Optimal Partial Tiling of Manhattan Polyominoes⋆

Olivier Bodini and Jérémie Lumbroso

LIP6, UMR 7606, CALSCI departement, Université Paris 6 / UPMC,
104, avenue du président Kennedy,
F-75252 Paris cedex 05, France

Abstract. Finding an efficient optimal partial tiling algorithm is still an open problem. We have worked on a special case, the tiling of Manhattan polyominoes with dominoes, for which we give an algorithm linear in the number of columns. Some techniques are borrowed from traditional graph optimisation problems.

For our purpose, a *polyomino* is the (non necessarily connected) union of unit squares (for which we will consider the vertices to be in $\mathbb{Z}^2$) and a *domino* is a polyomino with two edge-adjacent unit squares.

To solve the domino tiling problem [6,9,10,11,12,13] for a polyomino P is equivalent to finding a perfect matching in the edge-adjacent graph G_p of P's unit squares. A specific point of view to study tiling problems has been introduced by Conway and Lagarias in [4]: they transformed the tiling problem into a combinatorial group theory problem. Thurston [13] built on this idea by introducing the notion of height, with which he devised a linear-time algorithm to tile a polyomino without holes with dominoes. Continuing these works, Thiant [12], and later Bodini and Fernique [3] respectively obtained an $O(n \log n)$ algorithm which finds a domino tiling for a polyomino whose number of holes is bound, and an $O(n \log^3 n)$ algorithm with no constraint as to the number of holes. All these advances involved the "exact tiling problem".

Naturally, all aforementioned exact tiling algorithms are useless when confronted with polyominoes that cannot be perfectly tiled with dominos. Such algorithms will output only that the polyominoes cannot be tiled—and this is inconvenient because a perfect tiling is not necessarily required: we might want partial tilings with at most x untiled squares.

Thus, by analogy with matching problems in graph theory and the notion of maximum matching, it seems interesting to study the corresponding notion of optimal partial tiling. No algorithm has been specifically designed with this purpose in mind; and whether the optimal partial tiling problem for polyominoes can be solved by a linear-time algorithm has been an open problem for the past 15 years. In this paper, we are studying the partial optimal domino tiling problem on the Manhattan class of polyominoes (see figure 1).
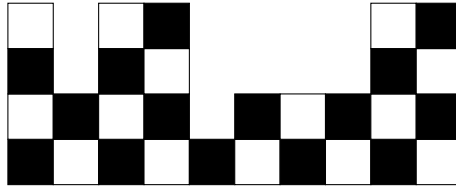

Fig. 1. A Manhattan polyomino

A *Manhattan polyomino* is a sequence of adjacent columns that all extend from the same base line (*"Manhattan"* refers to the fact that this polyomino looks like a skyline). We are exhibiting an algorithm that makes solving this problem much more efficient than solving the corresponding maximum matching problem. Indeed, the best known algorithm for maximum matchings in bipartite graphs is in $O\left(\sqrt{n}m\right)$ [7,8] (where n is the number of vertices and m the number of edges) and it yields an algorithm in $O\left(n^{3/2}\right)$ for the partial tiling problem, whereas our algorithm is linear in the number of columns. The choice of the Manhattan class of polyominoes is justified as a generalisation of the seminal result of Bougé and Cosnard [2] which establishes that a trapezoidal polyomino (i.e. a monotic Manhattan polyomino, in which the columns are ordered by increasing height) is tilable by dominoes if and only if it is balanced. The partial tiling problem on more general classes seems to be out of reach at present.

This paper first succinctly recalls some definitions on tilings, polyominoes in general and Manhahattan polyominoes in particular; we list the notations that we will use throughout the paper (section 1). Follows a presentation of the main idea of reducing the partial tiling problem to a network flow problem (section 2); we then prove this reduction to be valid (section 3). A greedy algorithm to solve this specific network problem is given (section 4), which allows us to conclude (section 5).

1   Definitions

An *optimal partial tiling* by dominoes of a polyomino P is a set $\{D_1, \ldots, D_k\}$ of vertical or horizontal dominoes placed in the discret plane such that:

  (i) $\forall i \in \{1, \ldots, k\}\, D_i \subset P$;
 (ii) the dominoes $D_1, \ldots, D_k$ have mutually disconnected interiors: for every i, j such that $1 \leqslant i \leqslant j \leqslant k$, we have $\mathring{D}_i \cap \mathring{D}_j = \emptyset$ where $\mathring{D}$ is the interior of D for the topology endowed with the euclidean distance on $\mathbb{R}^2$;
(iii) k is maximal: if k', $k' \geqslant k$, and $D'_1, \ldots D'_k$ exists with the above two conditions, then $k = k'$.

A *column* is the union of unit squares placed one on top of the other. A *Manhattan polyomino* is the connected union of columns aligned on the same horizontal line.

We will identify a Manhattan polyomino with a tuple: the sequence of the heights of all the columns of the polyomino (ordered from left to right). For example, the polyomino in figure 1 is represented as $(4, 2, 4, 4, 1, 2, 2, 2, 4, 4)$. This representation is clearly more convenient in memory space than, say, the list of all unit squares of a polyomino.

It is convenient to define *inclusion*, and we will do so using the tuple representation: polyomino $P = (p_1, p_2, \ldots, p_n)$ is said to be included in $Q = (q_1, q_2, \ldots, q_n)$ if $\forall i, p_i \leqslant q_i$ (both polyominoes can be padded with zero-sized columns if necessary).

In addition, the unit squares of a polyomino have a chessboard-like coloration; specifically[1], the unit square with position (i, j) (i.e.: unit square $(i, j) + [0, 1[^2)$ is *white* if $i + j$ is even and *black* otherwise. A column is said to be *white dominant* (or *black dominant*) if it has more white (or black) unit squares than black (or white) ones.

2   From Polyominoes to Flow Networks

In this section, without loss of generality, we consider the polyominoes are balanced (a polyomino is *balanced* if it contains the same count of white and black unit squares). Indeed, a polyomino can always be balanced by adding the necessary number of isolated unit squares of the lacking color.

Notations. Let $(h_1, \ldots, h_n)$ be a Manhattan polyomino P.

For $i \leqslant j$, we define $X(i, j)$ as the number $\min_{i \leqslant k \leqslant j} \{h_k\}$, the height of the smallest column contained in the subset $\{h_i, \ldots, h_j\}$.

We define $B(P)$ (and respectively $W(P)$) as the set of black unit squares, (or white unit squares) contained in P. Let I_P (and resp. J_P) be the set of indices of black dominant columns (and resp. white dominant columns). We define $s_1, \ldots, s_n$ as the elements of $I_P \cup J_P$ sorted in ascending order.

Let $G = (V, E)$ be a graph, and $S \subset V$. We define $\Gamma(S)$ as the subset of *neighbor vertices* of S, i.e. the vertices $y \in V$ such that there exists $x \in S$ with $(x, y) \in E$ or $(y, x) \in E$.

Construction of the flow network. For each polyomino P, we build a directed graph (*flow network*) that we call F_P:

- its vertex set is $\{s_1, \ldots, s_n\} \cup \{s, t\}$ where s and t are two additional vertices respectively called *source* and *sink* (be mindful of the fact that we use s_i both to refer to the vertices, and to the actual columns);
- its edge set E is defined by:

$$E = \{(s_i, s_{i+1}) ; i \in \{1, \ldots, n - 1\}\}$$
$$\cup \{(s_{i+1}, s_i) ; i \in \{1, \ldots, n - 1\}\}$$
$$\cup \{(s, a); a \in I_P\} \cup \{(a, t); a \in J_P\},$$

---

[1] And regardless of the basis, as both colors are symmetrical in purpose (i.e.: it doesn't matter whether a given unit square is white or black, so long as the whole polyomino is colored in alternating colors).

and each arc of E is weighted by a function c called *capacity* function defined by:

$$c(e) = \begin{cases} \left\lceil \frac{X(s_i, s_{i+1})}{2} \right\rceil & \text{if } e = (s_i, s_{i+1}) \text{ or } e = (s_{i+1}, s_i) \\ 1 & \text{if } e = (s, a) \text{ with } a \in I_P \\ 1 & \text{if } e = (b, t) \text{ with } b \in J_P \end{cases}$$

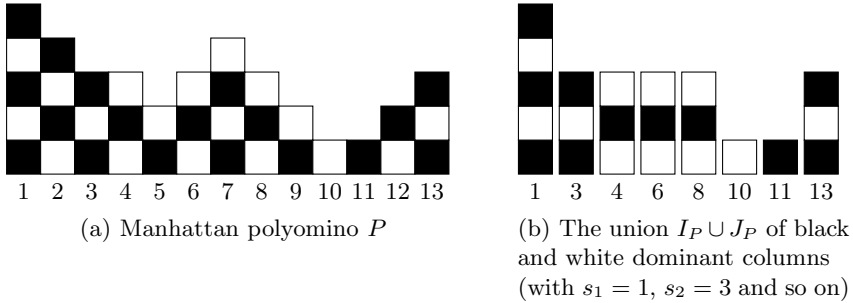where $x \mapsto \lceil x \rceil$ is the ceiling function (which returns the smallest integer not lesser than x).

(a) Manhattan polyomino P (b) The union $I_P \cup J_P$ of black and white dominant columns (with $s_1 = 1$, $s_2 = 3$ and so on)

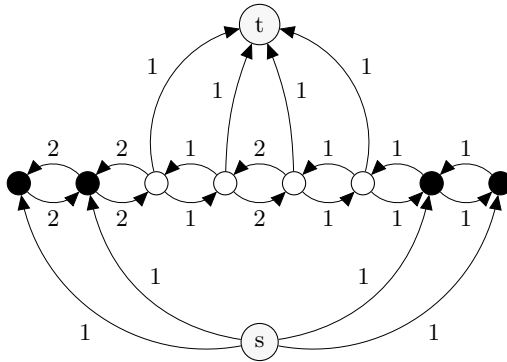**Fig. 2.** Construction of a $\{s_1, \ldots, s_n\}$ set

Fig. 3. Flow network associated with the polyomino of figure 2(a)

Example. Figures 2 and 3 illustrate the construction described above. The polyomino P in figure 2(a) contains four black dominant columns and four white dominant columns which have been extracted and are represented on figure 2(b).

This yields the set $I_P \cup J_P = \{s_1 = 1, s_2 = 3, \ldots, s_8 = 13\}$ from which we can build the flow network in figure 3.

For instance, the capacity of arc (s_1, s_2) is $c((s_1, s_2)) = 2$ because the smallest column between the indices $s_1 = 1$ and $s_2 = 3$ is the column 3 and its height is 3 (what this boils down to is $X(1, 3) = 3$).

3   Main Theorem: An Overview

As might have been evident in the previous sections, oddly-sized columns, whether white or black dominant, are what make domino tiling of Manhattan polyominoes an interesting problem: indeed, evenly-sized columns (of a polyomino comprising only such columns) can conveniently be tiled with half their size in dominoes.

3.1   Sketch of the Method

Let P be a polyomino. Our method can intuitively be summarized in several steps:

– we devise a "*planing*" transformation which partially tiles the polyomino P in a locally optimal way (i.e.: such that it is possible to obtain an optimal partial tiling for P by optimally tiling the squares which remain uncovered after the transformation); this transformation, as illustrated in figure 4(c), "*evens out*" a black dominant column and a white dominant column;
– we devise a way to construct a maximum flow problem that translates exactly how the planing transformation can be applied to the polyomino P;
– we prove that the solutions to the maximum flow problem constructed from P, and to the optimal partial tiling of P are closely linked (by a formula).

In truth, these steps are intertwined: the local optimality of the planing transformation is demonstrated using the maximum flow problem.

3.2   The Planing (or Leveling) Transformation

The planing transformation we just mentionned can be expressed as a map ϕ, which can only be applied to a polyomino P which has two *consecutive*[2] oddly-sized columns s_i, s_{i+1} of *different* dominant colors. This condition[3] can be summarized as

$$\exists i, (s_i \in I_P \text{ and } s_{i+1} \in J_P \quad \textbf{or} \quad s_i \in J_P \text{ and } s_{i+1} \in I_P).$$

Let $P = (h_1, \ldots h_n)$ be a Manhattan polyomino, and let i be the smallest integer such that s_i, s_{i+1} are two oddly-sized columns of different dominant colors and $c(s_i, s_{i+1}) \neq 0$, then $P' = \phi(P)$ is a new polyomino defined by

$$P' = (h_1, \ldots, h_{s_i-1}, a, \ldots, a, h_{s_{i+1}+1}, \ldots, h_n)$$

where $a = \min (h_{s_i} - 1, h_{s_i+1} - 2, \ldots, h_{s_{i+1}-1} - 2, h_{s_{i+1}} - 1)$, i.e.: all columns from s_i to s_{i+1} are *leveled* to height a, which is at least one unit square smaller

---

[2] By this we mean that the two dominant columns are consecutive in the $s_1, s_2, \ldots$ ordering (and this does not necessarily imply that they are adjacent in P); in figure 4(c), dominant columns s_1 and s_2 are consecutive (so are s_2 and s_3).
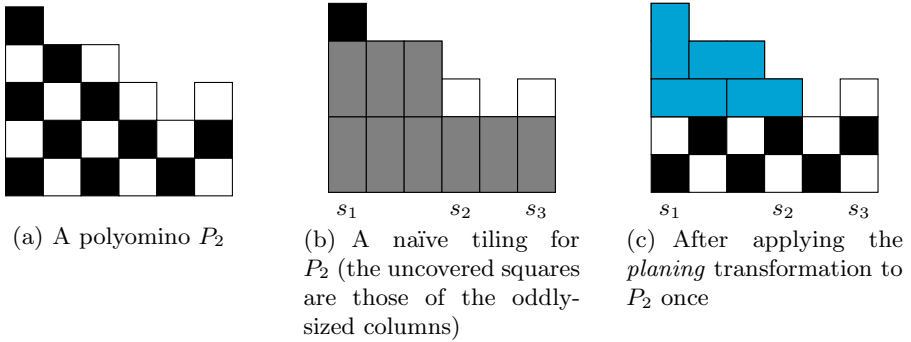
[3] Wherein it is obvious that the **or** is mutually exclusive.

(a) A polyomino P_2

(b) A naïve tiling for P_2 (the uncovered squares are those of the oddly-sized columns)

(c) After applying the *planing* transformation to P_2 once

Fig. 4. Tilings of P_2, which has three oddly-sized columns

(a) At least one unit square smaller than the smallest odd-sized column ...

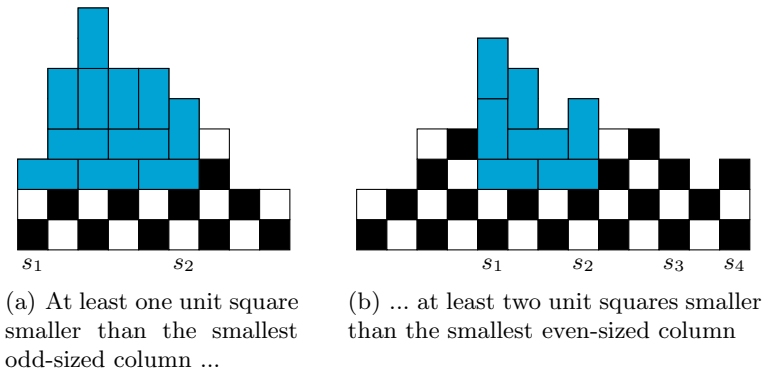(b) ... at least two unit squares smaller than the smallest even-sized column

Fig. 5. Generic examples of the planing transformation

than the smallest of s_i and s_{i+1} and two unit squares smaller than the smallest even-sized column of P from s_i to s_{i+1} .

Since we have only decreased the heights of the columns of P to obtain P', $\phi(P) = P' \subset P$ is trivial (following the definition of inclusion we gave in section 1).

We will show that this transformation is locally optimal: again, this means that if optimal tilings of $\phi(P)$ have x non-covered unit squares, then so do optimal tilings of P (the number of non-covered unit squares is invariant with regards to the transformation).

3.3   Main Theorem

Having constructed a flow network F_P from a given Manhattan polyomino P, we now show that, if we call $d(P)$ the number of non-covered unit squares in an

optimal partial tiling of P, then the value $v(P)$ of the maximum flow on F_P is such that:

$$|I_P| + |J_P| - 2v(P) = d(P).$$

The meaning of this equation is quite intuitive: it says that the number of uncovered squares $d(P)$, is exactly the number of single squares that come both from the black dominant columns I_P and from the white dominant columns J_P, from which we withdraw two squares for each of the $v(P)$ times we are able to apply the planing transformation (recall that this transformation evens out both a black dominant column and a white dominant column, hence the 2 factor).

Lemma 1. *Let P be a Manhattan polyomino. For all P', such that $P' = \phi^k(P)$ (the planing transformation applied k times to P), the following invariant holds:*

$$|I_P| + |J_P| - 2v(P) = |I_{P'}| + |J_{P'}| - 2v(P')$$

where $v(P)$ and $v(P')$ are the values of a maximum flow respectively in T_P and $T_{P'}$.

Proof (by induction on k). Let f be a maximum flow on $T_{P_{k-1}}$ which minimizes $\sum_e f(e)$ (i.e.: the flow takes the shortest path from s to t given a choice). Since it is possible to apply the planing transformation to P_{k-1}, then there must be two consecutive vertices s_i and s_{i+1} corresponding to oddly-sized columns of different dominant colors (given several such pairs, we consider that for which i is the smallest).

Let us now build T_{P_k}, the flow network associated with $P_k = \phi(P_{k-1})$: the vertices of T_{P_k} are the vertices of $T_{P_{k-1}}$ from which we remove $\{s_i, s_{i+1}\}$; the arcs of T_{P_K} are the trace of the arcs of the arcs of $T_{P_{k-1}}$ (*trace* is a non-standard term by which we mean all arcs that are incident neither to s_i nor to s_{i+1}) to which we add, should both vertices exist, the arcs (s_{i-1}, s_{i+2}) and (s_{i+2}, s_{i-1}) with capacity

$$\min\left(c\left((s_{i-1}, s_i)\right), c\left((s_i, s_{i+1})\right) - 1, c\left((s_{i+1}, s_{i+2})\right)\right).$$

Suppose s_i and s_{i+1} respectively represent a black and white dominant column (figure 6 illustrates the construction in this case); the other case is symmetrical. Then, because $\sum_e f(e)$ is minimal, maximum flow f routes a non-null amount through the path (s, s_i, s_{i+1}, t).

Thus the maximum flow f on $T_{P_{k-1}}$ induces a maximum flow f' on T_{P_k}, which is defined as the trace of f on T_{P_k} and updated, if necessary, by adding:

$$\begin{cases} f'((s_{i-1}, s_{i+2})) = f((s_{i-1}, s_i)) = f((s_{i+1}, s_{i+2})) \\ f'((s_{i+2}, s_{i-1})) = f((s_i, s_{i-1})) = f((s_{i+2}, s_{i+1})) \end{cases}.$$

Proving that f' is a maximum flow on T_{P_k} is straightforward (suppose there is a better flow, and reach a contradiction). We then immediately remark that $v(P_{k-1}) = v(P_k) + 1$; and because the planing transformation applied to P_{k-1}
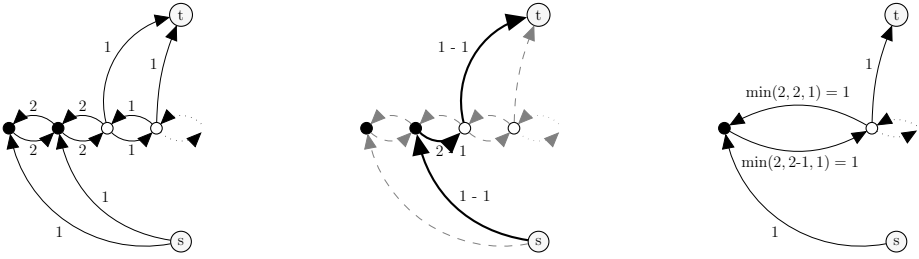
Fig. 6. Building T_{P_k} from $T_{P_{k-1}}$: a max flow for the network of figure 3 necessarily goes through the path (s, s_2, s_3, t). Here we show the impact of removing this path on the flow network's edges, vertices and capacities.

levels both a black dominant column and a white dominant column to obtain P_k, we have $|I_{P_k}| = |I_{P_{k-1}}| - 1$ and $|J_{P_k}| = |J_{P_{k-1}}| - 1$, thus:

$$\left|I_{P_{k-1}}\right| + \left|J_{P_{k-1}}\right| - 2v(P_{k-1}) = |I_{P_k}| + |J_{P_k}| - 2v(P_k),$$

from which, by induction on k, we derive the lemma's invariant.   □

Lemma 2. *Let P be a Manhattan polyomino and $v(P)$ the value of a maximum flow in T_P (cf. section 2). There is a partial tiling of P in which exactly $|I_P| + |J_P| - 2v(P)$ squares are uncovered.*

Proof. We prove the property by induction on the size of P. Let f be a maximum flow in T_P. We consider whether there are two consecutive vertices s_i and s_{i+1} which correspond to oddly-sized columns of different dominant colors, such that the flow of f going from s_i to s_{i+1} is non-null.

Either: no such vertices s_i and s_{i+1} exist[4], in which case we have $v(P) = 0$ (this would mean f is the *zero flow*). Indeed, any path from s to t would use at least one transversal arc (s_i, s_{i+1}) because, by construction, a vertex s_i cannot be simultaneously linked to s and t. We remark that the planing transformation cannot be applied. If we partially tile P, as in figure 4(b), with vertical dominoes alone, leaving one square uncovered per oddly-sized column, we obtain a partial tiling which globally leaves $|I_P| + |J_P|$ squares uncovered—thus proving our property for this case.

Or: such vertices s_i and s_{i+1} exist. We can then apply the planing transformation ϕ, $P' = \phi(P)$. By induction, the property holds for P'. As we've seen, the (union of the) squares removed by the transformation is a domino-tileable Manhattan polyomino (figure 5); and by lemma 1,

$$|I_{P'}| + |J_{P'}| - 2v(P') = |I_P| + |J_P| - 2v(P).$$

□

---

[4] This case arises when there are no oddly-sized columns, or only oddly-sized columns of a given color.

Theorem 1. *Let P be a Manhattan polyomino, and $d(P)$, the number of un-covered unit squares in an optimal partial tiling of P. The construction outlined in the proof of lemma 2 is optimal, i.e.:*

$$|I_P| + |J_P| - 2v(P) = d(p).$$

Remark 1. We are now going to work on G_P which is the edge-adjacency graph of polyomino P: the vertices of G_P are the unit squares of P, and an edge connects two vertices of G_P if and only if the two corresponding squares share an edge in P. (By contrast, T_P is the flow network constructed in section 2.)

The idea behind the proof of theorem 1 is as follow: we isolate a subset $\mathcal{B}(P)$ of black squares of P, such that the (white) neighbors (in graph G_P) to squares of $\mathcal{B}(P)$ verify the relation

$$|\mathcal{B}(P)| - |\Gamma(\mathcal{B}(P))| = |I_P| + |J_P| - 2v(P).$$

We can then conclude, using Hall's theorem, that $d(P) \geqslant |\mathcal{B}(P)| - |\Gamma(\mathcal{B}(P))|$. To construct $\mathcal{B}(P)$, we use a minimal cut[5] of graph T_P, from which we deduce a list of "*bottlenecks*". These bottlenecks mark the boundary of zones containing squares of $\mathcal{B}(P)$.

Intuitively, once bottlenecks are planed (that is to say when we have tiled them following the template given by our planing transformation), they isolate zones which have either too many white or too many black squares.

3.4   Greedy Algorithm

Having proven that our optimal partial tiling problem can be reduced to a network flow problem, we will now present an algorithm to efficiently solve this specific brand of network flow problem (indeed, a general-purpose network flow algorithm would not take into account those properties of our network which are a consequence of the way it was constructed).

We will first introduce the notion of f-tractability for two vertices of our flow network. The pair (s_i, s_j) (for which we consider that $i < j$) is said to be f-tractable if:

- s_i and s_j correspond to two oddly-sized columns of *different* dominant columns[6];
- if s_i corresponds to a black-dominant column, then the arcs (s_i, s_{i+1}), ..., (s_{j-1}, s_j) are not saturated;

---

[5] Recall that a cut of graph $G = (V, E)$ is a partition of its vertices in two subsets X_1 and X_2; let E_C be the set of edges such that one vertex is in X_1 and the other in X_2: the value of the cut is the sum of the capacity of each edge in E_C; a minimal cut is a cut which minimizes this sum. **In the case of flow networks**, a cut separates the source s from the sink t.

[6] Recall that this is *almost* the condition under which the planing transformation can be applied: the planing transformation requires, in addition, that s_i and s_j be so that $j = i + 1$ —per the ordering which have defined in section 2.
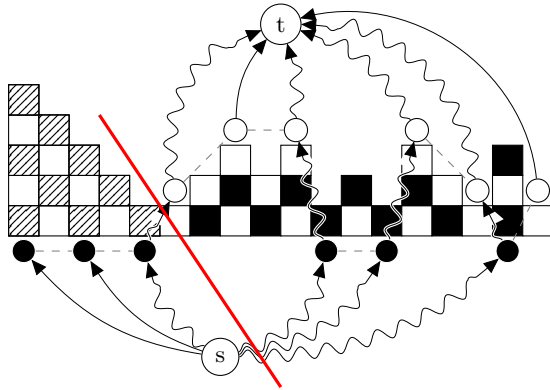
Fig. 7. A polyomino P, its flow network T_P (the transversal arcs are only hinted at, with dashed lines; the max flow is represented with snake-lines), and, in red, a minimal cut of T_P which separates P in two zones (each zone is connected here, but that may not be the case). The black striped squares on the left belong to $\mathcal{B}(P)$: in accordance with Hall's theorem, $|\mathcal{B}(P)| - |\Gamma(\mathcal{B}(P))| = 9 - 7 = 2$ is the number of uncovered squares in the optimal partial tiling.

- or else, if s_i corresponds to a white-dominant column, then the arcs (s_j, s_{j-1}), ..., (s_{i+1}, s_i) are not saturated;
- for every k, $i < k < j$, if $s_k \in I_P$ (resp. $s_k \in J_P$) then (s, s_k) (resp. (s_k, t)) is saturated.

This might seem like an elaborate notion, but in fact, it only translates whether, in terms of flow increase, we can apply our transformation.

We are going to prove that the following algorithm builds a maximum flow:

1. we begin with a null flow f;
2. while there exists two vertices s_i and s_j with $i < j$ which are f-tractable:
   (a) we take the first pair of vertices in lexicographic order which is f-tractable,
   (b) and we augment the flow f by 1 on the arcs belonging to the path $[s, s_i, s_{i+1}, \ldots, s_j, t]$ or $[s, s_j, s_{j-1}, \ldots, s_i, t]$ (depending on the dominant color of s_i);
3. we return the flow f.

Proof (Correctness of the greedy algorithm). Let (s_i, s_{i+1}) be the first pair of vertices in lexicographic order which, at the beginning of the algorithm, is f-tractable. We suppose here that s_i corresponds to a black dominant column (the other case is symetrical). We want to prove that there is a maximum flow f, which minimizes $\sum_{v \in T_P} f(v)$, such that none of the following values are null: $f((s, s_i)), f((s_i, s_{i+1})), f((s_{i+1}, t))$. If a maximum flow following this last condition does not exist, then one of those three arcs must be null and another

one must be saturated, otherwise the flow f can be augmented. Now, by the minimality of $\sum_{v \in T_P} f(v)$, it follows that (s, s_i) is not saturated (otherwise we can remove 1 to f on every arc belonging to a minimal path from s to t using (s, s_i) with non-zero flow, and add 1 to $f((s, s_i)), f((s_i, s_{i+1})), f((s_{i+1}, t))$. We obtain a maximum flow which contradicts the minimality of $\sum_{v \in T_P} f(v)$.) The arc (s_i, s_{i+1}) cannot be saturated for the same reason. So, (s_{i+1}, t) is saturated. But in this case, as (s, s_i) and (s_i, s_{i+1}) are not saturated, we can remove 1 to f on every arc belonging to a non-zero flow path from s to t using (s_{i+1}, t), and add 1 to $f((s, s_i)), f((s_i, s_{i+1})), f((s_{i+1}, t))$. We again obtain a new maximum flow with total weigth at most $\sum_{v \in T_P} f(v)$. That proves what we want. Now, the next recursive steps of the algorithm are exactly proved by the same way but on transport networks obtained by convenient diminution of the arc capacities.

To briefly but correctly analyze the complexity of this algorithm, we have to describe the representation of the entry. If the Manhattan polyomino is given by the list of its column heights, it can be stored in $O(l \log(\max(h_i)))$ bits where l is the number of columns. We can consider two steps. The first one consists in obtaining the transport network T_P from P. This can be done in one pass and we make $O(l)$ elementary operations (min, division by 2). The second part consists in solving the transport networks problem. Using stacks, the algorithm can solve the problem making a unique pass on the vertices of T_P. So, its complexity is in $O(|I_P|)$. Thus, globally the algorithm works in $O(l)$ elementary operations. If we make the hypothesis that the number of columns is approximatively the average of the height of the columns. Our algorithm is clearly sublinear in n the number of unit squares. Moreover, if we consider the complexity according to the size of the input, this algorithm is linear.

4 Conclusion

In this paper, we have illustrated that particular optimal partial domino tiling problems can be solved with specific algorithms. We have not yet been able to evidence a linear algorithm for the general problem of partial domino tiling of polyominoes without holes. This appears to be an attainable goal, as [13] has devised such an algorithm for the exact domino tiling problem.

Currently, the best known complexity for the general partial domino tiling problem is $O(n \cdot \sqrt{n})$ using a *maximum matching algorithm in bipartite graphs.* To improve this bound, it seems irrelevant to obtain an analogue to the notion of height which is a cornerstone of the tiling algorithms. Nevertheless, it can be proven that two optimal partial tilings are mutually accessible using extended flips (which contain classical flips [1,11] and taquin-like moves). This fact involves in a sense a weak form of "lattice structure" over the set of optimal partial tilings. This is relevant and could be taken into account to optimize the algorithms.

References

1. Bodini, O., Latapy, M.: Generalized Tilings with Height Functions. Morfismos 7 3 (2003)
2. Bougé, L., Cosnard, M.: Recouvrement d'une pièce trapézoidale par des dominos, C. R. Académie des Sciences Paris 315, Série I, pp. 221–226 (1992)
3. Bodini, O., Fernique, T.: Planar Tilings. In: Grigoriev, D., Harrison, J., Hirsch, E.A. (eds.) CSR 2006. LNCS, vol. 3967, pp. 104–113. Springer, Heidelberg (2006)
4. Conway, J.H., Lagarias, J.C.: Tiling with polyominoes and combinatorial group theory. JCT Series A 53, pp. 183–208 (1990)
5. Ford, L.R., Fulkerson, D.R.: Maximal Flow through a Network. Canadian Journal of Mathematics, 399 (1956)
6. Fournier, J.C.: Tiling pictures of the plane with dominoes. Discrete Mathematics 165/166, 313–320 (1997)
7. Gabow, H.N., Tarjan, R.E.: Faster scaling algorithms for network problems. SIAM J. Comput. 18(5), 1013–1036 (1989)
8. Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM J. Comput. 2(4), 225–231 (1973)
9. Ito, K.: Domino tilings on planar regions. J.C.T. Series A 75, pp. 173–186 (1996)
10. Kenyon, R.: A note on tiling with integer-sided rectangles. JCT Series A 74, pp. 321–332 (1996)
11. Rémila, E.: The lattice structure of the set of domino tilings of a polygon. Theoretical Computer Science 322(2), 409–422 (2004)
12. Thiant, N.: An O (n log n)-algorithm for finding a domino tiling of a plane picture whose number of holes is bounded. Theoretical Computer Science 303(2-3), 353–374 (2003)
13. Thurston, W.P.: Conway's tiling groups. American Mathematics Monthly 95, 757–773 (1990)

A   Annexe (Proof of Theorem 1)

So, we have proved that on the sequence of Manhattan polyominoes obtained by successive planing transformations, the function $|I_{\_}| + |J_{\_}| - 2v(\_)$ is invariant. We need another invariant to conclude that $|I_P| + |J_P| - 2v(P) \leq d(P)$. It follows from a list of "bottlenecks" which prevent the matching of columns.

We denote by $P_{\text{red}} = (h_1^{\text{red}}, ..., h_n^{\text{red}})$ the Manhattan polyomino obtained from P after $v(P)$ planing transformations.

We can suppose that the least x such that (x, x') belongs to Y_P (the set of edges of F_P that crosses the minimal cut) corresponds to a black dominant column (if this is not the case, we can inverse the colors). Let $g(x, x')$ be the smallest $k \in \{x, \ldots, x'\}$ such that $h_k = X(x, x')$. We denote by $\mathcal{G} = \{g(x, x'); (x, x') \in Y_P\}$ the *list of bottlenecks* of P and we sort the elements of $\mathcal{G}$ in ascending order.

$\mathcal{G} = (g_1^0, \ldots, g_k^0)$. Now, if i is odd (resp. even) and is the index of a white dominant column, we put $g_i = g_i^0 - 1$ (resp. $g_i = g_i^0 + 1$) otherwise we put $g_i = g_i^0$. With the hypothesis, we can observe that $(h_1, h_2, \ldots, h_{g_1})$ has more black unit squares than white ones. We put

$$H_P = B\left((h_1, h_2, \ldots, h_{g_1})\right) \cup B\left((h_{g_2}, \ldots, h_{g_3})\right) \cup \ldots$$

and

$$K_P = W\left((h_1, h_2, \ldots, h_{g_1})\right) \cup W\left((h_{g_2}, \ldots, h_{g_3})\right) \cup \ldots$$

We have $2(|H_P| - |\Gamma(H_P)|) \leq d(P)$ and we want to prove that $2(|H_P| - |\Gamma(H_P)|) = |I_P| + |J_P| - 2v(P)$. In order to do that, let P' be the Manhattan polyomino obtained from P by a planing transformation as we have processed previously. Firstly, we are going to show that $|H_P| - |\Gamma(H_P)| = |H_{P'}| - |\Gamma(H_{P'})|$ where

$$H_{P'} = B\left((h_1', h_2', \ldots, h_{g_1}')\right) \cup B\left((h_{g_2}', \ldots, h_{g_3}')\right) \cup \ldots$$

We put $\mathcal{B}(H_P)$ (resp. $\mathcal{W}(H_P)$) the number of odd black dominant columns (resp. white) in P which belongs to the columns of indices in $E = \{1, 2, \ldots g_1\} \cup \{g_2, \ldots, g_3\} \cup \ldots$. An easy observation allows us to see that

$$|H_P| - |\Gamma(H_P)| = |H_P| - |K_P| - \sum_{v \in Y_P} c(v)$$

$$= \mathcal{B}(H_P) - \mathcal{W}(H_P) - \sum_{v \in Y_P} c(v)$$

When we apply the planing transformation on P, we delete :

- a black dominant column in E and a white column in E^c and $\sum_{v \in Y_{P'}} c(v) = \sum_{v \in Y_P} c(v) - 1$,
- a black dominant and a white dominant column in E or a black dominant and a white column in E^c and $\sum_{v \in Y_{P'}} c(v) = \sum_{v \in Y_P} c(v)$.

This fact proves that $|H_{\_}| - |\Gamma(H_{\_})|$ is invariant on every sequence of Manhattan polyominos obtained by successive planing transformations.

Now, let us remark that

$$|H_{P_{\mathrm{red}}}| - |\Gamma(H_{P_{\mathrm{red}}})| = \mathcal{B}(H_{P_{\mathrm{red}}}) = |I_{P_{\mathrm{red}}}|$$

$(\mathcal{W}(H_{P_{\mathrm{red}}}) = \sum_{v \in Y_{P_{\mathrm{red}}}} c(v) = 0)$ and that $|I_{P_{\mathrm{red}}}| = |I_P| - v(P)$. Indeed, we have made $v(P)$ planing transformations and each of them reduces by 1 the number of black dominant columns.

So, $|H_P| - |\Gamma(H_P)| = |H_{P_{\mathrm{red}}}| - |\Gamma(H_{P_{\mathrm{red}}})| = |I_P| - v(P)$.

Moreover, we have assumed initially that the polyomino is balanced. So, $|I_P| = |J_P|$. Consequently, we have proved that $2(|H_P| - |\Gamma(H_P)|) = |I_P| + |J_P| - 2v(P)$. Thus, we can conclude that $|I_P| + |J_P| - 2v(P) \leq d(P)$.