社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－AL－97 （9）
2004／10／15

タイル縁に上書きルールを用いた敷き詰め問題

岩間 一雄[†] 泉 公輔[††] 宮野 英次[†††] 小野 廣隆[††††]

† 京都大学大学院情報学研究科 〒606-8501 京都市左京区吉田本町
†† 九州工業大学情報工学部 〒820-8502 福岡県飯塚市川津 680-4
††† 九州工業大学大学院情報工学研究科 〒820-8502 福岡県飯塚市川津 680-4
†††† 九州大学大学院システム情報科学研究院 〒812-8581 福岡市東区箱崎 6-10-1
E-mail: †iwama@i.kyoto-u.ac.jp, ††izumi@theory.ces.kyutech.ac.jp, †††miyano@ces.kyutech.ac.jp,
††††ono@csce.kyushu-u.ac.jp

あらまし タイリング問題は古くから研究されている問題であり，多くの結果が報告されている．最も代表的なものとしては，各辺を色塗りされた 1×1 の正方形タイル (Wang タイル) を，2 枚のタイルの接触辺 (縁) の色が一致するときのみ貼り合わせることができるという条件の下で，与えられた枠の中に敷き詰めることができるかを問う問題である．本研究では，任意の 2 種類のタイルの貼り合わせが許されているが貼り合わせ後の辺の色は後に貼ったタイルの辺の色になる，という規則（辺上書きルール）の下でのタイリング，特に敷き詰め後に現れる図形の実現に関する問題を考える．まず，利用できるタイルの集合が与えられたとき任意の図形を実現できるための必要十分条件を示す．また，与えられた図形を実現するために必要なタイルの枚数を最小化する問題の NP 困難性とその問題に対する近似アルゴリズムを示す．

キーワード タイリング問題，タイリング可能性，NP 困難性，近似アルゴリズム

Tiling Problems with the Edge-Ovewriting Rule

Kazuo IWAMA[†], Kosuke IZUMI[††], Eiji MIYANO[†††], and Hirotaka ONO[††††]

† School of Informatics, Kyoto University, Yoshida Honmachi, Kyoto 606-8501, Japan.
†† Department of Systems Innovation and Informatics, Kyushu Institute of Technology, Kawazu, Fukuoka 820-8502, Japan.
††† Department of Systems Innovation and Informatics, Kyushu Institute of Technology Kawazu, Fukuoka 820-8502, Japan.
†††† Department of Computer Science and Communication Engineering, Kyushu University, Hakozaki, Fukuoka, 812-8581, Japan.
E-mail: †iwama@i.kyoto-u.ac.jp, ††izumi@theory.ces.kyutech.ac.jp, †††miyano@ces.kyutech.ac.jp,
††††ono@csce.kyushu-u.ac.jp

Abstract The tiling problems have received considerable attention for the last decades, and a lot of results are reported. One of the most popular variants of the problems could be tilings of regions with Wang tiles, which are 1×1 squares with colored edges, by using the following rule: If two tiles touch, then the color of their shared edges must be the same. In this paper, we consider a new tiling problem under the edge-overwriting rule in which any pair of tiles can touch but the color of the shared edges after tiling is determined by the edge color of the tile placed later. The problem input is a pair of a target figure and a tile set to be used, and the output is a sequence of tiles which realizes the target figure under the edge-overwriting rule. We show the sufficient and necessary condition of a tile set to realize any target figure, and then show NP-hardness proof and an approximation algorithm for the problem of minimizing the number of tiles realizing target figures.

Key words tiling problem, tileability, NP-hardness, approximation algorithm

1. Backgrounds, Problems and Our Contribution

The tiling problem, which looks simple but is actually very deep, has been attracting a lot of researchers [1], [2]. Among many other different settings for tiles (shapes, colors, etc.), *Wang tiles* are the simplest; each tile is a square of 1×1 and its four edges have colors [3]. If two tiles touch, then the color of their shared edges must be the same. The typical problem is called the *boundary coloring problem*: We are given a region with a colored boundary (like the rectangle of 2×4 in Figure 1-(a)) and a set of tiles (as shown in (b)), where a solid line denotes a color, say, black and a dotted one, say, white. It turns out that we have a solution as shown in Figure 1-(c), which has the same outer-border colors as (a). Inner-border colors are different but it is allowed if touching edges of two tiles have the same color. (One can see that if the size of the rectangle is 2×5, then we have no answer any longer.) Lewis showed [6] that the general problem is $\mathcal{NP}$-hard, but recently Moore, Rapaport, and Rémile showed [7] that if we can use only two colors, then there is a polynomial-time algorithm for almost all sets of tiles, by using an interesting algebraic property.

Although the above problem requires to color only outer borders, it would be more desirable if we can color inner borders also as designated. This stronger requirement appears quite natural for application like picture drawing. Unfortunately, the problem becomes trivial. For example, the rectangle of Figure 1-(a) cannot be colored obviously by the set (b). However, what if we change the rule of touching edges from coincidence to *overwriting*? Then we can color the rectangle (a) as shown in Figure 1-(d), where numbers denote the order of placing tiles. This new rule creates another kind of impossibility as shown later.

The main objective of this paper is to consider what kind of interesting problems we have by this new setting. Among others, what is especially important is that we can formulate a natural optimization version of the tiling problem, namely, minimizing the number of tiles. We also show a couple of "first-step" results which nevertheless suggest the possibility of wide extensions in the future.

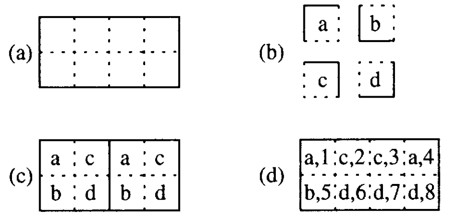


Figure 1 Tiling with Wang tiles

2. Tiling problems with Edge-Overwriting Rule

A *Wang tile* is a square of 1×1 and its four edges have either of two colors, *black* and *white*. Let W_{all} be the set of all Wang tiles with two colors. As shown in Figure 2, there are six kinds of different tiles if we allow these tiles to be rotated, where a solid and a dotted lines stand for black and white, respectively. Tiles (1) through (6) are referred to by I, L, II, C, O, Φ, respectively. We use an angle in $\{0, \pi/2, \pi, 3\pi/2\}$

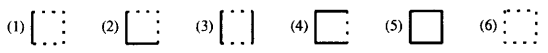


Figure 2 Wang tiles with two colors, (1) I-tile, (2) L-tile, (3) II-tile, (4) C-tile, (5) O-tile, (6) Φ-tile

to specify the rotation of tiles. Figure 3 show the example of rotations of L-tile, where L(θ) is the rotation of L-tile with angle θ.

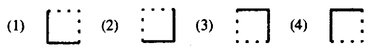


Figure 3 L-tile and its rotations: (1) L(0), (2) L($\pi/2$), (3) L(π), (4) L($3\pi/2$)

Let Λ be a set of two-dimensional $n \times n$ squares. Our problem of this paper, *tiling with the edge-overwriting rule*, is formulated as follows: Given the whole set of Λ's colored borders, called a *target figure* (i.e., as a blueprint), we find a *tiling sequence* of triples (**tile**, **angle**, **cell**)'s such that it can color all the borders according to the blueprint, where every tile is in a given subset $W \subseteqq W_{\text{all}}$ and all the four borders of each **cell** in the sequence are updated to new colors. In the following, we assume that the initial color of every border is white, i.e., if no tile is placed at cells facing a border, it remains white. On the other hand, black borders must be drawn by tiling. In this sense, we say that a target figure is *drawable* if there exists a tiling sequence realizing it. A tile set W is said to be *complete* if any figure can be drawable by W according to its border colors. We assume that Λ is encircled with cells where we can place tiles to color the boundary. For example, it is easy to say Figure 1-(d) cannot be drawn by $W_1 \overset{\text{def}}{=} \{I\}$ without the encircled cells. These help to exclude such trivial cases.

Under the assumptions, the following problem naturally arises: ***Tileability***: *For a tile set $W \subseteqq W_{\text{all}}$, is a given target figure f drawable by W?* It is obvious that any figure f is drawable by W_{all}. However, this is not true for much smaller W. Indeed, an I-tile can be simulated by neither $\{L\}$, $\{C\}$, $\{II\}$, nor $\{O\}$, as any unit length black border is not drawable by each of them. On the other hand, one may consider that $W_1 = \{I\}$ can draw any figure, but it is not true, i.e., W_1 is not complete. Figure 4-(1) shows an example; a cross-like figure cannot be drawn by W_1 since it forms a kind of cycle (Figure 4-(2))). How difficult is problem Tileability for W_1? We can judge if f is drawable by W_1 in linear time, since a necessary and sufficient condition of the drawability for W_1 is the non-existence of a cycle in a planar dualization of f.

[Proposition 1] ([5]) Given a given target figure f, its *neighbourhood graph* $G(f) = (V(f), E(f))$ is defined by

$$V(f) = C,$$
$$E(f) = \{(c_i, c_j) \mid c_i \text{ and } c_j \text{ shares a black border,}\}$$

where C is a set of all the cells. Then f is drawable by W_1 if and only if $G(f)$ is a forest. □

These observations derive a new problem: ***Completeness***: *Which tile set is complete?* A partial answer is that $W_2 \overset{\text{def}}{=} \{I, L\}$ is enough for example, since we have the following simple algorithm (say *CF algorithm*), which provides a tiling sequence drawing f by W_2.

```
Algorithm  CF (column-first) algorithm
Input:  A target figure $f$.
Output:  A tiling sequence $< s_1, s_2, \ldots, s_N >$.
0. Set i:=1,
1. For each $x$ from 1 to $n$,
2.   For each $y$ from 1 to $n$,
3.     let $s_i = (w, \theta, (x, y))$ where $w \in W_2$ such that the
       colors of top and left borders of $w(\theta)$ coincide the
       colors of the corresponding borders of cell $(x, y)$ of
       $f$. Set $i := i + 1$.
4. Output the sequence $< s_1, \ldots, s_N >$.
```

Figure 4-(3) illustrates how CF algorithm works: CF algorithm place tiles drawing its left and top borders, in column-first order, i.e., from top to bottom and then from left to right.

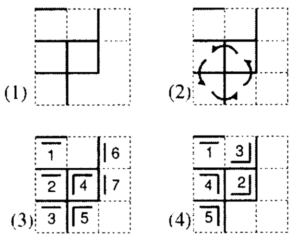


Figure 4 (1) Target figure f, (2) non-drawable by W_1, (3)a tiling sequence by W_2, (4) an optimal one

This simple algorithm also provides a condition for the completeness of a tile set.

[Proposition 2] A tile set W is complete if and only if all figures of 2×2 can be drawn by placing tiles of W on the 2×2 space†. □

This can be shown by the fact that tile sets satisfying the condition can simulate CF algorithm. By Proposition 2 we have the following corollary.

[Corollary 1] Tile sets $\{I, L\}, \{L, \Phi\}$, $\{I, C, \Phi\}$, and $\{I, O, \Phi\}$ are all of the minimal complete tile sets.

Next interest is how efficiently f can be drawn by a complete tile set W. Usually, a tiling sequence drawing f is not unique, e.g., in Figure 4, we have a tiling sequence with length 5 (Figure 4-(4)) though the algorithm's output has length 7 (Figure 4-(3)). Here we consider the following: ***Minimum Tiling Sequence (MTS). Find a minimum tiling sequence of W drawing f.*** We have results for a complete tile set W_2: Problem MTS for $W_2 = \{I, L\}$ is $\mathcal{NP}$-hard, and we have a 1.5-approximation algorithm solving it. In the next section, we show these results.

Remark There are several directions for future research. An obvious one is to extend Problem MTS for other sets of tiles. For $\{I, L, II, C\}$, we have already shown its $\mathcal{NP}$-hardness. As for Problem Tileability, probably there are no hard cases, but it is quite interesting if any. Another interesting extension is to introduce larger Wang tiles, say, 2×3 tiles, which have both outer and inner colored edges. This is closely related to the rectilinear polygon covering and the picture drawing. Three or more color cases and other edge-operations such as OR and XOR for all problems would be also interesting.

3. Complexity of MTS for $W_2 = \{I, L\}$

In this section, we consider the complexity of Problem MTS for a minimal complete tile set $W_2 = \{I, L\}$, which seems to be the most basic. Unfortunately, it is $\mathcal{NP}$-hard.

[Theorem 1] Problem MTS for $W_2 = \{I, L\}$ is $\mathcal{NP}$-hard.

Proof. The proof is by polynomial reduction from planar MAX 2-SAT whose hardness is proved in [4]. We shall show that for a given Boolean formula ϕ of planar 2-SAT type we can construct a figure F such that F can be drawn by a tiling sequence of length k or shorter if and only if there exists a truth assignment that satisfy a given or larger number of clauses in ϕ. The reduced figure consists of *variable*, *clause*, and *path gadgets*. Roughly speaking, to minimize the

†: This condition is redundant and a more essential condition exists, though it requires a little more complicated statement.

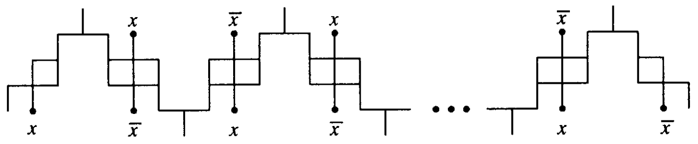


Figure 5 variable gadget

sequence length, we replace as many pairs of two I-tiles with one L-tile as possible. Thus the key idea of the reduction is to associate one of the satisfied clauses with such a replacement.

Suppose that $\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ with n variables and m clauses. Also, suppose that, for $i = 1, 2, \cdots, m$, each clause C_i has exactly two distinct literals l_1^i and l_2^i. Our variable gadget consists of n connected components. One of the components, e.g., for variable x and its negation $\overline{x}$, shapes a jagged structure, and has upward and downward branches with terminals labeled x or $\overline{x}$, as illustrated in Figure 5.

The clause gadget has m components, each of which forms a zig-zag path from five edges as shown in Figure 6. For example, if $l_1^i = \overline{x}$ and $l_2^i = y$ for the ith clause C_i, then the terminals labeled l_1^i and l_2^i are connected with the $\overline{x}$ and the y terminals in the clause gadget by the path gadget, which are also zigzag paths. The complete construction for $C_i = \overline{x} + y$ is shown in Figure 7.

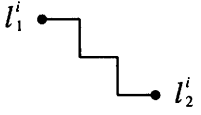


Figure 6 clause gadget

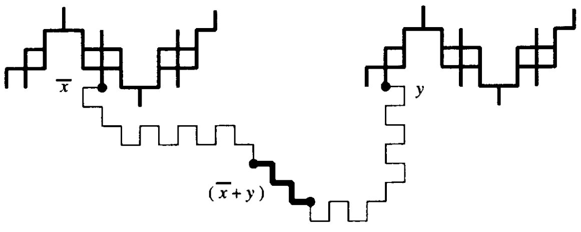
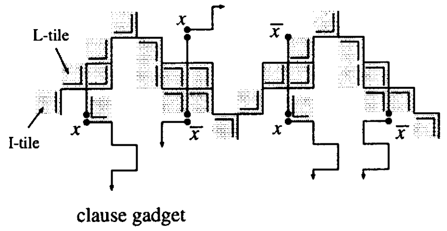


Figure 7 $C_i = \overline{x} + y$

The above construction guarantees that there are only two minimum tiling sequences for each component of the variable gadget (see Figure 8): If the leftmost vertical border is drawn by using an I-tile, then this corresponds to the variable being set to *true*, otherwise *false*. One can see that, for example, if x = *true* (top), then we can use L-tiles, each which can draw one border of the variable gadget and one border of the clause gadget, at the points x's, otherwise at the points $\overline{x}$'s (bottom). (Note: Strictly speaking, the L-tile draws one border of the variable gadget and one border of the *path* gadget. However, since all borders of the path gadget can be drawn only by using L-tiles except for connecting parts with the other gadgets, we regard the latter border as one of the clause gadget in the following.)



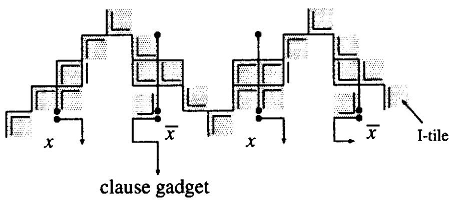


Figure 8 x = *true* (top) and *false* (bottom)

See Figure 6 again. If at least one border can be drawn together with one border of the variable gadget by an L-tile, then each clause gadget can be drawn only by two L-tiles, otherwise we need at least three tiles. Hence the question whether there exists a truth assignment that satisfy a given or larger number of clauses in a planar MAX 2-SAT formula ϕ is equivalent to asking whether F can be drawn by a tiling sequence of length k or shorter (the value of k can be obviously obtained by a simple calculation). □

[Theorem 2] There exists a 1.5-approximation algorithm solving Problem MTS for $W_2 = \{I, L\}$.

Proof Sketch of Theorem 2. Here we give a 1.5-approximation algorithm CF* for MTS of $\{I, L\}$, which utilizes CF tiling. The algorithm CF* is simple, and roughly speaking, the idea is just to replace as many pairs of two I tiles of CF's output with L tiles, as possible. To explain this, let us call a pair of two tilings $(I, 0, (x, y))$ and $(I, 3\pi/2, (x, y+1))$ in a given tile sequance *Type* A (Figure

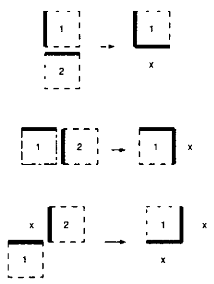


Figure 9 Type A pair of I's (top), B (center), C (bottom)

9, top), and call a pair of $(I, 3\pi/2, (x, y))$ and $(I, 0, (x+1, y))$ *Type* B (Figure 9, center). Also we call a pair of $(I, 3\pi/2, (x, y))$ and $(I, 0, (x+1, y-1))$ where cell $(x-1, y-1)$ is empty *Type* C.

The CF* algorithm replaces these types of pairs of tilings with L tiles so that all the replacements preserve the drawability of the original sequence. (see Figure 9 again.)

```
Algorithm   Algorithm CF*
Input:  A target figure $f$.
Output:  A tiling sequence $< s_1, s_2, \ldots, s_N >$.
0. Apply CF to $f$ and let $< s_1, s_2, \ldots, s_n >$ its output.
1. while (T $\in$ {A, B})
2.   For each $x$ from 1 to $n$,
3.     For each $y$ from 1 to $n$,
4.       Replace all Type T pairs of tilings with $L$ tiles.
5.   For each $x$ from 1 to $n$,
6.     For each $y$ from 1 to $n$,
7.       Delete all Type C pairs of $I$ tiles, and insert
         the equivalent $L$ tiles to the the sequence top.
8.   Output the sequence $< s_1, \ldots, s_N >$.
```

In case of Types A and B, the replacements are simple: Step 4 of the algorithm replaces Type A pair of tilings with one L tile; i.e., for $s_i = (I, 0, (x, y))$ and $s_{i+1} = (I, 3\pi/2, (x+1, y))$, set $s_i := (L, 0, (x, y))$ and $s_i := \emptyset$. (In case of Type B, for $s_i = (I, 3\pi/2, (x, y))$ and $s_j = (I, 0, (x+1, y))$, set $s_i := (L, 3\pi/2, (x, y))$ and $s_j := \emptyset$.) On the other hand, Type C pairs require a careful treatment: Type C's replacement, different from Type A and B, places the L tile on a new cell $(x-1, y-1)$. This may affect the colors of the left and the top borders of $(x-1, y-1)$; To avoid this, the algorithm changes the tiling order. In step 7 of the algorithm, for $s_i = (I, 3\pi/2, (x, y))$ and $s_j = (I, 0, (x+1, y-1))$, first Type C tiles are deleted, i.e., set $s_i := \emptyset$ and $s_j := \emptyset$. Then, insert $(L, \pi/2, (x, y))$ to the sequence top, i.e., for $r = 1, 2, \ldots, i-1$, set $s_{r+1} := s_r$ and $s_1 := (L, \pi/2, (x, y))$. This order change preserves the colors of the left and the top borders of $(x-1, y-1)$. Also the colors of the right and the bottom borders of $(x-1, y-1)$ are not over-written, because no tile is placed on cells (x, y) and $(x+1, y-1)$. Consequently, the replacements preserves the drawability.

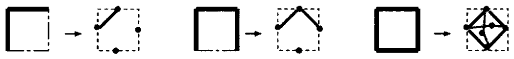


Figure 10 Border graph

Here, we show the approximation ratio of 1.5. We call a tiling sequence S *minimal* if in S borders once colored black is kept black. If S drawing f is minimal, then the length of S is $p(S) + q(S)$, and $m = 2p(S) + q(S)$ holds, where $p(S)$ and $q(S)$ are the numbers of L and I tiles in S, respectively. Notice that a tiling sequence by algorithm CF* is minimal. Now, we restrict our attention into minimal tiling sequences, since any tiling sequence drawing f can be reduced to a minimal one. Under this restriction, our problem is equivalent to the problem of finding minimal S (drawing f) that maximizes $p(S)$. To estimate an upper bound on $p(S)$, we relax S's condition as follows: It can be computed as a maximum matching of a *border graph* defined by a dual transformation as Figure 10, and L tiles located by CF* form maximal matching on the border graph. Since the size of any maximal matching is at least 1/2 of the size of maximum matching, $p(S_A) \geq 1/2 \cdot p(S^\top)$ holds, where S_A is a tiling sequence by CF* and $S^\top$ is one giving the upper bound. The approximation ratio is at most $\frac{p(S_A)+q(S_A)}{p(S^\top)+q(S^\top)} \leq 2 - \frac{p(S_A)}{p(S^\top)} \leq 1.5$. We have many tight examples for CF*. □

References

[1] C. Allauzen and B. Durand, "Tiling problems", In E.Borger, E.Gradel, Y. Gurevich, "The classical decision problem", Springer-Verlag, 1997.

[2] R. Berger, "The undecidability of the domino problem," *Memoirs Amer. Math. Soc.* 66, 1–72 (1966).

[3] K. Culik and J. Kari, "On Aperiodic Sets of Wang Tiles,", *Foundations of Computer Science: Potential - Theory - Cognition* Lecture Notes in Computer Science 1337, 183–208, (1997).

[4] L.J. Guibas, J.E. Hershberger, J.S.B. Mitchell, and J.S. Snoeyink, "Approximating polygons and subdivisions with minimum link paths," *Int. J. Comp. Geom. and Applications*, Vol. 3, No. 4, 383–415 (1993).

[5] K. Iwama, K. Izumi, E. Miyano and H. Ono, "Drawing Borders Efficiently", Tech. Rep. of IEICE, COMP2004-25, 47–54 (2004).

[6] H. Lewis, "Complexity of solvable cases of the decision problem for predicate calculus," *Symposium on Foundations of Computer Science*, 35–47 (1978).

[7] C. Moore, I. Rapaport, E. Rémila, "Tiling groups for Wang tiles," *Symposium on Discrete Algorithms*, 402–411 (2002).