \rho(P)$. Then we have $color(p) \neq color(p_1)$ and $color(p) \neq color(p_2)$ by Definition 2.9. Note that for each pair of adjacent squares with different colors, they must be matched in $\tau$. Thus, $p$ is matched with $p_1$ and $p_2$. This contradicts the fact that there is no overlapping dominoes in a $k$-tiling. $\square$

**Definition 2.11** ***Color Partition Graph:*** *Let $P_1, P_2, ..., P_m$ be the color partition associated with a $k$-tiling $\tau$ of a $k$-tileable layout polygon $P$. We define the* color partition graph *to be a graph with vertices $v_1, v_2, ..., v_m$. There is an edge $(v_i, v_j)$, $i \neq j$, in the color partition graph if the partitions $P_i$ and $P_j$ share an edge on their boundaries.*

Notice that if we color the nodes of a color partition graph $G$ according to the colors of the corresponding color partitions, then we have a proper $k$-coloring of $G$. Likewise, a $k$-coloring of $G$ defines a $k$-tiling of the original polygon. Thus a $k$-coloring of a color partition graph corresponds exactly to a $k$-tiling. This is the key observation that is exploited in the following section.

# 3 Colored Domino Tilings and Graph Coloring

In this section we obtain tight bounds on the number of colors needed to tile simple and non-simple layout polygons.

**Definition 3.1** ***Chromatic Number of*** $P$***:*** *The* chromatic number *of a layout polygon $P$, denoted $\chi(P)$, is the minimum number $k \geq 2$ for which $P$ is $k$-tileable.*

We begin by describing a simple result that will be useful in obtaining forthcoming results.

**Lemma 3.2** *In any $k$-tiling of a layout polygon $P$, each vertex of any square in $\rho(P)$ is incident with at most 2 different colors.*

**Proof:** Since the Lemma obviously holds for $k \leq 2$, suppose $k \geq 3$. Suppose that $v$ is a vertex of some square in $\rho(P)$ which is incident with at least 3 different colors. Let $p_1, p_2, p_3 \in \rho(P)$ be three squares which are incident with $v$ and have 3 different colors respectively. Without loss of generality, suppose that $p_1$ is adjacent to $p_2$ and $p_3$. Note that for each pair of adjacent squares in $\rho(P)$ with different colors, they must be matched in the $k$-tiling. Thus, $p_1$ is matched with $p_2$ and $p_3$ . This contradicts the fact that there are no overlapping dominoes in a tiling. □

**Lemma 3.3** *If $P$ is a $k$-tileable simple layout polygon with a $k$-tiling $\tau$, for $k \geq 2$, then the color partition graph $G$ associated with $\tau$ is a tree.*

**Proof:** Since $P$ is a simple layout polygon, it is easy to see that $G$ is connected. Suppose that $G$ has a cycle $v_1 v_2 ... v_a v_1$, $3 \leq a \leq m$. Let $P'$ be the union of $P_1, P_2, ..., P_a$. Now we use Lemma 3.2 to show that there must exist a hole $H$ in $P'$ which encloses at least one square. Let $x_1$ and $x_2$ be the extreme points on the boundary between $P_1$ and $P_2$ (refer to Figure 2(a)). Consider the path $v_2, v_3, ..., v_a, v_1$ implied by the cycle in $G$ (see Figure 2(b)). Notice that the parts along this path, i.e. $P_2, P_3, ..., P_a, P_1$, enclose either $x_1$ or $x_2$. Let $x_i \in \{x_1, x_2\}$ be the point enclosed by this path. Assuming that $P'$ does not contain a hole, $x_i$ must be incident with at least one square that is not contained in either $P_1$ or $P_2$. We have two cases to consider.

Firstly, it may be the case that $x_i$ is incident with two squares $s_1, s_2 \notin P_1 \cup P_2$ (ex. $x_1$ from Figure 2(b)). W.l.o.g., suppose $s_1$ is adjacent to a square from $P_1$, while $s_2$ is adjacent to a square from $P_2$ If $s_1$ and $s_2$ are the same color, then $x_i$ is incident with three colors, contradicting Lemma 3.2. If $s_1$ and $s_2$ are different colors, then $s_1$ must be matched with both $s_2$ and a square from $P_1$, which is a contradiction.

Otherwise $x_i$ is incident with three squares in $P_1 \cup P_2$ (ex. $x_2$ from Figure 2(b)). In this case $x_i$ is incident with three colors since both $P_1$ and $P_2$ are maximally connected sets of like-colored squares. This contradicts Lemma 3.2. We conclude that there exists a hole $H$ in $P'$.

By Lemma 2.10, $H$ is not a hole in any of $P_i$, $1 \leq i \leq a$. Let $u_1, u_2, u_3$ be three consecutive vertices along a path on the boundary of $H$ such that $u_2$ is a reflex vertex of $P'$. Let $p_1, p_2 \in \rho(P')$ be such that $p_1$ has the edge $u_1 u_2$, while $p_2$ has the edge $u_2 u_3$. Let $p \notin \rho(P')$ be such that $p$ has edges $u_1 u_2$ and

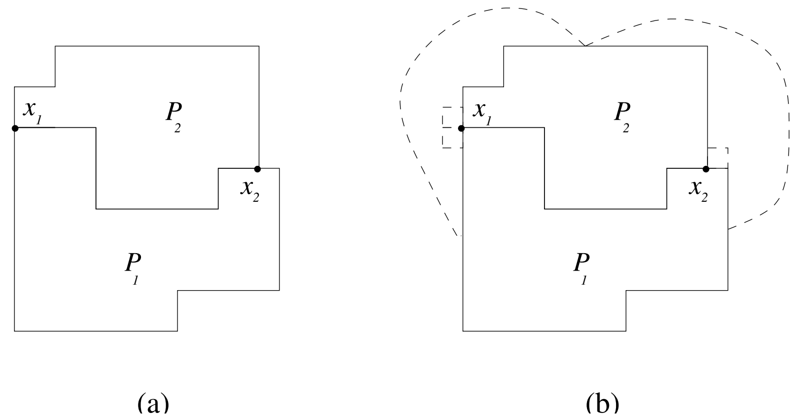



**Fig. 2:** (a) The boundary between $P_1$ and $P_2$. (b) The dashed lines represent the path $v_2, v_3, ..., v_a, v_1$.

$u_2u_3$. Since $P$ is a simple polygon, we know that $p \in \rho(P)$. Then we have $color(p) \neq color(p_1)$ and $color(p) \neq color(p_2)$. Note that for each pair of adjacent squares with different colors, they must be matched in $\tau$. Thus, $p$ is matched with $p_1$ and $p_2$. This contradicts the fact that there is no overlapping dominoes in a tiling. Since $G$ is connected and contains no cycles, $G$ is a tree. □

Since trees are 2-colorable, we have the following:

**Theorem 3.4** *If $P$ is a $k$-tileable simple layout polygon for $k \geq 2$, then $\chi(P) = 2$.*

The following Corollary, which will be useful when considering an algorithm for tiling simple layout polygons, is implied by the 2-colorability of trees and the proof of Lemma 3.3:

**Corollary 3.5** *Let $P$ be a simple layout polygon. $P$ has a $k$-tiling with a perfect matching $M$ of $\rho(P)$ for $k \geq 2$ if and only if $P$ has a 2-tiling with a perfect matching $M$ of $\rho(P)$.*

Now we turn our attention to non-simple layout polygons, i.e. layout polygons that contain one or more holes.

**Lemma 3.6** *There exists a (non-simple) layout polygon such that $k$-tilings can only exist for $k \geq 4$.*

**Proof:** We claim that the layout polygon depicted in Figure 3(a) requires at least four colors in any twin-free colored domino tiling. We begin by identifying all the leaves and corners in this layout polygon (see Figure 3(b)). The identification of the leaves and corners forces certain squares to be matched with each other. These forced matchings are shown in Figure 3(c).

Now we consider how the leftmost-lowest square in $P$, which we will call $l$, is matched (see Figure 4). It is either matched with the neighbor to its right, or $l$ is matched with the neighbor that is above $l$.

**Case(i):** First suppose that $l$ is matched with its neighbor to the right. This forces the set of matchings shown in Figure 4 (a). Now consider how the square labelled as $p$ in Figure 4(a) is matched. It cannot be matched with its right neighbor since the domino involved in such a matching would be forced to be a twin. Thus $p$ must be matched with the neighbor that is above $p$. This forces the matchings in Figure

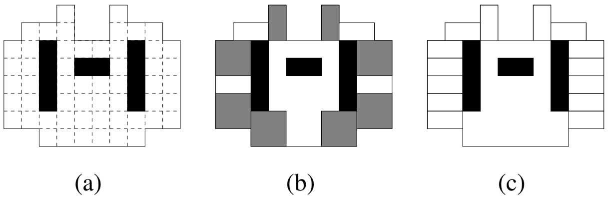



**Fig. 3:** (a) A layout polygon $P$. (b) The leaves and corners of $P$ have been lightly shaded. (c) The matchings that are forced by the leaves and corners of $P$.

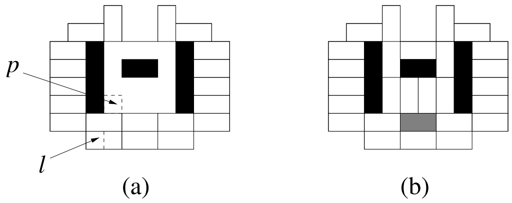



**Fig. 4:** (a) The matchings implied by $l$ being matched with its right neighbor. (b) The matchings implied by $p$ being matched with the neighbor above $p$. A forced twin domino has been shaded.

4(b). We can see that such a matching forces two twin dominoes, one of which is indicated in Figure 4(b). Thus $l$ cannot be matched with its right neighbor.

**Case(ii):** Suppose $l$ is matched with the neighbor that is above $l$. This matching implies other matchings that are shown in Figure 5(a). The corresponding color partition is shown in Figure 5(b). The resulting color partition graph is shown in Figure 5(c). Since $K_4$ is a subgraph of this graph, at least 4 colors are required in any coloring, as claimed. □

If we combine Lemma 3.6 with the fact that any color partition graph of a layout polygon is planar, and is therefore 4-colorable, we obtain the following:

**Theorem 3.7** *Let $P$ be a $k$-tileable layout polygon for $k \geq 2$. Then $\chi(P) \leq 4$, and this bound is tight.*

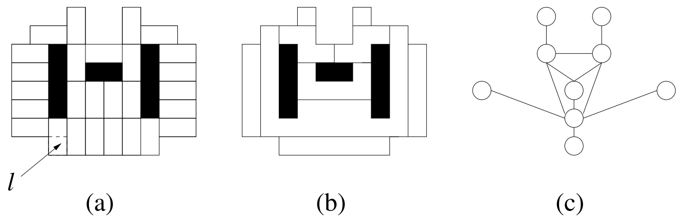



**Fig. 5:** (a) The matchings implied by matching $l$ with the neighbor above $l$. (b) The color partition implied by the matching. (c) The resulting color partition graph contains a $K_4$.

# 4 Tiling Simple Layout Polygons

In this Section we describe an algorithm for the *TWINLESS TILING* problem for simple layout polygons. We note that our algorithm not only decides whether or not a simple layout polygon is $k$-tileable for $k \geq 2$, it also computes such a tiling if one exists. Our algorithm operates on the following decomposition of the layout polygon.

**Definition 4.1** ***Leaf-Corner Decomposition:*** *A* leaf-corner decomposition *of a layout polygon $P$, denoted $\mathcal{L}(P)$, is defined recursively as follows: if $P$ does not contain a leaf or a corner, then $\mathcal{L}(P) := \emptyset$. Otherwise, let $X$ be a leaf or corner of $P$, and let $P_1, P_2, ..., P_l$ be the simple layout polygons obtained by removing $X$ from $P$. Then we define $\mathcal{L}(P) := \{X\} \bigcup \mathcal{L}(P_1) \bigcup \mathcal{L}(P_2) \bigcup ... \bigcup \mathcal{L}(P_l)$.*

We pause here to consider an important note concerning the members of a leaf-corner decomposition $\mathcal{L}(P)$. A member of $\mathcal{L}(P)$ is not necessarily a leaf or a corner of $P$, although each member of $\mathcal{L}(P)$ is a leaf or a corner of some subpolygon of $P$. For convenience, we will refer to the members of $\mathcal{L}(P)$ as leaves or corners. We can justify this naming convention by observing that Lemma 2.6 and Lemma 2.7 hold for the members of $\mathcal{L}(P)$:

**Lemma 4.2** *Let $P$ be a simple layout polygon that is $k$-tileable for $k \geq 2$, and let $\mathcal{L}(P)$ be a leaf-corner decomposition of $P$. If $L \in \mathcal{L}(P)$ is a $1 \times 2$ rectangle then in any $k$-tiling of $P$ for $k \geq 2$, there is only one domino that covers $L$. Furthermore, if $C \in \mathcal{L}(P)$ is a $2 \times 2$ rectangle then in any $k$-tiling of $P$, for $k \geq 2$, there are only two dominoes that cover $C$.*

**Proof:** We induct on $|\rho(P)|$. The Lemma is trivially true for $|\rho(P)| \leq 4$. So assume that the Lemma is true for $|\rho(P)| \leq 2d$.

Let $P$ be a simple layout polygon with $|\rho(P)| = 2(d+1)$ squares that is $k$-tileable for $k \geq 2$. Let $\mathcal{L}(P)$ be a leaf-corner decomposition of $P$, and let $X$ be the leaf or corner mentioned in Definition 4.1 that is used to define $\mathcal{L}(P)$. By induction, the Lemma holds for each of the simple layout polygons obtained by removing $X$ from $P$. Since $X$ is a leaf or a corner of $P$, Lemma 2.6 or Lemma 2.7 implies the result. □

We say that a leaf-corner decomposition $\mathcal{L}(P)$ *covers* $P$ if and only if the union of all the members in $\mathcal{L}(P)$ is $P$.

**Lemma 4.3** *Given a simple layout polygon $P$ and a leaf-corner decomposition $\mathcal{L}(P)$, if $\mathcal{L}(P)$ does not cover $P$ then $P$ is not $k$-tileable for $k \geq 2$.*

**Proof:** Let $\mathcal{L}(P)$ be a leaf-corner decomposition of $P$. Suppose that $P$ is $k$-tileable even though $\mathcal{L}(P)$ does not cover $P$. Define $\mathcal{P} := \{P_1, P_2, ..., P_l\}$ to be the set of simple layout polygons obtained by removing $\mathcal{L}(P)$ from $P$. By Definition 4.1, we have that each $P_i \in \mathcal{P}$ does not contain a leaf or a corner. It follows from Lemma 2.8 that each $P_i \in \mathcal{P}$ is not $k$-tileable for $k \geq 2$. But since $P$ is $k$-tileable, and by Lemma 4.2, each $P_i \in \mathcal{P}$ is also $k$-tileable, and we arrive at a contradiction. □

Due to the following result, the main objective of our algorithm is to compute a perfect matching of $\rho(P)$ that does not contain any twin-forcing arrangements.

**Lemma 4.4** *Let $P$ be a simple layout polygon. $P$ is $k$-tileable with matching $M$, for $k \geq 2$, if and only if $M$ is a perfect matching of $\rho(P)$ that does not contain a twin-forcing arrangement.*

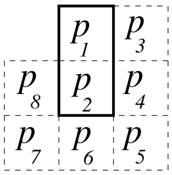



**Fig. 6:** The squares surrounding the matched squares $p_1$ and $p_2$.

**Proof:** ($\Rightarrow$) If $P$ is $k$-tileable, for $k \geq 2$, then a $k$-tiling of $P$ defines a perfect matching of $\rho(P)$. Since $k \geq 2$, this matching does not contain any twin-forcing arrangements.

($\Leftarrow$) We proceed by induction on $|\rho(P)|$. The sufficiency of the Lemma can easily be shown for simple layout polygons with $|\rho(P)| \leq 4$. Assume the sufficiency of the Lemma for all simple layout polygons $P$ such that $|\rho(P)| \leq 2d$ squares.

Let $P$ be a simple layout polygon with $2(d + 1)$ squares. Suppose there exists a perfect matching $M$ of $\rho(P)$ that does not contain any twin-forcing arrangements. Let $p_1$ be a square from $\rho(P)$ that is incident with a convex vertex of $P$. Let $p_2$ be the square that $p_1$ is matched with, and w.l.o.g assume $p_2$ is below $p_1$. Define $P_1, P_2, ..., P_l$ to be the set of mutually edge disjoint simple layout polygons obtained by removing $p_1$ and $p_2$ from $P$. For the remainder of the proof we will refer to $P_1, P_2, ..., P_l$ collectively as the *components*. Define $M_i$ to be the matching of $P_i$ that is implied by $M$, for $1 \leq i \leq l$. Since $|\rho(P_i)| \leq 2d$, by induction we have that $P_i$ is $k$-tileable with matching $M_i$, for $k \geq 2$ and $1 \leq i \leq l$. Furthermore, by Corollary 3.5, each $P_i$ is 2-tileable with matching $M_i$. So let us tile the portion of $P$ that is covered by $P_1, P_2, ..., P_l$ using the guaranteed 2-tilings that exist for each of these components. To complete the tiling of $P$, we must cover $p_1$ and $p_2$ with a multicolored domino.

Now let us consider the squares that may surround $p_1$ and $p_2$ (see Figure 6). Notice that not all of $p_3, ..., p_8$ are necessarily in $P$. Also, if $p_i$ and $p_j$ are in different components, then inverting the colors in $p_i$'s component (i.e. changing white squares to black, and black squares to white) has no effect on the colors in $p_j$'s component.

Cover $p_1$ and $p_2$ with a non-twin domino, ensuring that this domino's colors coincide with $color(p_3)$ and $color(p_4)$ if either or both of $p_3$ or $p_4$ are in $P$. The fact that $M$ does not contain twin-forcing arrangements guarantees that such a non-twin domino exists. If both of $p_3$ and $p_4$ are not in $P$ then cover $p_1$ and $p_2$ with a non-twin domino according to the color of $p_6$.

Now suppose $color(p_6) \neq color(p_2)$. We can remedy this by inverting the colors in the component that contains $p_6$. This inversion does not change $color(p_3)$ and $color(p_4)$ if $p_6$ and $p_4$ are in different components. So suppose $p_6$ is in the same component as $p_4$. We have that $p_5 \in \rho(P)$, otherwise $p_5$ is a hole, or part of a hole, in $P$. Since $color(p_2) = color(p_4)$, we have $color(p_6) \neq color(p_4)$. Therefore, the color of $p_5$ is different to at least one of $p_6$ or $p_4$, and must be matched with that one, but this also causes a twin-forcing arrangement involving $p_2, p_4, p_5$, and $p_6$, which is a contradiction. We conclude that $color(p_6) = color(p_2)$.

Similarly, we can argue that $color(p_8) = color(p2)$, or the color of $p_8$ can be inverted without changing $color(p_6), color(p_4)$, and $color(p_3)$. Thus $P$ can be $k$-tiled for $k \geq 2$, as required. □

Let $L$ be a leaf, and $C$ be a corner, from a leaf-corner decomposition $\mathcal{L}(P)$ of a simple layout polygon $P$. According to Lemma 4.2, the squares in $L$ must be matched, and the squares contained $C$ may be matched one of two ways, in any $k$-tiling of $P$, for $k \geq 2$. Suppose we are given a perfect matching

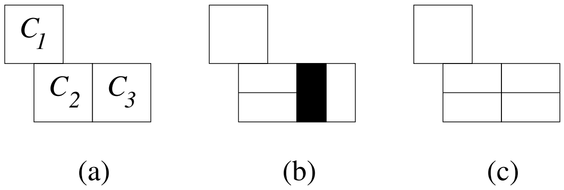



**Fig. 7:** A matching that forces another matching. (a) Three label corners. (b) Matchings of $C_2$ and $C_3$ that cause of twin-forcing arrangement. (c) A matching of $C_2$ that forces a matching of $C_3$.

of $\rho(P)$ that is associated with a $k$-tiling of $P$ for $k \geq 2$, and let $p$ be the highest and leftmost square in $C$. If $p$ is matched with its right neighbor, then we say that $C$ is matched "horizontally", and deem $C$ matched "vertically" otherwise. The key to our algorithm is noticing that the matching of a leaf or a corner may "force" neighboring leaves and corners to be matched a certain way if we are to avoid twin-forcing arrangements. To illustrate this, consider the corners $C_1, C_2$, and $C_3$ from the leaf-corner decomposition of a layout polygon depicted in Figure 7(a). Suppose the corner $C_2$ is matched as in Figure 7(b). We can see that $C_1$ can be matched in either of two ways without causing a twin-forcing arrangement. This is not the case for $C_3$. If $C_3$ is matched "vertically" (see Figure 7(b)) then a twin-forcing arrangements arises. This forces $C_3$ to be matched "horizontally" as in Figure 7(c).

We can generalize these observations in the following Lemma. We say that two members $X$ and $Y$ of a leaf-corner decomposition are *adjacent* if there exists two squares $p, q \in \rho(P)$, such that $p \in X$, $q \in Y$, and $p$ and $q$ are adjacent. Also, an *edge* of a leaf or corner $x$ is an edge from the bounding rectangle of $x$.

**Lemma 4.5** *Let $P$ be a simple layout polygon with a perfect matching $M$ of $\rho(P)$, and a leaf-corner decomposition $\mathcal{L}(P)$ that covers $P$. $\mathcal{L}(P)$ satisfies the following rules with respect to $M$ if and only if $M$ does not contain a twinforcing arrangement:*

1. *For each edge $e$ of each leaf $L \in \mathcal{L}(P)$, there is at most one member of $\mathcal{L}(P)$ that both intersects $e$ and is adjacent to $L$.*
2. *If $C_i, C_j \in \mathcal{L}(P)$ are two corners that share an entire edge, then $C_i$ and $C_j$ are both matched horizontally or they are both matched vertically.*
3. *If $L, C_i \in \mathcal{L}(P)$ are a leaf and a corner such that $L$ shares an entire vertical (resp. horizontal) edge with $C_i$, then $C_i$ is matched vertically (resp. horizontally).*
4. *If $X, Y \in \mathcal{L}(P)$ are both adjacent to a corner $C_i \in \mathcal{L}(P)$, such that both $X$ and $Y$ intersect the same vertical (resp. horizontal) edge of $C_i$, then $C_i$ is matched horizontally (resp. vertically).*

**Proof:** ($\Rightarrow$) Suppose for the sake of contradiction that $M$ contains a twinforcing arrangement. Let $p, q, r$, and $s$ be four squares that form a twinforcing arrangement, such that $p$ and $q$ are matched, while $r$ and $s$ are unmatched. Lemma 4.2 implies that since $p$ and $q$ are matched, they are either both contained in the same leaf or both contained in the same corner from $\mathcal{L}(P)$. First suppose $p$ and $q$ are both contained in the leaf $L \in \mathcal{L}(P)$. If $r$ and $s$ are contained in different members of $\mathcal{L}(P)$, then rule (1) is violated. So $r$ and $s$ are contained in the same member of $\mathcal{L}(P)$, and since $r$ and $s$ are not matched, they must be contained in a corner. But rule (3) implies that $r$ and $s$ are matched, which is a contradiction. So suppose $p$ and $q$

are both contained in the corner $C_1 \in \mathcal{L}(P)$, and w.l.o.g. assume that $C_1$ is matched vertically. If $r$ and $s$ are both contained in the corner $C_2 \in \mathcal{L}(P)$, then by rule (2), $C_2$ is matched vertically, which implies $r$ and $s$ are matched, which is a contradiction. So $r$ and $s$ must be contained in different members of $\mathcal{L}(P)$. This contradicts rule (4), and hence we conclude that $M$ does not contain any twinforcing arrangements.

($\Leftarrow$) Suppose for the sake of contradiction that one of rules (1)-(4) is does not hold for $\mathcal{L}(P)$. If rule (1) is violated then there exists two unmatched squares that are each adjacent to a different square in $L$, and since the squares in $L$ are necessarily matched by Lemma 4.2, a twin-forcing arrangement is formed, which is a contradiction. If rule (2), (3), or (4) is violated then there is a vertically (resp. horizontally) matched pair of squares next two unmatched squares, and a twin-forcing arrangement is formed, which implies a contradiction. □

Now we can describe our algorithm for solving the *TWINLESS TILING* problem for simple layout polygons. Before considering our algorithm, we make an important note concerning the input to our algorithm. The time taken by algorithms that operate on polygons is typically expressed in terms of the number of vertices that the polygon has. This presents a problem for the *TWINLESS TILING* problem since we can describe a layout polygon with $O(1)$ vertices that contains an arbitrary number of squares. Although we can deal with this problem in a number of different ways, we adopt the following assumption: the layout polygon $P$ is described by $\rho(P)$ in the input in the form of an adjacency list of squares and neighbors. Using this assumption, we can access the neighbors of each square in $O(1)$ time since each square has at most four neighbors.

**Theorem 4.6** *The* TWINLESS TILING *problem can be solved in $O(|\rho(P)|)$ time on simple layout polygons, or equivalently, $O(n)$ time, where $n$ is the number of dominoes needed in the tiling.*

**Proof:** The algorithm begins by constructing a leaf-corner decomposition $\mathcal{L}(P)$ of $P$. If $\mathcal{L}(P)$ does not cover $P$ then the algorithm halts and reports that $P$ is not $k$-tileable for $k \geq 2$. The correctness of this step of the algorithm is guaranteed by Lemma 4.3.

Then the algorithm uses rules (1)-(4) of Lemma 4.5 to construct a boolean expression $\phi$, which is an instance of the 2SAT problem. For each corner $C_i \in \mathcal{L}(P)$ we create two variables $h_i$ and $v_i$, which correspond to $C_i$ being matched horizontally and vertically, respectively. In order to ensure that exactly one matching of $C_i$ is chosen, we add the following clauses to $\phi$: $(h_i \vee v_i)$ and $(\neg h_i \vee \neg v_i)$. If rule (1) is violated for any leaf in $\mathcal{L}(P)$, we add the unsatisfiable clause $(false)$ to $\phi$. We encode rules (2)-(4) as clauses by inspecting the neighbors of each leaf and corner in $\mathcal{L}(P)$. If the condition of rule (2) is satisfied, then we add the following clauses to $\phi$: $(\neg h_i \vee h_j)$, $(\neg h_j \vee h_i)$, $(\neg v_i \vee v_j)$, and $(\neg v_j \vee v_i)$. If the condition of rule (3) is satisfied, we add the following clause to $\phi$: $(v_i)$ (resp. $(h_i)$). For rule (4) we conditionally add the following clause: $(h_i)$ (resp. $(v_i)$).

By construction, $\phi$ is satisfiable if and only if $\mathcal{L}(P)$ conforms to rules (1)-(4) of Lemma 4.5. Furthermore, a satisfying assignment of $\phi$ corresponds exactly to a perfect matching $M$ of $\rho(P)$, and by Lemma 4.5, $M$ does not contain a twin-forcing arrangement. Thus by Lemma 4.4, $\phi$ is satisfiable if and only if $P$ is $k$-tileable. Thus we can decide if $P$ is $k$-tileable for $k \geq 2$ by solving the 2SAT problem on $\phi$.

To construct a tiling of $P$, we use Corollary 3.5 to observe that $M$ is a matching of some 2-tiling. We compute the colors of the dominoes in a 2-tiling of $P$ as follows. We color an arbitrary square $p \in \rho(P)$ using the color black. Then we proceed using Depth-First Search (DFS), only visiting those squares that are uncolored. When we visit a square $q$, we assign a color to $q$ based upon the color of a neighboring square: if a neighbor is colored black, and $q$ is matched with this neighbor, then $q$ is colored white, otherwise $q$ is colored black. Since DFS is used to visit the squares, each square is assigned a color, and we are guaranteed that no twin dominoes arise since $M$ is the matching of a 2-tiling.

$\mathcal{L}(P)$ can be computed in $O(|\rho(P)|)$ time as follows. First we identify all leaves and corners in $P$ by examining each square and its neighbors. Place each of these leaves and corners in a list called $T$. Notice that some members of $T$ may overlap. Then we iteratively remove leaves and corners from $P$ until $T$ is empty:

1. Remove the next item from $T$ and call it $X$.
2. If $X$ is still completely contained in $P$ then add $X$ to $\mathcal{L}(P)$.
3. $P = P - X$.
4. Add to $T$ all new leaves and corners that are created by removing $X$.

Notice that Step 2 of this procedure is conditional since some members of $T$ may overlap. This procedure stops after $O(|\rho(P)|)$ steps, and each step can be accomplished in $O(1)$ time. Computing $\phi$ can clearly be accomplished in $O(|\rho(P)|)$ time by examining the $O(1)$ neighbors of each leaf and corner from $\mathcal{L}(P)$. Notice that $\phi$ has $O(|\rho(P)|)$ clauses. Thus a satisfying assignment of $\phi$ can be computed in $O(|\rho(P)|)$ time using the algorithm from (2). Assigning colors to the squares in $\rho(P)$ uses DFS and hence can be accomplished in $O(|\rho(P)|)$ time. □

# 5 Tiling Non-Simple Layout Polygons

In this Section we show that the *TWINLESS TILING* problem is NP-complete for non-simple layout polygons, i.e. layout polygons that contain holes. This problem is in NP since we can "guess" a tiling and verify that this tiling is a $k$-tiling for $k \geq 2$ in non-deterministic polynomial time.

Our reduction is from a boolean satisfiability problem called *PLANAR 3,4SAT*. Let $\phi$ be a boolean expression in conjunctive normal form. We define the *inclusion graph* of $\phi$, denoted $G(\phi)$, to be the graph whose vertices are the variables and clauses of $\phi$. An edge $(v_i, c_j)$ exists in $G(\phi)$ if and only if the variable $v_i$ appears negated or unnegated in clause $c_j$.

**Definition 5.1** PLANAR 3,4SAT
***INSTANCE:*** *A boolean expression $\phi$ in conjunctive normal form such that:*

*(i) Each clause contains exactly 3 literals,*

*(ii) Each variable appears negated or unnegated at most 4 times in $\phi$, and*

*(iii) The inclusion graph $G(\phi)$ is planar.*

***DECISION:*** *Is $\phi$ satisfiable?*

The PLANAR 3,4SAT problem is known to be NP-complete (see (12) and (15)). Our reduction will construct a layout polygon $P$ that is $k$-tileable for $k \geq 2$ if and only if $\phi$ is satisfiable. To accomplish this, we will use a planar embedding of $G(\phi)$ as a template for $P$. Specifically, we will use a *planar orthogonal grid drawing* of $G(\phi)$, which we denote as $\mathcal{D}(\phi)$, which has the following properties:

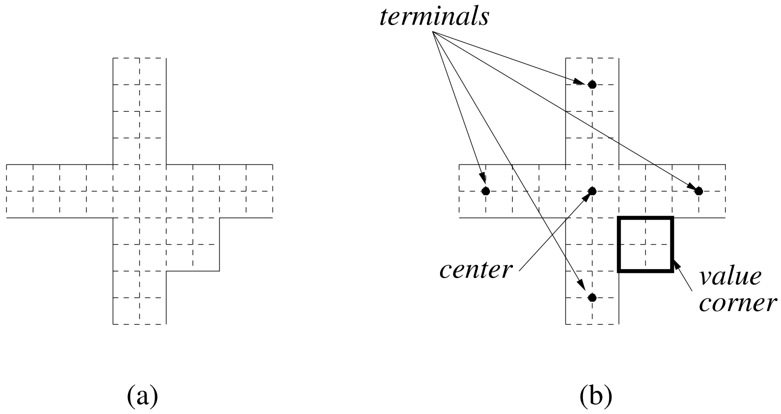



**Fig. 8:** (a) The variable gadget. (b) The terminals, center, and value corner of a variable gadget.

(i) Each edge of $G(\phi)$ is drawn as a chain of orthogonal line segments,

(ii) Each bend in the drawing of an edge lies at an integer valued coordinate,

(iii) Each node of $G(\phi)$ drawn as a point that is located at an integer valued coordinate, and

(iv) All edges are disjoint except when their endpoints meet at a node.

We maintain the graph theoretic terminology when referring to $\mathcal{D}(\phi)$. For example, when we say "an edge $e$ in $\mathcal{D}(\phi)$" we are referring to the chain of orthogonal line segments that constitute the drawing of the edge $e$ in $G(\phi)$. Similarly, when we say "the vertex $v$ in $\mathcal{D}(\phi)$" we are referring to the point in the integer plane associated with the vertex $v$ in $G(\phi)$.

As already noted, $\mathcal{D}(\phi)$ will be used as a template for $P$. So that $|\rho(P)|$ is not too large, we must ensure that the edges in $\mathcal{D}(\phi)$ are not too long. It is known that each edge in $\mathcal{D}(\phi)$ has a length in $O(n^2)$ if $\mathcal{D}(\phi)$ occupies $O(n^2)$ space, where $n$ is the number of variables in $\phi$. Such a planar orthogonal grid drawing can be computed in $O(n)$ time (see, for example, (6)). We note here that as part of our reduction, $\mathcal{D}(\phi)$ will also be scaled by a constant in order to ensure that $P$ can be constructed correctly. The exact value of this constant will be discussed in what follows.

To construct $P$, we will overlay various gadgets over the nodes and edges in $\mathcal{D}(\phi)$. These gadgets, which are layout polygons, will correspond to different aspects of $\phi$. We employ gadgets that represent variables, negation, and clauses. We also make use of a special gadget called a *spacer* that is used to ensure proper connections between variable gadgets and clause gadgets.

**Variable Gadget:** A variable gadget represents a particular variable from $\phi$. The tiling of a variable gadget will correspond to the variable being assigned a particular truth value. The variable gadget is depicted in Figure 8(a). Figure 8(b) identifies important aspects of the variable gadget. The *terminals* of a variable gadget will be used to attach a variable gadget to the appropriate clause gadgets. The *center* will be used to position a variable gadget on the appropriate node in $\mathcal{D}(\phi)$. Specifically, if a variable gadget is representing a variable $v_i$, then the variable gadget is positioned in the plane so its center is located at the node associated with $v_i$ in $\mathcal{D}(\phi)$. The *value corner* is used to "set" the truth value of variable. Specifically,

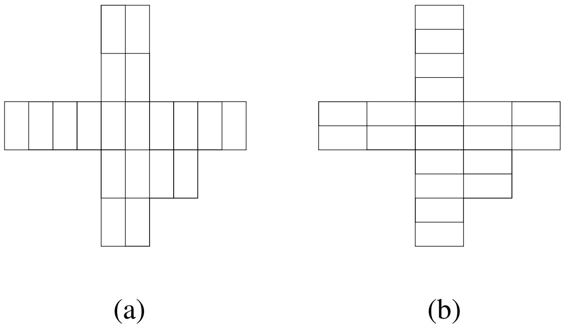



**Fig. 9:** (a) The variable gadget matched vertically representing "false". (b) The variable gadget matched horizontally representing "true".

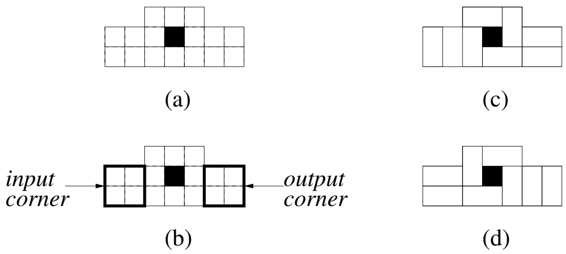


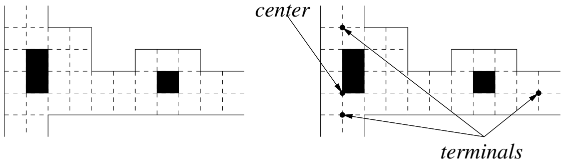

**Fig. 10:** (a) The negation gadget. (b) The input and output corners. (c)(d) The two possible matchings of the negation gadget in a $k$-tiling for $k \geq 2$.

the orientation (i.e. horizontal or vertical) of the matching of the value corner with respect to a $k$-tiling for $k \geq 2$ will determine the truth value of the variable. Figure 9 depicts the two possible ways that a variable gadget can be matched so that a twin-forcing arrangement does not arise. Also, one can see that both of these matchings have a corresponding $k$-tiling for $k \geq 2$. Thus we have the following:

**Lemma 5.2** *In a $k$-tiling of a variable gadget for $k \geq 2$, the value corner of a variable gadget is matched horizontally (resp. vertically) if and only if all squares in the variable gadget are also matched horizontally (resp. vertically).*

We use these two possible matchings of a variable gadget to represent the two possible truth values of the variable that the gadget represents. If the variable gadget is matched vertically, then the variable is set to "false", and if the variable gadget is matched horizontally, then the variable is set to "true".

**Negation Gadget:** The "truth" value of a variable gadget will be transmitted to a clause gadget according to the orientation of the tiling of the variable gadget. When a variable appears negated in a clause, we must "flip" this orientation. The gadget that accomplishes this is called the negation gadget and is depicted in Figure 10(a). Figure 10(b) identifies two important regions of the negation gadget called the *input corner* and the *output corner*. Figure 10(c) and (d) illustrates the fact that the orientation of the matching associated with a $k$-tiling for $k \geq 2$ of the input corner is always opposite to that of the output corner. Thus we have demonstrated the following:




(a) (b)

**Fig. 11:** (a) The clause gadget. (b) The terminals and center of the clause gadget.

**Lemma 5.3** *In any $k$-tiling of a negation gadget for $k \geq 2$, if the input corner of the negation gadget is matched horizontally (resp. vertically), then the output corner is matched vertically (resp. horizontally).*

**Clause Gadget:** A clause gadget is used to represent a clause from $\phi$ (see Figure 11(a)). Figure 11(b) identifies important aspects of the clause gadget. The *terminals* in a clause gadget are used to connect a clause gadget to the appropriate variable gadgets. The *center* of the clause-gadget is used to position a clause gadget on the appropriate vertex in $\mathcal{D}(\phi)$. Specifically, if a clause gadget is representing a clause $c_i$, then the clause gadget is positioned in the plane so that the center is located at the vertex associated with $c_i$. The clause gadget is then rotated if necessary so that each of the terminals intersect a line segment that is incident with the node associated with $c_i$.

The clause gadget is constructed such that it is $k$-tileable for $k \geq 2$ if and only if at least one of the variable gadgets that is connected to it is tiled horizontally, i.e. set to true. Before considering the details concerning how we connect a variable gadget to a clause gadget, let us consider how we represent satisfiability in a clause gadget. Figures 12(a)-(h) depict all the possible matchings of the clause gadget that do not create twin-forcing arrangements. We can see that no twin dominoes need to be used in the matchings in Figures 12(a)-(g). Notice that in each of these Figures (i.e. (a)-(g)), at least one of the $2 \times 2$ squares that surround the terminals is matched horizontally, i.e. in the "true" orientation. Now consider Figure 12(h). In this Figure, each of the $2 \times 2$ squares surrounding the terminals are matched vertically, i.e. in the "false" orientation. Although Figure 12(h) does not contain a twin-forcing arrangement, one can show that there is no $k$-tiling for $k \geq 2$ that uses this matching as follows: Figure 12(i) depicts the matching from Figure 12(h) with ten squares emphasized. These squares must have the same color in any $k$-tiling for $k \geq 2$ that uses this matching. Since two of these squares are matched, a twin domino must be used, and hence the matching depicted in Figure 12(h) cannot be used by any $k$-tiling for $k \geq 2$. The following summarizes these observations:

**Lemma 5.4** *In any $k$-tiling for $k \geq 2$ of a clause gadget, at least one of the $2 \times 2$ squares that surround the terminals must be tiled horizontally, i.e. in the "true" orientation.*

Before describing the final gadget, we consider a partial construction of $P$ using the techniques that have been discussed thus far, and we describe how we connect variable gadgets to clause gadgets. Our example will demonstrate the need for the final gadget.

Suppose we define $\phi := (v_1 \vee v_2 \vee \neg v_3) \wedge (\neg v_1 \vee v_2 \vee v_3)$. Figure 13(a) depicts a sample planar orthogonal grid drawing of $G(\phi)$, which we denote as $\mathcal{D}(\phi)$. Figure 13(b) shows the appropriate variable and clause gadgets positioned on $\mathcal{D}(\phi)$. Consider the chain of orthogonal line segments that represent the edge between $v_2$ and $c_1$ in $\mathcal{D}(\phi)$. We can see that the second segment along this chain intersects the

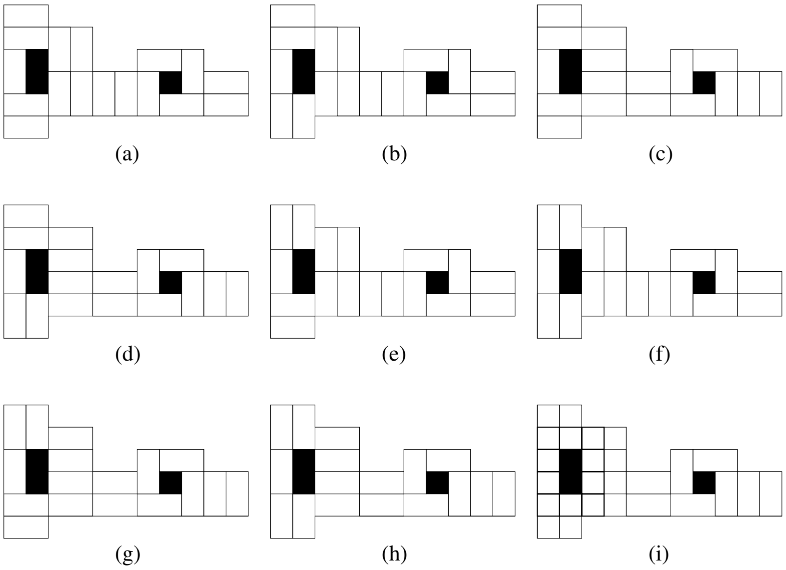



**Fig. 12:** The matchings of the clause gadget that do not contain a twin-forcing arrangement.

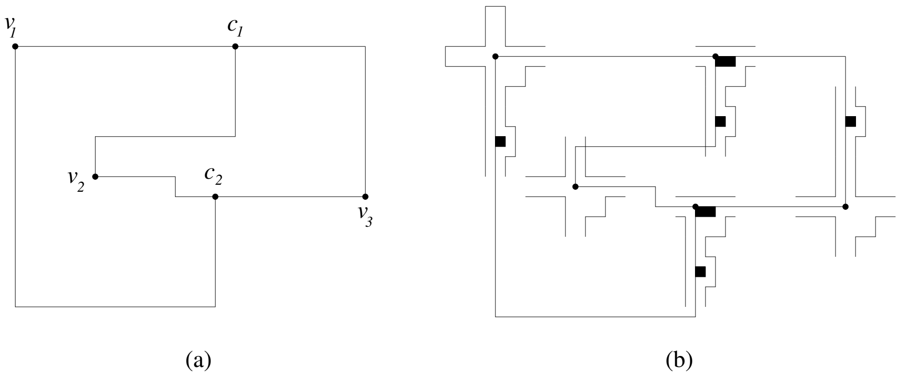
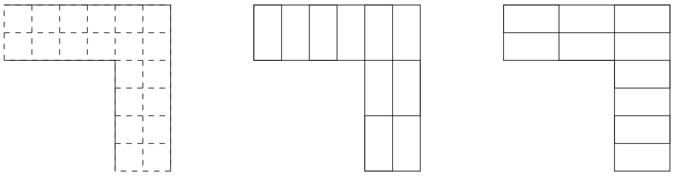



**Fig. 13:** (a) A sample planar orthogonal grid drawing $\mathcal{D}(\phi)$. (b) The variable and clause gadgets positioned on $\mathcal{D}(\phi)$.



(a) (b) (c)

**Fig. 14:** (a) An example of a sequence of $2 \times 2$ groups of squares used to connect a variable gadget to a clause gadget. (b) A sequence of squares transmitting "false". (c) A sequence of squares transmitting "true".

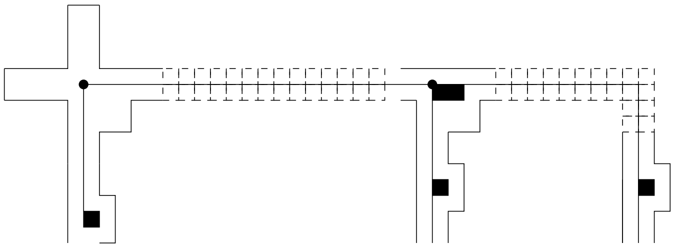


**Fig. 15:** A detailed view of the top portion of $\mathcal{D}(\phi)$.

variable and clause gadget associated with $v_2$ and $c_1$. Thus certain line segments in $\mathcal{D}(\phi)$ may be too short to properly position the gadgets. Although not illustrated in Figure 13(b), disjoint line segments in $\mathcal{D}(\phi)$ may be so close together that two or more gadgets may intersect. We can remedy both of these problems by first scaling $\mathcal{D}(\phi)$ by a constant $a$ so that the boundary of each gadget is disjoint from $\mathcal{D}(\phi)$, and so that no gadgets intersects. This scaling should also take into account the possible need for negation gadgets, as well as the final gadget that we have yet to describe. Although we do not give an explicit value for the scaling constant $a$, we can take $a$ be the sum of the longest side of the orthogonal bounding box for each gadget to ensure that $\mathcal{D}(\phi)$ is large enough to accommodate the gadgets. Since we are scaling $\mathcal{D}(\phi)$ by a constant, $\mathcal{D}(\phi)$ still occupies $O(n^2)$ space in the plane.

The last issue we discuss motivates the need for the final gadget. To connect a variable gadget to the appropriate clause gadget(s), we will use a sequence of $2 \times 2$ groups of squares. Figure 14 illustrates this idea. Notice that the orientation of the matching in any $k$-tiling for $k \geq 2$ is the same throughout the sequence of $2 \times 2$ squares, even if the sequence contains a bend. This ensures that the proper "truth" value is transmitted from the variable gadget to the clause gadget. These squares are added along the line segments that represent the edge between the variable and the clause in $\mathcal{D}(\phi)$. Figure 15 depicts a detailed view of the top portion of Figure 13(a). Collections of $2 \times 2$ groups of squares has been used in an attempt to connect variable gadgets to clause gadgets. This works fine for the connection between $v_3$ and $c_1$, but does not work for the connection between $v_1$ and $c_1$. This is because the distance between the appropriate terminals for the gadgets representing $v_1$ and $c_1$ is an odd number. We cannot complete the connection between $v_1$ to $c_1$ with a $2 \times 1$ group of squares since then the truth value cannot be transmitted properly:

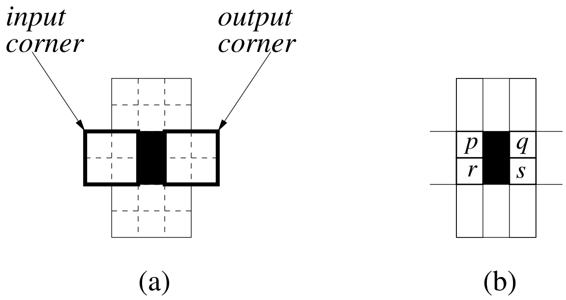



**Fig. 16:** (a) The spacer gadget with the input and output corners identified. (b) The forced matchings of the spacer gadget.

if we simply insert a $2 \times 1$ group of squares to complete the connection between $v_1$ and $c_1$, then the only truth value that can correctly be transmitted is "false", i.e. the connection between $v_1$ and $c_1$ must be tiled vertically. We cannot simply scale $\mathcal{D}(\phi)$ by a factor of 2 to remedy this problem because we need odd distances between centers of variable gadgets and clauses gadgets. We can however remedy this problem by inserting a *spacer* gadget, which is depicted in Figure 16(a).

Now we show that the spacer gadget is an identity gadget in the sense that the input corner will always be matched with the same orientation as the output corner in any $k$-tiling for $k \geq 2$. Figure 16(b) depicts a set of matchings that clearly must exists in a $k$-tiling for $k \geq 2$. Four squares from the spacer gadget have also been labeled $p, q, r$, and $s$. Due to the forced matchings in the spacer gadget, we can observe that $p$ and $q$ (resp. $r$ and $s$) must have the same color in any $k$-tiling of the spacer gadget for $k \geq 2$. This implies the following:

**Lemma 5.5** *In any $k$-tiling of a spacer gadget for $k \geq 2$, the input and output corners are either both matched horizontally or both vertically.*

This Lemma is only useful once we notice another important feature of the spacer gadget: the distance between the input and output corners is odd. Thus spacer gadgets can be inserted on line segments in $\mathcal{D}(\phi)$ when necessary in order to ensure that we can properly connect variable and clause gadgets.

From our construction we can see that $P$ is $k$-tileable if and only if $\phi$ is satisfiable. Thus we have the following:

**Theorem 5.6** *The* TWINLESS TILING *problem is NP-complete for non-simple layout polygons.*

# 6 Open Problems

We have characterized the so-called chromatic number, denote $\chi(P)$, of layout polygons, which is the minimum number of colors needed in a colored domino tiling. We showed that for each $k$-tileable simple layout polygon $P$, $\chi(P) = 2$, for $k \geq 2$. For each $k$-tileable non-simple layout polygon $P$, we showed that $\chi(P) \leq 4$, for $k \geq 2$. We also showed that this bound is tight in the sense that there exists a non-simple layout polygon $P$ such that $\chi(P) = 4$. We are interested in characterizing those layout polygons that require at most three colors in any $k$-tiling for $k \geq 2$. Also, we are interested in the computational complexity of tiling such layout polygons.

## Acknowledgements

We would like to thank Therese Biedl for useful comments regarding the results presented in Section 4.

## References

[1] L. Adleman, J. Kari, L. Kari, and D. Reishus. On the decidability of self-assembly of infinite ribbons. *Proceedings of FOCS 2002, IEEE Symposium on Foundations of Computer Science*, pages 530–537, 2002.

[2] B. Aspvall, M. F. Plass, and R. E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Inf. Process. Lett.*, 8(3):121–123, 1979.

[3] D. Beauquier, M. Nivat, E. Rémila, and M. Robson. Tiling figures of the plane with two bars. *Computational Geometry*, 5:1–25, 1995.

[4] R. Berger. The undecidability of the domino problem. *Memoirs Amer. Math. Soc.*, 66, 1966.

[5] T. Biedl. The complexity of domino tiling. In *Proceedings of the 17th Canadian Conference on Computational Geometry (CCCG'05)*, pages 187–190, 2005.

[6] T. Biedl and G. Kant. A better heuristic for orthogonal graph drawings. *Comput. Geom. Theory Appl.*, 9(3):159–180, 1998.

[7] M. F. Cohen, J. Shade, S. Hiller, and O. Deussen. Wang tiles for image and texture generation. *ACM Trans. Graph.*, 22(3):287–294, 2003.

[8] G. Csizmadia, J. Czyzowicz, L. Gasieniec, F. Kranakis, and J. Urrutia. Domino tilings of orthogonal polygons. In *Canadian Conference on Computational Geometry (CCCG'99)*, pages 154–157, 1999.

[9] K. Culik. An aperiodic tiling of 13 wang tiles. *Discrete Math*, 160:245–251, 1996.

[10] J. Czyzowicz, E. Kranakis, and J. Urrutia. Domino tilings and two-by-two squares. In *Canadian Conference on Computational Geometry (CCCG'97)*, 1997.

[11] S. Hiller, O. Deussen, and A. Keller. Tiled blue noise samples. In *VMV '01: Proceedings of the Vision Modeling and Visualization Conference 2001*, pages 265–272. Aka GmbH, 2001.

[12] K. Jansen and H. Müller. The minimum broadcast time problem for several processor networks. *Theoretical Computer Science*, 147(1–2):69–85, 1995.

[13] J. Kari. Infinite snake tiling problems. In M. Ito and M. Toyama, editors, *Developments in Language Theory, 6th International Conference, DLT 2002, Kyoto, Japan, September 18-21, 2002, Revised Papers*, volume 2450 of *Lecture Notes in Computer Science*, pages 67–77. Springer, 2003.

[14] C. Kenyon and R. Kenyon. Tiling a polygon with rectangles. In *33rd Fundamentals of Computer Science (FOCS)*, pages 610–619, 1992.

[15] D. Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11(2):329–343, 1982.

[16] M. L. N. Elkies, G. Kuperberg and J. Propp. Alternating sign matrices and domino tilings, part 1. *Journal of Algebraic Combinatorics 1*, pages 111–132, 1992.

[17] M. L. N. Elkies, G. Kuperberg and J. Propp. Alternating sign matrices and domino tilings, part 2. *Journal of Algebraic Combinatorics 1*, pages 219–234, 1992.

[18] J. Propp. A reciprocity theorem for domino tilings. *The Electronic Journal of Combinatorics*, 8, 2001.

[19] W. Thurston. Conway's tiling groups. *American Mathematics Monthly*, 95:757–773, 1990.

[20] H. Wang. Proving theorems by pattern recognition. ii. *Bell Systems Technical Journal*, (40):1–41, 1961.

[21] C. Worman and M. Watson. Tiling layouts with dominoes. In *Proceedings of the 16th Canadian Conference on Computational Geometry (CCCG'04)*, pages 86–90, 2004.

[22] C. Worman and B. Yang. On the computation and chromatic number of colored domino tilings. In *Proceedings of the 17th Canadian Conference on Computational Geometry (CCCG'05)*, pages 188–191, 2005.

[23] C. Worman and B. Yang. On the computation of colored domino tilings of simple and non-simple orthogonal polygons. In *ISAAC*, volume 3827 of *Lecture Notes in Computer Science*, pages 859–868. Springer, 2005.