

Planar Dimer Tilings

Olivier Bodini and Thomas Fernique

LIRMM, 161 rue Ada 34392 Montpellier Cedex 5 - France,
{bodini,fernique}@lirmm.fr

Abstract. Domino tilings of finite domains of the plane are used to model dimer systems in statistical physics. In this paper, we study *dimer tilings*, which generalize domino tilings and are indeed equivalent to perfect matchings of planar graphs. We use *height functions*, a notion previously introduced by Thurston in [10] for domino tilings, to prove that a dimer tiling of a given domain can be computed using any Single-Source-Shortest-Paths algorithm on a planar graph. We also endow the set of dimers tilings of a given domain with a structure of distributive lattice and show that it can be effectively visited by a simple algorithmical operation called flip.

1 Dimer tilings

A *cell* is a closed polygonal set of \mathbb{R}^2 and a *domain* is a finite set of cells with disjoint interiors. Two cells of a domain are said *adjacent* if they share at least one boundary edge. A domain \mathcal{C} is said *tileable* if all its cells can be grouped two by two, two grouped cells being adjacent. If it exists, such a grouping is called a *dimer tiling* of \mathcal{C} . Notice that a tileable domain can admit many dimer tilings: we denote by $\Delta(\mathcal{C})$ the set of dimer tilings of \mathcal{C} . Fig. 1 illustrates these notions.

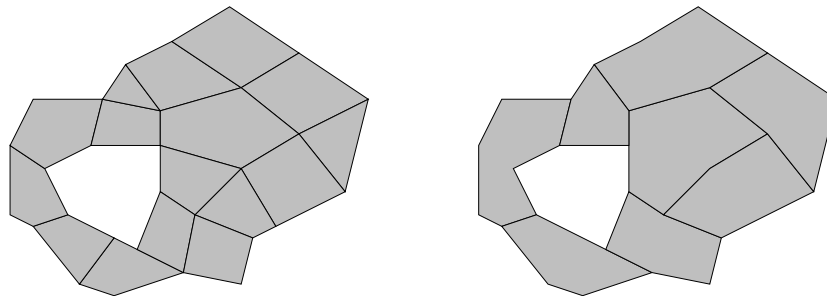


Fig. 1. A domain of 16 cells (left) and a dimer tiling of it, the grouped cells being represented by a single polygon (right).

Dimer tilings arise for example in statistical physics to model the behavior of dimer systems: cells are squares and thus dimer are rectangles, called *dominoes*

(see e.g. [5, 6]). In this context, it is particularly interesting to endow the set of dimer tilings of a domain with a structure suitable for performing random sampling.

Dimer tilings are also connected with *perfect matchings* of planar graphs. Indeed, we can associate to a set of cells \mathcal{C} an undirected planar graph denoted by $A(\mathcal{C})$: to each cell of \mathcal{C} corresponds a vertex of $A(\mathcal{C})$, two of them being connected by an edge if the corresponding cells are adjacent. Then, there is a natural bijection between the perfect matchings of $A(\mathcal{C})$ and the dimer tilings of \mathcal{C} (see Fig. 2).

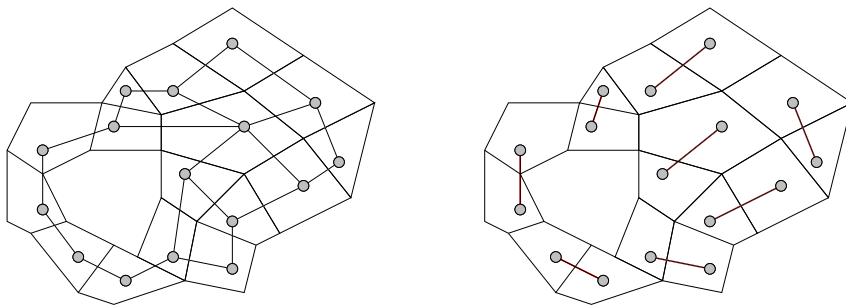


Fig. 2. Left, the planar graph corresponding to the set of cells of Fig. 1. Right, the perfect matching of this graph corresponding to the dimer tiling of Fig. 1.

In this paper, we mainly focus on the two following problems: the first want to compute a dimer tiling while the second want to provide a way to move on the set of dimer tilings of a fixed domain. This paper is organized as follows. In Section 2, we associate to a domain a weighted directed graph and define *height function* over the vertices of this graph. Such an approach has been firstly used in [10] to compute in linear time a dimer tiling of a simply connected domain made of square cells, and then extended in [1] to regular cells (all with the same number of edges). We extend here these results to domain with holes made of non-regular cells. First, we show in Section 3 and 4 that the computation of a dimer tiling of a bipartite domain (or, equivalently, of a perfect matching of a bipartite planar graph) can be reduced to a single-source-shortest-paths problem on a planar graph. This yields in particular a $\mathcal{O}(n \ln(n)^3)$ -algorithm to compute a dimer tiling of a domain which is, contrarily to [10, 1], neither necessarily simply connected nor made of regular cells. Second, we endow in Section 5 the set of dimer tilings of a domain with a structure of distributive lattice and defines a simple effective operation, called *flip*, which allows to visit it. This can be used for example to perform random sampling on the set of dimer tilings.

2 General settings

2.1 Weight and height function

A directed graph is denoted by $G = (V, E)$, V (resp. E) being the set of vertices (resp. edges) of G .

A *directed path* is a sequence e_1, \dots, e_p of edges such that e_i points to the vertex e_{i+1} starts from, and a *circuit* is a directed path whose last edge points to the vertex the first edge of this path start from.

A *weight function* over G is a map $w : E \rightarrow \mathbb{R}$, extended to a set of edges X (in particular a directed path or a circuit) by:

$$w(X) = \sum_{e \in X} w(e).$$

Then, the *height function* associated to the weight function w and to the vertex $v^* \in V$ is the map from V to $\mathbb{R} \cup \{-\infty\}$ denoted by h_{w,v^*} and defined by:

$$\forall v \in V, \quad h_{w,v^*}(v) = \inf\{w(p) \mid p \text{ is a path from } v^* \text{ to } v\}.$$

2.2 The graph of a bipartite domain

Let \mathcal{C} be a domain as defined in Section 1. We suppose that \mathcal{C} is *bipartite*: we split it into two sets \mathcal{C}_b and \mathcal{C}_w (resp. the *black* cells and the *white* ones), such that two adjacent cells never belong to the same set (that is, are of different colors). Then, given an orientation of the plane, we define the directed graph $G(\mathcal{C})$ as follows (see Fig. 3):

- to each vertex of \mathcal{C} corresponds a vertex of $G(\mathcal{C})$;
- to an edge shared by two adjacent cells of \mathcal{C} (and thus of different colors) corresponds in $G(\mathcal{C})$ an edge directed so that the black cell is on its left;
- to an edge on the boundary of the domain (and thus belongs to only one cell) correspond in $G(\mathcal{C})$ a bidirected edge.

In particular, to each cell of \mathcal{C} naturally corresponds a circuit of $G(\mathcal{C})$, called *cell-circuit* in all what follows.

From now on and up to the end of the paper, \mathcal{C} stands for a set of cells whose union is a bipartite connected domain, and v^* stands for a fixed vertex of the outer boundary of $G(\mathcal{C})$.

Once v^* is fixed, we will simply denote by h_w the height function h_{w,v^*} associated to the weight function w . Then, the idea of the paper is to define particular weight functions on the graph $G(\mathcal{C})$, such that height functions can be used to compute the ones which correspond to dimer tilings of \mathcal{C} .

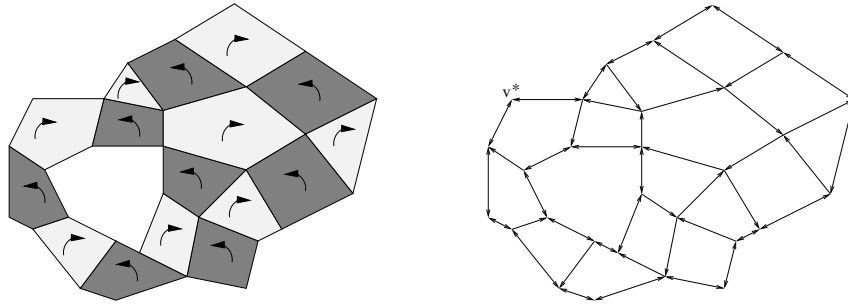


Fig. 3. The orientations of black and white cells of the bipartite domain of Fig. 1 (left). The corresponding directed graph, whose bidirected edges correspond to the boundaries of the domain, the vertex v^* being on the outer boundary (right).

3 Counters and dimer tilings

We define here *counters* and use them to give a characterization of tileable domains. The results provided here are then used in the next section to compute effectively a dimer tiling.

Definition 1. A counter over $G(\mathcal{C})$ is a weight function δ such that $\delta(e) = 0$ for any bidirected edge e and $\delta(c) = 1$ for any cell-circuit c . A counter is moreover said binary if $\delta(e) \in \{0, 1\}$ for any edge e .

Clearly, a binary counter weights exactly one edge of a cell-circuit by 1, the other ones having weight 0. Hence, grouping two cells which share an edge of weight 1 yields a dimer tiling (see Fig. 4). Conversely, it is straightforward to similarly derive a binary counter from a dimer tiling. Thus, we use indifferently the terms *dimer tiling* or *binary counter* in all that follows.

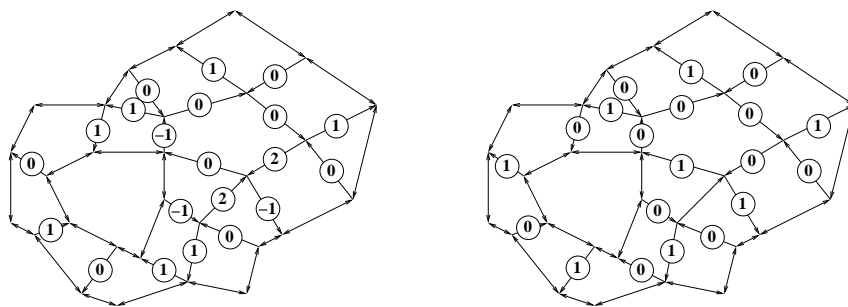


Fig. 4. A counter (left) and the binary counter corresponding to the dimer tiling of Fig. 1 (right). For the sake of clarity, the weights 0 of bidirected edges are not represented.

Let us now consider two counters δ and δ' , and let c be a circuit of $G(\mathcal{C})$. One easily proves by induction on the size of c that $\delta(c) = \delta'(c)$. In particular, if \mathcal{C} is tileable then one can consider δ' to be a *binary* counter: one has $\delta'(c) \in \mathbb{N}$, and this yields $\delta(c) \in \mathbb{N}$. Thus:

Proposition 1. *If \mathcal{C} is tileable, then $\delta(c) \in \mathbb{N}$ for any counter δ .*

Conversely, suppose that for any counter δ' and any circuit c , $\delta'(c) \in \mathbb{N}$. In such a case, we can define the notion of δ -shortest path from a vertex v to a vertex v' : it is a path p (not necessarily unique) which satisfies

$$\delta(p) = \min\{\delta(p') \mid p' \text{ is a path from } v \text{ to } v'\}.$$

Then one has the following properties:

Proposition 2. *If δ and δ' are counters over $G(\mathcal{C})$, then any δ -shortest path is also a δ' -shortest path.*

Proposition 3. *If δ and δ' are counters over $G(\mathcal{C})$, then $h_\delta = h_{\delta'}$ yields $\delta = \delta'$.*

We then prove:

Theorem 1. *Let δ be a counter over $G(\mathcal{C})$ and set for an edge e from v to v' :*

$$\delta_\perp(e) = \delta(e) - (h_\delta(v') - h_\delta(v)).$$

Then, δ_\perp is the binary counter such that $h_{\delta_\perp}(v) = 0$ for any vertex v .

Thus, Th. 1 and Prop. 1 yield that \mathcal{C} is tileable if and only $\delta(c) \in \mathbb{N}$ for any counter δ over $G(\mathcal{C})$ and any circuit c of $G(\mathcal{C})$. This provides a characterization of tileable bipartite domains that we use in the following section.

4 Computing a binary counter

The previous section has defined counters and binary counters. We are especially interested in binary counters since they correspond to dimer tilings. Here, we first show how to compute a counter in linear time, and we then use Th. 1 to derive a binary counter from it.

The first step to compute counter consists in constructing a particular weighted tree. Let $A(\mathcal{C})$ be the undirected graph associated to \mathcal{C} as explained in the introduction. Since the domain is bipartite and connected, so is $A(\mathcal{C})$. Let us assign color black or white to the vertices of $A(\mathcal{C})$, so that two linked vertices have different colors. Let then T be a spanning tree of $A(\mathcal{C})$. If we remove from T an edge e between a black vertex b and a white vertex w , this splits T into two trees: we denote by $T_{e,b}$ the one which contains the vertex b and we set:

$$d_T(e) = \#\{\text{black vertices in } T_{e,b}\} - \#\{\text{white vertices in } T_{e,b}\}.$$

It defines a function d_T from the edges of T to \mathbb{Z} (see Fig. 5). One checks:

Proposition 4. *Let v be a vertex of T and e_1, \dots, e_k be the edges of T containing v . If the domain has as much black as white cells, then one has:*

$$\sum_{i=1}^k d_T(e_i) = 1.$$

Notice that a tileable bipartite domain has necessarily as much black as white cells since each black cell is grouped with a white cell. Simple examples show that the converse is however false.

The second step to compute counter consists in deriving from the function d_T a counter δ_T over $G(\mathcal{C})$. We proceed as follows. We set $\delta_T(e) = 0$ for any bidirected edge (that is, an edge of the boundaries of the domain). Otherwise, a directed edge e corresponds to an edge shared by two cells of \mathcal{C} , say C_b and C_w , which correspond in the graph $A(\mathcal{C})$ respectively to two vertices b and w , connected by an undirected edge e' of $A(\mathcal{C})$; we set $\delta_T(e) = d_T(e')$ if e' belongs to the tree T , $\delta_T(e) = 0$ else (see Fig. 5). Prop. 4 then yields that $\delta_T(c) = 1$ for any cell-circuit c . Thus, δ_T is a counter.

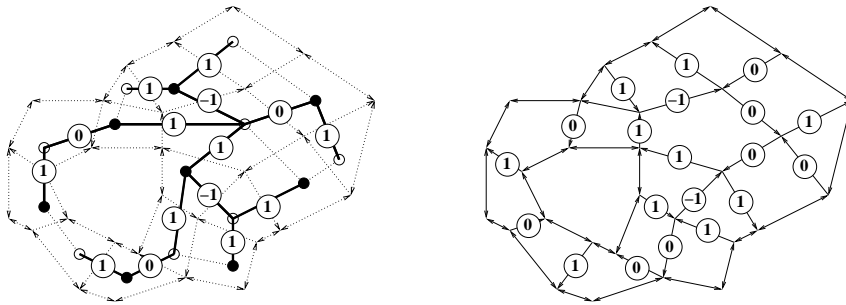


Fig. 5. Left, a spanning tree T over $A(\mathcal{C})$ and the weights d_T . Right, the corresponding weight function δ_T over $G(\mathcal{C})$: it is a counter.

Let us study the complexity of the construction of this counter. Let n be the number of cells of the domain \mathcal{C} , or equivalently, the number of vertices of $A(\mathcal{C})$. Constructing a spanning tree T can be done in linear time by a greedy algorithm. Then, the weight function d_T can be computed recursively in linear time, starting from the leaves of T . Deriving δ_T from d_T can be performed in linear time since, the graph $G(\mathcal{C})$ being planar, it has $\mathcal{O}(n)$ edges. Thus, the counter δ_T can be computed in linear time.

Then, Th. 1 allows to derive the binary counter δ_\perp from the height function of δ_T , which can be computed by any single-source-shortest-paths algorithm on the

planar graph $G(\mathcal{C})$ weighted by δ_T , the source being the vertex v^* . In particular, [3] provides a $\mathcal{O}(n \ln(n)^3)$ -algorithm to do this. Finally one has:

Theorem 2. *A dimer tiling of a bipartite tileable domain can be constructed by a $\mathcal{O}(n \ln(n)^3)$ -algorithm.*

The previous algorithm can also be used to detect the case no dimer tiling exist:

- if the bipartite domain is not balanced (hence not tileable), then the construction of the weight function d_T leads to a vertex such that the sum of the weights of its adjacent edges is not equal to 1;
- otherwise, δ_T is a counter, and \mathcal{C} non-tileable yields the existence of a circuit c such that $\delta_T(c) \notin \mathbb{N}$, more precisely $\delta_T(c) < 0$ since $\delta_T(c) \in \mathbb{Z}$ by construction. Thus, shortest paths are not defined, and the algorithm of [3] detects it (as most of the shortest paths algorithms).

Notice that our algorithm has a complexity similar to the $\mathcal{O}(n \ln(n))$ algorithm of [9], which deals with the case of square cells and domain with a bounded number of holes.

5 Random sampling

In this section, we suppose that there exists at least one binary counter over $G(\mathcal{C})$, that is, $\Delta(\mathcal{C})$ is not empty. We endow $\Delta(\mathcal{C})$ by a structure of distributive lattice which can be visited using a simple operation called *flip*.

5.1 Flips

Definition 2. *Let δ be a binary counter over $G(\mathcal{C})$. A δ -nodule is a maximal¹ set of vertices of $G(\mathcal{C})$ such that any two of them are linked by a directed path p which satisfies $\delta(p) = 0$.*

Notice that if δ is a binary counter, then a path p such that $\delta(p) = 0$ is always a shortest path for δ . So it follows from Prop. 2 that two binary counters define the same nodules: we thus simply speak about *nodules*. Moreover, notice that h_δ is always constant over the vertices of a nodule: intuitively, a nodule can be seen as an expanded vertex.

So we call *incoming* (resp. *outcoming*) edge of a nodule a directed edge of $G(\mathcal{C})$ which links a vertex outside the nodule to a vertex inside of it (resp. inside to outside). We also denote by A^* the nodule which contains the vertex v^* , and we then define the following operation:

Definition 3. *Let A be a nodule, $A \neq A^*$. Suppose that δ is a binary counter such that $\delta(e)$ is equal to 1 (resp. 0) for each incoming edge of A and 0 (resp. 1) for each outcoming edge of A . We call decreasing flip (resp. increasing flip) the operation which exchanges the weights of the incoming and outcoming edges of A .*

¹ for inclusion

One easily checks that a flip on a nodule A transforms a binary counter δ into a binary counter, say δ' (see Fig. 6). Moreover, if it is an increasing flip (resp. decreasing), then $h_{\delta'}(v) = h_{\delta}(v) + 1$ (resp. $h_{\delta'}(v) = h_{\delta}(v) - 1$) if $v \in A$ and $h_{\delta'}(v) = h_{\delta}(v)$ otherwise: flips act on heights in a very simple way.

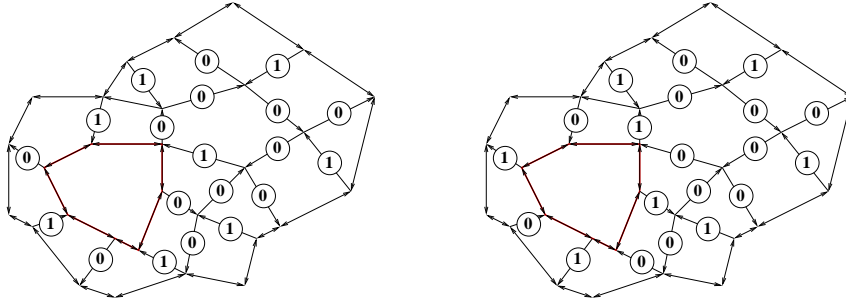


Fig. 6. The seven vertices which belong to the bidirected edges around the hole form a nodule. A (decreasing) flip on this nodule transforms the binary counter on the left into the one on the right. It corresponds on Fig. 7 to the flip from the upper dimer tiling to the one immediately below.

5.2 A distributive lattice

We define two operations \vee and \wedge over the set $\Delta(\mathcal{C})$ of dimer tilings of \mathcal{C} :

Proposition 5. *Let δ and δ' be two binary counters over $G(\mathcal{C})$. Then the height functions $\min(h_{\delta}, h_{\delta'})$ and $\max(h_{\delta}, h_{\delta'})$ are height functions of binary counters over $G(\mathcal{C})$, respectively denoted by $\delta \wedge \delta'$ and $\delta \vee \delta'$:*

$$h_{\delta \wedge \delta'} = \min(h_{\delta}, h_{\delta'}) \quad \text{and} \quad h_{\delta \vee \delta'} = \max(h_{\delta}, h_{\delta'}).$$

It is then easy to check that $(\Delta(\mathcal{C}), \wedge, \vee)$ is a distributive lattice. We denote by \preceq the associated partial order:

$$\delta \preceq \delta' \Leftrightarrow \delta = \delta \wedge \delta' \Leftrightarrow h_{\delta} \leq h_{\delta'}.$$

Notice that it is not difficult to endow a finite set with a structure of distributive lattice. The interest of this specific definition follows from its link with the flips. Recall first that, given δ and δ' in $\Delta(\mathcal{C})$, one says that δ' covers δ for the partial order \preceq if $\delta \prec \delta'$ and if, for any $\delta'' \in \Delta(\mathcal{C})$, $\delta \prec \delta''$ yields $\delta' \preceq \delta''$. Then one has:

Theorem 3. *A binary counter δ' covers a binary counter δ if and only if δ can be obtained performing a decreasing flip on δ' .*

In other words, the Hasse's diagram of the distributive lattice $(\Delta(\mathcal{C}), \wedge, \vee)$ (two elements are linked if and only if one covers the other) is isomorphic to the

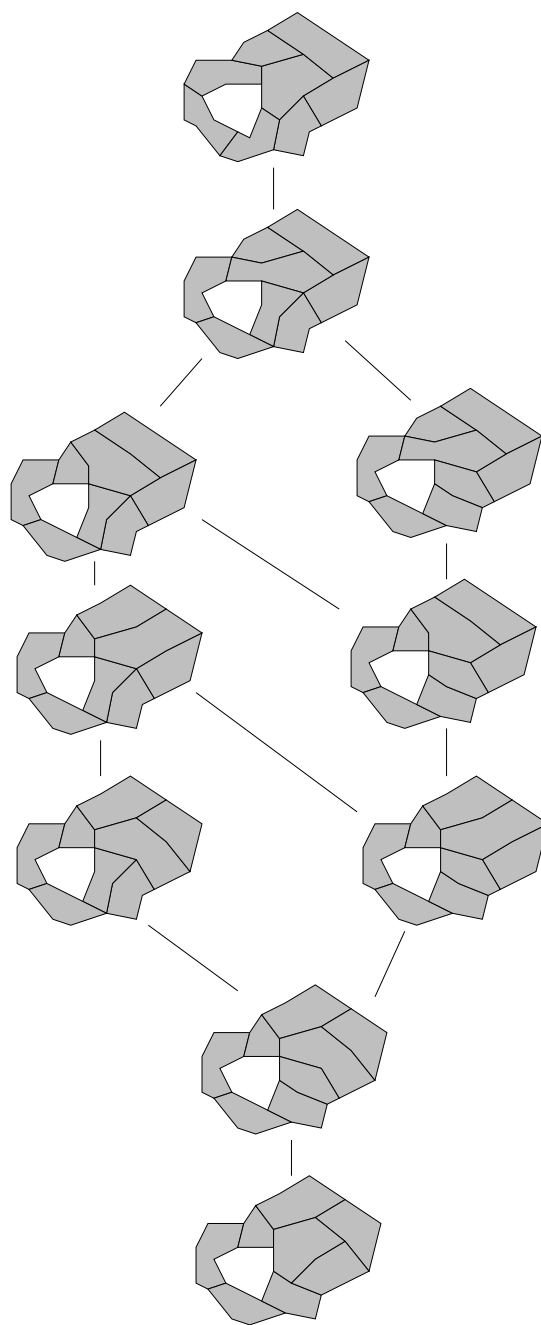


Fig. 7. The distributive lattice of all the dimer tilings of a domain. The dimer tiling δ_{\perp} defined in Th. 1 is the bottom of this lattice. The flips allow to move between linked tilings. Notice that, excepted the upper flip which is performed on the nodule shown Fig. 6, all the other flips are performed on nodules reduced to a single vertex.

undirected graph whose vertices correspond to dimer tilings, each of them linked to the ones it is covered by. Figure 7 illustrates this.

Such a structure of distributive lattice, together with the effective operation flip, can then be used to generate randomly a dimer tiling of a given domain (see [8]). In [4], we also use this lattice and the flips to generate all the $|\Delta(\mathcal{C})|$ dimer tilings of \mathcal{C} with less than $2|\Delta(\mathcal{C})|$ flips. Since a flip can be easily performed in linear time, it leads to an algorithm in $\mathcal{O}(n|\Delta(\mathcal{C})|)$. It thus improves the $\mathcal{O}(n^2)$ algorithm of [2] which is moreover restricted to the case of square cells.

References

1. O. Bodini, M. Latapy, *Generalized Tilings with Height Functions*. *Morfismos* **7** (2003).
2. S. Desreux, M. Matamala, I. Rapaport, E. Remila, *Domino tiling and related models : space of configurations of domains with holes*. *Theoret. Comput. Sci.* **319** (2004), 83–101.
3. J. Fakcharoenphol, S. Rao, *Planar graphs, negative weight edges, shortest paths, and near linear time*. *FOCS 2001*, 232–241.
4. T. Fernique, *Pavages d'une polycellule*. LIRMM Research Report 04002 (2004), available at http://www.lirmm.fr/~fernique/info/memoire_mim3.ps.gz
5. P. W. Kasteleyn, *The statistics of dimers on a lattice. I. The number of dimer arrangements on a quadratic lattice*. *Physica* **27** (1961), 1209–1225.
6. R. Kenyon, *The planar dimer model with boundary: a survey*. *Directions in mathematical quasicrystals*, M. Baake and R. Moody, eds. CRM monograph series (AMS, Providence, RI, 2000).
7. J. Propp, *Lattice structure of orientations of graphs*. Preprint (1993), available at <http://www.math.wisc.edu/propp/orient.html>.
8. J. Propp, *Generating random elements of finite distributive lattices*. Preprint (1997), available at <http://www.math.wisc.edu/~propp/wilf.ps.gz>.
9. N. Thiant, *An $\mathcal{O}(n \log n)$ -algorithm for finding a domino tiling of a plane picture whose number of holes is bounded*. *Theoret. Comput. Sci.* **303** (2003), 353–374.
10. W. P. Thurston, *Conway's tiling group*. *American Mathematical Monthly*, **97** (1990), 757–773.

Appendix

Proof of Proposition 2:

Let v and v' be two vertices. Let ψ and ψ' be two counters. Let c be a ψ -shortest path and c' be a ψ' -shortest path, both linking v to v' . Let r be any path from v' to v . Prop. 1 applied to circuits $c.r$ and $c'.r$ yields $\psi(c.r) = \psi'(c.r)$ and $\psi(c.r) = \psi'(c.r)$. It follows that:

$$\psi(c') - \psi(c) = \psi'(c') - \psi'(c).$$

But since c is a ψ -shortest path, $\psi(c') - \psi(c) \geq 0$. Similarly, c' being a ψ' -shortest path, $\psi'(c') - \psi'(c) \leq 0$. Thus $\psi(c') - \psi(c) = \psi'(c') - \psi'(c) = 0$, that is, c is a shortest path for both ψ and ψ' (and c' too). \square

Proof of Proposition 3:

Let ψ and ψ' be two counters such that $h_\psi = h_{\psi'}$. Let $e = (v, v')$ be an edge. If $e \notin I$, then $\psi(e) = \psi'(e) = 0$ by definition.

Otherwise, let c be a shortest path from v^* to v (for both ψ and ψ' according to Prop. 2). Hence $\psi(c) = h_\psi(v) = h_{\psi'}(v) = \psi'(c)$. Similarly, there exists a path c' from v^* to v' such that $\psi(c') = \psi'(c')$. Considering then a path r from v' to v^* and applying Proposition 1 to circuits $c.e.r$ and $c'.r$, one gets easily:

$$\psi(c.e) - \psi'(c.e) = \psi(c') - \psi'(c').$$

Combined with $\psi(c) = \psi'(c)$ and $\psi(c') = \psi'(c')$, it yields $\psi(e) = \psi'(e)$. \square

Proof of Theorem 1:

It is easy to check that δ_\perp is a counter. Let us prove that it is a **binary** counter. Let $e = (v, v')$ be an edge:

- If there exists a ψ -shortest path p from v^* to v' such that e is the last edge of p , then $h_\psi(v') = h_\psi(v) + \psi(e)$, that is, $\delta_\perp(e) = 0$.
- Otherwise, one has $h_\psi(v') < h_\psi(v) + \psi(e)$, that is, $0 < \delta_\perp(e)$. But the edge e belongs to a circuit-cell c whose other edges form a path from v' to v of length $1 - \psi(e)$ (since $\psi(c) = 1$). Hence $h_\psi(v) \leq h_\psi(v') + 1 - \psi(e)$, that is, $\delta_\perp(e) \leq 1$. Thus $0 < \delta_\perp(e) \leq 1$: proving that $\delta(e)$ is entire will yield $\delta(e) = 1$. Let a and b be ψ -shortest paths from v^* to respectively v and v' and r be a path from v' to v^* . One has $\delta_\perp(a) = \psi(a) - h_\psi(v)$, that is, since a is a ψ -shortest path, $\delta_\perp(a) = 0$. Similarly, $\delta_\perp(b) = 0$, and so:

$$\begin{aligned} \delta_\perp(a.e.r) &= \delta_\perp(a) + \delta_\perp(e) + \delta_\perp(r) \\ &= 0 + \delta_\perp(e) + \delta_\perp(r) + \delta_\perp(b) \\ &= \delta_\perp(e) + \delta_\perp(b.r). \end{aligned}$$

Since $a.e.r$ and $b.r$ are circuits, $\delta_\perp(a.e.r) \in \mathbb{N}$ and $\delta_\perp(b.r) \in \mathbb{N}$ according to Prop. 1, and thus $\delta_\perp(e) \in \mathbb{N}$. Since $0 < \delta_\perp(e) \leq 1$, it yields $\delta_\perp(e) \in \{0, 1\}$.

It remains to prove that $h_{\delta_{\perp}} = 0$. Let v be a vertex and c be a δ_{\perp} -shortest path from v^* to v . One computes $h_{\delta_{\perp}}(v) = \delta_{\perp}(c) = \psi(c) - h_{\psi}(v)$. Since, according to Prop. 2, c is a ψ -shortest path too, $h_{\psi}(v) = \psi(c)$ and thus $h_{\delta_{\perp}}(v) = 0$. \square

Proof of Proposition 5:

Let $h = \max(h_{\delta}, h_{\delta'})$. Let ψ be defined on an edge $e = (v, v') \in I$ by:

$$\psi(e) = \delta_{\perp}(e) + (h(v') - h(v)),$$

where δ_{\perp} is defined in Theorem 1. We prove that ψ is a binary counter whose height function is h . The proof is similar for the $\min(h_{\delta}, h_{\delta'})$ case.

Since the sum $(h(v') - h(v))$ vanishes when taken over the edges of a cell-circuit (term by term cancellation), ψ is, as δ_{\perp} , a counter. Since δ and δ' take only entire values, so does $h = \max(h_{\delta}, h_{\delta'})$, and thus ψ (δ_{\perp} being binary). Hence, proving that $0 \leq \psi \leq 1$ will ensure that ψ is a binary counter. Let $e = (v, v')$ be an edge. According to Theorem 1, $\delta_{\perp}(e)$ can be written in two ways: $\delta_{\perp}(e) = \delta(e) + h_{\delta}(v) - h_{\delta}(v') = \delta'(e) + h_{\delta'}(v) - h_{\delta'}(v')$. This allows two different writings of $\psi(e)$, and a case-study then gives:

$h(v)$	$h(v')$	$\psi(e)$
$h_{\delta}(v)$	$h_{\delta}(v')$	$\delta(e)$
$h_{\delta}(v)$	$h_{\delta'}(v')$	$\delta(e) \leq \psi(e) \leq \delta'(e)$
$h_{\delta'}(v)$	$h_{\delta}(v')$	$\delta'(e) \leq \psi(e) \leq \delta(e)$
$h_{\delta'}(v)$	$h_{\delta'}(v')$	$\delta'(e)$

Since $0 \leq \delta(e), \delta'(e) \leq 1$, one has $0 \leq \psi(e) \leq 1$.

It remains to verify that the height function of ψ is h . Let v be a vertex and c a shortest path from v^* to v . c being a shortest path for ψ , $h_{\psi}(v) = \psi(c) = \delta_{\perp}(c) + h(v) - h(v^*) = \delta_{\perp}(c) + h(v)$. c being a shortest path for δ_{\perp} , $\delta_{\perp}(c) = h_{\delta_{\perp}}(v) = 0$ (recall that $h_{\delta_{\perp}}$ is constant and equal to zero). It follows that $h_{\psi}(v) = h(v)$, and thus the result. \square

Proof of Theorem 3:

Let δ and δ' be two binary counters such that $\delta \preceq \delta'$ and $\delta \neq \delta'$. Among the nodules A such that $h_{\delta}(A) < h_{\delta'}(A)$, let A_0 be one with $h_{\delta'}(A_0)$ maximum. If A_0 has an incoming edge e_0 such that $\delta'(e_0) = 0$, let A_1 be the nodule e_0 comes from. One has: $h_{\delta'}(A_0) \leq h_{\delta'}(A_1) + \delta'(e_0) = h_{\delta'}(A_1)$. Iterating this process, we obtain a sequence of nodules $A_0, A_1, \dots, A_k, \dots$ such that $h_{\delta'}(A_i) \leq h_{\delta'}(A_{i+1})$ and there is an edge e_i from A_{i+1} to A_i with $\delta'(e_i) = 0$. If in this sequence $A_j = A_i$ for some $i < j$, then the union of A_i, A_{i+1}, \dots, A_j would itself be a nodule (thanks to the edges $e_i, e_{i+1}, \dots, e_{j-1}$): maximality of nodules for inclusion avoids this. Hence the process is finite: let A_k be the last nodule obtained. $A_k \neq A^*$ since $0 \leq h_{\delta}(A_0) < h_{\delta'}(A_0) \leq h_{\delta'}(A_k)$, and each incoming edge of A_k is valued to 1 by δ' : we thus can do a decreasing flip on A_k (see Fig. 8).

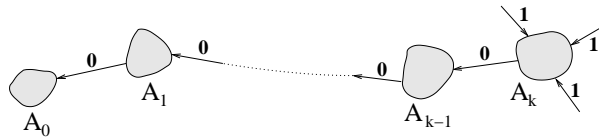


Fig. 8. Search, backward from A_0 , of a nodule to do a decreasing flip.

Let δ'' the binary counter obtained by performing this decreasing flip on A_k . We prove that $\delta \preceq \delta''$, that is, $h_\delta(A_k) \leq h_{\delta''}(A_k)$ since the flip modifies only the height of A_k . More precisely, since $h_{\delta''}(A_k) = h_{\delta'}(A_k) - 1$, it suffices to prove that $h_\delta(A_k) < h_{\delta'}(A_k)$. We already know that $h_\delta(A_k) \leq h_{\delta'}(A_k)$. Suppose that $h_\delta(A_k) = h_{\delta'}(A_k)$ to obtain a contradiction.

We first prove that there exists i_0 , $0 \leq i_0 \leq k$ such that $h_\delta(A_{i_0}) = h_{\delta'}(A_{i_0}) = h_{\delta'}(A_0)$. One has $h_\delta(A_0) < h_{\delta'}(A_0) \leq h_{\delta'}(A_k) = h_\delta(A_k)$. If $h_{\delta'}(A_k) = h_{\delta'}(A_0)$, $i_0 = k$ suits. Otherwise, let i_0 be the greatest index such that $h_{\delta'}(A_{i_0}) = h_{\delta'}(A_0)$. By this choice, $h_{\delta'}(A_{i_0+1}) > h_{\delta'}(A_{i_0}) = h_{\delta'}(A_0)$. Thus $h_\delta(A_{i_0+1}) < h_{\delta'}(A_{i_0+1})$ is impossible, since A_0 is of **maximum** δ' -height among the nodules A such that $h_\delta(A) < h_{\delta'}(A)$. Hence $h_\delta(A_{i_0+1}) \geq h_{\delta'}(A_{i_0+1})$, and since $\delta \preceq \delta'$ it yields $h_\delta(A_{i_0+1}) = h_{\delta'}(A_{i_0+1})$. But it is easily seen that two nodules linked by an edge have heights that differ by 0 or 1: applied to nodules A_{i_0} and A_{i_0+1} (linked by e_{i_0}), knowing that $h_\delta(A_{i_0+1}) = h_{\delta'}(A_{i_0+1})$ and $h_\delta(A_{i_0}) \leq h_{\delta'}(A_{i_0})$ (since $\delta \preceq \delta'$), this proves $h_\delta(A_{i_0}) = h_{\delta'}(A_{i_0})$, that is, i_0 suits.

Let us then prove that the existence of such an index i_0 leads to a contradiction. Let c be a shortest path from v^* to A_{i_0} . The edges e_0, \dots, e_{i_0-1} ensure that there exists a path c_{i_0} from A_{i_0} to A_0 such that $\delta'(c_{i_0}) = 0$ (Figure 9). Then $\delta'(c.c_{i_0}) = \delta'(c) = h_{\delta'}(A_{i_0}) = h_{\delta'}(A_0)$. Thus $c.c_{i_0}$ is a shortest path from v^* to A_0 . In particular, $h_\delta(A_0) = \delta(c.c_{i_0}) \geq \delta(c)$. But c being a shortest path from v^* to A_{i_0} , one has $\delta(c) = h_\delta(A_{i_0}) = h'_\delta(A_0)$ (the last equality coming from the choice of i_0). Hence $h_\delta(A_0) \geq h'_\delta(A_0)$. That contradicts that the fact that A_0 was chosen in $\{A \mid h_\delta(A) < h_{\delta'}(A)\}$, and thus ends the proof. \square

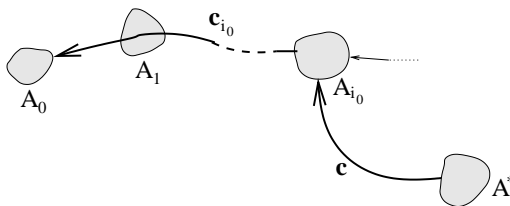


Fig. 9. The path c_{i_0} links A_{i_0} to A_0 and is such that $\delta'(c_{i_0}) = 0$.