RAIRO-Inf. Theor. Appl. **38** (2004) 229-243
DOI: 10.1051/ita:2004011

ALGEBRAIC TOOLS FOR THE CONSTRUCTION OF COLORED FLOWS WITH BOUNDARY CONSTRAINTS

Marius Dorkenoo[1], Marie-Christine Eglin-Leclerc[1] and Eric Rémila[1, 2]

Abstract. We give a linear time algorithm which, given a simply connected figure of the plane divided into cells, whose boundary is crossed by some colored inputs and outputs, produces non-intersecting directed flow lines which match inputs and outputs according to the colors, in such a way that each edge of any cell is crossed by at most one line. The main tool is the notion of height function, previously introduced for tilings. It appears as an extension of the notion of potential of a flow in a planar graph.

Mathematics Subject Classification. 05C25, 05C85.

Introduction

The motivations of the authors of this paper partially come from tiling problems: fix a finite set of basic tiles (which are called prototiles). Given a figure of the plane, does there exist a tiling of this figure with copies of the prototiles? In the affirmative, how can such a tiling be exhibited? Conway and Lagarias [2] defined tiling groups which are a very efficient tool for studying these problems. Their method gives a lot of necessary conditions for a simply connected figure to be tileable (see [7, 9]).

This work has been extended by Thurston [11] who introduced (in some particular cases) the notion of height function associated with a tiling. Using this

Keywords and phrases. Height function, planar flows.

[1] Groupe de Recherche en Informatique et Mathématiques Appliquées, IUT Roanne (Université Jean Monnet Saint-Etienne), 20 avenue de Paris, 42334 Roanne Cedex, France; e-mail: `dorkenoo@univ-st-etienne.fr; leclermc@univ-st-etienne.fr`

[2] Laboratoire de l'Informatique du Parallélisme, UMR 5668 CNRS-INRIA-Univ. Lyon 1-ENS Lyon, 46 allée d'Italie, 69364 Lyon Cedex 07, France; e-mail: `Eric.Remila@ens-lyon.fr`

new notion, algorithms to tile a simply connected figure with "dominoes" (*i.e.* 2×1 rectangles), or with "calissons" (*i.e.* lozenges of unit side formed from two equilateral triangles sharing an edge), have been produced.

Thurston's ideas have been taken again and generalized, providing other results about tilings on regular lattices [5,10], but these ideas did not seem easy to apply to an irregular lattice: before this paper, the only result on a partially irregular lattice was the result of Chaboud [1], which produced an algorithm for tiling with (generalized) dominoes formed from two cells of the lattice. This author only assumes that the cells of the lattice are 2-colorable and that all the cells of the lattice have the same number of edges.

In this paper, we prove that Thurston's method can be applied to solve a flow problem introduced here: given a figure F divided into cells, with colored arrows crossing the boundary of F, coming into F or going out of F, can we join coming in arrows and going out arrows by non-intersecting directed polygonal lines, with respect to colors, in such a way that any edge of any cell is crossed by at most one line? In the affirmative, how can such directed lines be exhibited? Notice that we do not assume that the structure of F has any regularity.

This paper is divided as follows: in Section 1, we present our problem and give an algebraic interpretation of it. This algebraic model permits us to apply Thurston's method: introduction of a group function and a height function (Sect. 2), creation of an order on the solutions and study of minimal solutions according to this order (Sect. 3), construction of a minimal solution (Sect. 4). In Sections 5 and 6, we present two similar problems and the elements that permit us to produce the algorithms which solve them.

1. The colored inputs-outputs problem

1.1. Cellular figure

A *cell* f is a (closed) polygon of the plane $\mathbb{R}^2$. A *planar matching* (see Fig. 1) of the edges of f is a (not necessarily perfect) matching of edges of f defined by a set of pairwise disjoint topological paths, included in the cell, linking the centers of matched edges.

A *cellular figure* F is a finite set of cells such that the intersection of two distinct cells is a set of vertices and edges of both cells. A cellular figure canonically induces an undirected graph whose vertices and edges are respectively the vertices and edges of its cells, and a closed subset of $\mathbb{R}^2$ formed by the union of its cells. We uniformly denote by F this graph and this closed subset (the context avoids ambiguity).

An edge is shared by at most two cells of F. Two distinct cells of a cellular figure are *neighbors* if they share an edge. Two distinct edges are *neighbors* if they are on the boundary of the same cell. Two distinct vertices are *neighbors* if they are the endpoints of the same edge.

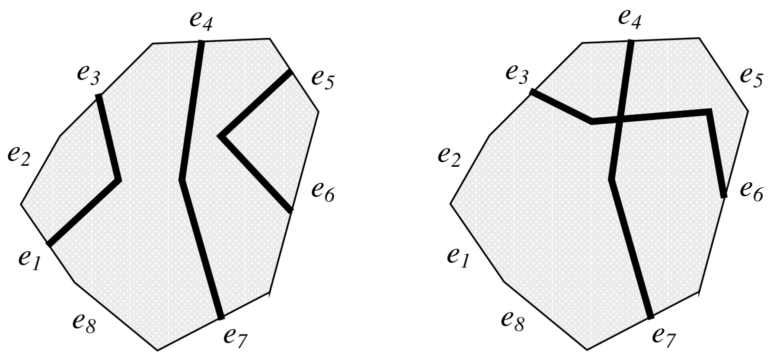

FIGURE 1. Matchings of edges of a cell: the matching $\{\{e_1, e_3\}, \{e_4, e_7\}, \{e_5, e_6\}\}$ is planar, while the matching $\{\{e_3, e_6\}, \{e_4, e_7\}\}$ is not planar.

A *path* of vertices (respectively edges, cells) is a sequence of pairwise distinct vertices (respectively edges, cells) of F such that two successive vertices (respectively edges, cells) are neighbors.

If F is simply connected (*i.e.* F and $\mathbb{R}^2 \setminus F$ are both connected), then the edges of its boundary form a sequence of edges (called a *boundary cycle*) such that two successive edges have a unique common endpoint and each edge of the boundary appears exactly once in the sequence. There are two types of boundary cycles: the *clockwise* ones and the *counterclockwise* ones. In this paper (even when it is not explicitly written), we only consider simply connected figures.

We (arbitrarily) fix an orientation for each edge of F. For each edge e, this orientation allows us to define the *terminal endpoint* of e (denoted by $t(e)$) and the *origin endpoint* of e (denoted by $o(e)$). Moreover, we can also define the *right side* of e and the *left side* of e (which are closed half-planes whose common frontier is a straight line which contains e). A cell f which has e on its boundary is *the right cell* (respectively *the left cell*) of e if the connected component of intersection of the right side (respectively the left side) of e with f is not reduced to e.

1.2. THE PROBLEM INTRODUCED

We focus on a new problem about flows, defined below. Let $C = \{c_1, ..., c_p\}$ be a set of *colors* and let B be a symbol which is not an element of C. A *boundary condition* of a figure F is a set of colored arrows which cross the boundary of F (coming into F or going out of F), in such a way that each edge of the boundary is crossed at most once. Formally, a boundary condition is a mapping μ from the set of edges of the boundary of F to the set $C \times \{in, out\} \cup \{B\}$ (see Fig. 2).

Informally, the *colored inputs-outputs problem* (CIO problem for short) is the following: given a simply connected figure F and a boundary condition of F, can we join all the coming in arrows with all the going out arrows by non-intersecting directed polygonal lines in the interior of the support of F, with respect to the colors (*i.e.* each line joins two arrows of the same color), in such a way that each edge of a cell is crossed by at most one line? And, when it is possible, how can one exhibit a solution?

Formally, the problem is to construct (if it is possible) a set $\{P_1, P_2, \dots, P_pi\}$ of *non-intersecting* paths of edges such that:

- for each path $P_j = (e_{j,0}, e_{j,1}, \dots, e_{j,q_j})$, the edges $e_{j,0}$ and, e_{j,q_j} are both on the boundary of F and there exists a color c_i such that $\mu(e_{j,0}) = (c_i, in)$ and $\mu(e_{j,q_j}) = (c_i, out)$;
- for each edge e of the boundary such that $\mu(e) \neq B$, there exists a path P_j whose first or last element is e;
- for each cell f of F, the matching M_f of edges of f induced by the paths (*i.e.* two edges are matched in M_f if they are consecutive in a path P_j) is planar.

With the notations above, a path P_j is called a *(directed) flow line*. The *color* of P_j is c_i. This color is also given to all the edges forming P_j.

Let $(e_{j,k}, e_{j,k+1})$ be consecutive edges of P_j and let f be the unique cell such that $e_{j,k}$ and $e_{j,k+1}$ are edges of f. We say that the flow line P_j *comes into* f *through* $e_{j,k}$, and *goes out of* f *through* $e_{j,k+1}$.

An edge e is *crossed by a flow line from left to right* if either e has a right cell f and a flow line comes into f through e, or e has no right cell f and a flow line goes out of the left cell of e through e.

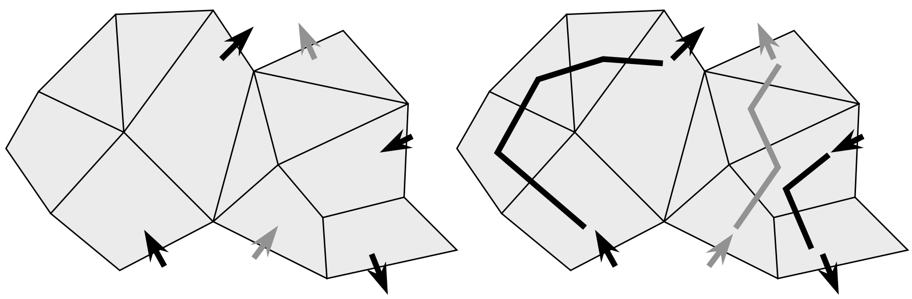

Figure 2. An instance of the CIO problem and a corresponding solution.

1.2.1. *Motivations*

The applicative motivations are very clear: imagine that a planar ground has to be crossed by pipes for water, gas, or anything else, and that incoming and outgoing places are fixed. How can the pipes be placed? Notice that these pipes have a fixed thickness or that, for safety, two distinct pipes cannot be too close one to the other. The above problem is a discretization of this concrete situation, the conditions of thickness or safety being interpreted by the condition about the crossing of each edge by at most one line. Similar problems arise naturally in the design of electronic processors.

1.2.2. *Extended solution*

Informally, an *extended solution* of the CIO problem is obtained by adding colored directed cycles, called *flow cycles*, to a (classical, as defined above) solution of the CIO problem in such a way that each edge remains crossed at most once. We will see later that, given an extended solution, one can easily find a (classical) solution of the CIO problem.

Precisely, such a flow cycle is defined as a path $(e_0, e_1, \ldots, e_q, e_{q+1})$ of edges of F such that $e_i = e_j$ and $i \neq j$ if and only if $\{i, j\} = \{0, q+1\}$. A color is given to each flow cycle, and for each cell, the matching induced by flow lines and flow cycles remains planar.

1.3. Algebraic encoding

The first step towards solving the CIO problem is to give an algebraic translation of it. We introduce the free group generated by the set $C = \{c_1, ..., c_p\}$ of colors. With the classical notations, $G = \langle c_1, ..., c_p | \rangle$ (see for example [8] for basic definitions of group theory).

A *labeling* of the (undirected) edges of F is a function (denoted by lab) from the set of edges of F to G. For each cell f, and each edge e of the boundary of f, one defines $lab_f(e)$ as follows: $lab_f(e) = lab(e)$ if f is the right cell of e, and $lab_f(e) = lab(e)^{-1}$ otherwise. For each vertex v which is an endpoint of e, we also define $lab_v(e)$ by: $lab_v(e) = lab(e)$ if $v = t(e)$, and $lab_v(e) = lab(e)^{-1}$ otherwise. For each cell f, and each vertex v of the contour of f, the *clockwise contour element* of f starting in v (which is denoted by $g_{lab}(f, v)$) is defined as follows: let $e_1, e_2, ..., e_q$ be the sequence of successive edges in a clockwise boundary cycle of the cell f starting and finishing at v, we have $g_{lab}(f, v) = lab_f(e_1)lab_f(e_2)...lab_f(e_q)$.

1.3.1. *From a solution to a labeling of edges*

Given an extended solution S, each edge of F can be labeled by an element of G, as follows (see Fig. 3): let $lab_S(e)$ denote the label given to the edge e:

- if the edge e is crossed by a flow line (or cycle) of color c_i from left to right, then $lab_S(e) = c_i$;
- if the edge e is crossed by a flow line (or cycle) of color c_i from right to left, then $lab_S(e) = c_i^{-1}$;
- if the edge e is not crossed, then $lab_S(e) = 1_G$ (the unit element of G).

Hence $lab_{S,f}(e) = c_i$ (respectively c_i^{-1}) if and only if a flow line of color c_i comes into f (respectively goes out of f) through the edge e.

Remark that, for each cell f and each vertex v of f, $g_{lab_S}(f, v) = 1_G$ since the matching induced on the edges of f is planar: following the boundary cycle, we necessarily consecutively meet two matched edges, which induces a simplification on the word $lab_{S,f}(e_1)lab_{S,f}(e_2)...lab_{S,f}(e_q)$. Once this simplification is done, the remaining word also corresponds to a planar matching, thus another simplification can be done, and so on until no edge is matched.

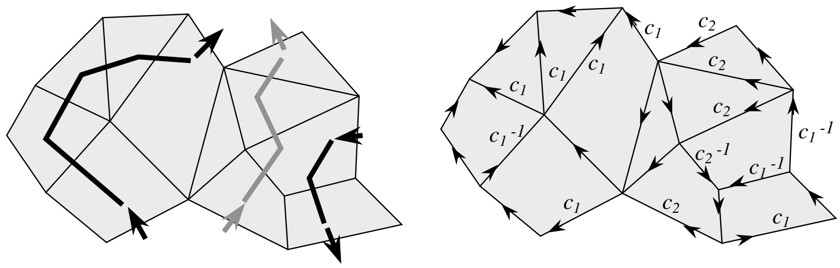

Figure 3. A solution and the corresponding labeling of edges (the 1_G labels are not indicated).

1.3.2. *From a labeling of the edges to a solution*

Conversely, assume that a labeling (denoted by lab) of the edges of F is given, in such a way that:

- each edge of F is labeled by an element of the set $L_C = \{c_1, c_2, \ldots, c_p, c_1^{-1}, c_2^{-1}, \ldots, c_p^{-1}, 1_G\}$;
- for each edge e of the boundary of F, $lab(e)$ is the label induced by the boundary condition: let f be the unique cell which has e for edge; $lab(e) = 1_G$ if $\mu(e) = B$, $lab(e) = c_i$ if either $\mu(e) = (c_i, in)$ and the cell f is the right cell of e or $\mu(e) = (c_i, out)$ and the cell f is the left cell of e, and $lab(e) = c_i^{-1}$ otherwise;
- for each cell f and each vertex v of the boundary of f, $g_{lab}(f, v) = 1_G$.

Such a labeling canonically gives a (possibly extended) solution of the CIO problem, using the following routine:

Routine. For each cell f, choose a vertex v of its contour. We use a stack *Stack* of labels which, for initialization, is empty. When *Stack* is not empty, the element element at the top is denoted by g.

Starting from v, follow the clockwise contour of f and successively, for each edge e, read $lab_f(e)$. If *Stack* is not empty and $lab_f(e) = g^{-1}$, then the edge e is matched with the edge e' which had previously forced g to be placed at the top of the stack.

Otherwise, if moreover $lab_f(e) \neq 1_G$, place $lab_f(e)$ at the top of the stack.

Informally, this routine searches for consecutive edges (ignoring edges labeled by 1_G) of inverse labels and matches these edges. These instructions are repeated, ignoring previously matched edges, until all edges are matched. This matching is clearly planar and canonically induces a solution of the CIO problem.

2. Group function and height function

2.1. Group function

Proposition 2.1. *Let F be a simply connected cellular figure, v_0 be a fixed vertex of the boundary of F and S be an extended solution. There exists a function $func_S$ (called a* group function*) from the set of vertices of cells of F to G such that, for each edge e, $func_S(t(e)) = func_S(o(e))lab_S(e)$.*

Moreover, there exists a unique function $func_{S,v_0}$ satisfying the property above such that $func_{S,v_0}(v_0) = 1_G$.

This proposition is a particular case of the following theorem (with the free group G), which is an extension of a theorem of Conway and Lagarias about tilings [2].

Theorem 2.2. *Let Γ be a simply connected cellular figure with directed edges, v_0 be a vertex of Γ and H be a group.*

Assume that we are given a labeling lab from edges of Γ to H, such that for each cell f of Γ and each vertex v of the contour of f, $g_{lab}(f, v) = 1_G$. There exists a function $func_0$ from the set of vertices of cells of Γ to H such that for each edge e, $func_0(t(e)) = func_0(o(e))lab(e)$.

Moreover, there exists a unique function $func_{v_0}$ satisfying the property above such that $func_{v_0}(v_0) = 1_G$.

Proof. We prove the existence by induction on the number of cells. If Γ is reduced to a unique cell, then the result is obvious, from the hypothesis on g_{lab}.

Note that if $func_0$ is a group function and u is an element of H, then the function $func_u$ defined by $func_u(v) = ufunc_0(v)$ is also a group function. This yields that, if a group function exists, then for each vertex w, there exists a group function $func_w$ such that $func_w(w) = 1_H$.

Now assume that Γ has at least two cells. Then there exists a path P_0 of vertices of Γ joining two vertices of the boundary of Γ, whose edges are not edges of the boundary of F. Such a path canonically defines two connected subfigures Γ_1 and Γ_2 of Γ such that Γ_1 and Γ_2 only share the path P_0. By induction hypothesis, there exists a group function $func_1$, defined on Γ_1, and a group function $func_2$ defined on Γ_2 .

Let v_{aux} be any vertex of p_0. From the remark above, the functions $func_1$ and $func_2$ can be chosen in such a way that $func_1(v_{aux}) = func_2(v_{aux}) = 1_H$ (which yields that $func_1(v) = func_2(v)$ for each vertex of P_0). In such a case, we can define the function $func_0$ on the vertices of Γ by: $func_0(v) = func_1(v)$ if v is a vertex of Γ_1, and $func_0(v) = func_2(v)$ otherwise.

The uniqueness part of the theorem is obvious, since Γ is connected. □

The proposition below gives a characterization of group functions.

Proposition 2.3. *Let $func$ be a function from the set of vertices of cells of F to G. For each edge e, we define $lab_{func}(e) = func((o(e))^{-1}func(t(e))$.*

If the function lab_{func} satisfies the conditions stated in Section 1.3.2, then there exists a (possibly extended) solution S such that $func_S = func$.

Proof. Obvious. □

For the following, we fix a vertex v_0 of the boundary of F. Given an extended solution S, $func_S$ denotes the unique function defined by Proposition 1, such that $func_S(v_0) = 1_G$. This convention yields that, for each vertex v of the boundary of F and for each pair (S, S') of solutions satisfying the same boundary condition, $func_S(v) = func_{S'}(v)$.

2.2. Height function

As each finitely presented group G can be represented by a directed labeled graph (called the *Cayley graph* of G) whose vertices are the elements of G, such that two elements g and g' are joined by a directed edge labeled by c_i, from g to g', if $g' = gc_i$ (thus, the set of labels is C) (see Fig. 4).

Since G is free, its Cayley graph is an infinite tree, which yields that one can construct an integer height function h from G to the set $\mathbb{Z}$ of integers, using, for example, the following conditions:

- for each integer n of $\mathbb{Z}$, $h(c_1^n) = n$;
- each element g of G has exactly one neighbor g' (called the *father* of g) such that $h(g') = h(g) - 1$;
- the other neighbors g'' of g (which are called the *sons* of g) are such that $h(g'') = h(g) + 1$.

Definition 2.4. The *height function* h_S induced by the extended solution S is a function from the set of the vertices of F to the set $\mathbb{Z}$ of integers such that, for each vertex v of F, $h_S(v) = h(func_S(v))$.

Note that,

- for each pair (v, v') of endpoints of a same edge e, $|h_S(v) - h_S(v')| \leq 1$;
- for each vertex v of the boundary of F and for each pair (S, S') of solutions, $h_S(v) = h_{S'}(v)$.

3. Order on the solutions

We say that a solution S is *lower than* a solution S' if for each vertex v of F, $h_S(v) \leq h_{S'}(v)$. In this way we define a (partial) order on the set of the extended solutions of a fixed instance of the CIO problem.

3.1. Local flips

Let v be a vertex of a cell of F and let S be a (possibly extended) solution satisfying the boundary condition.

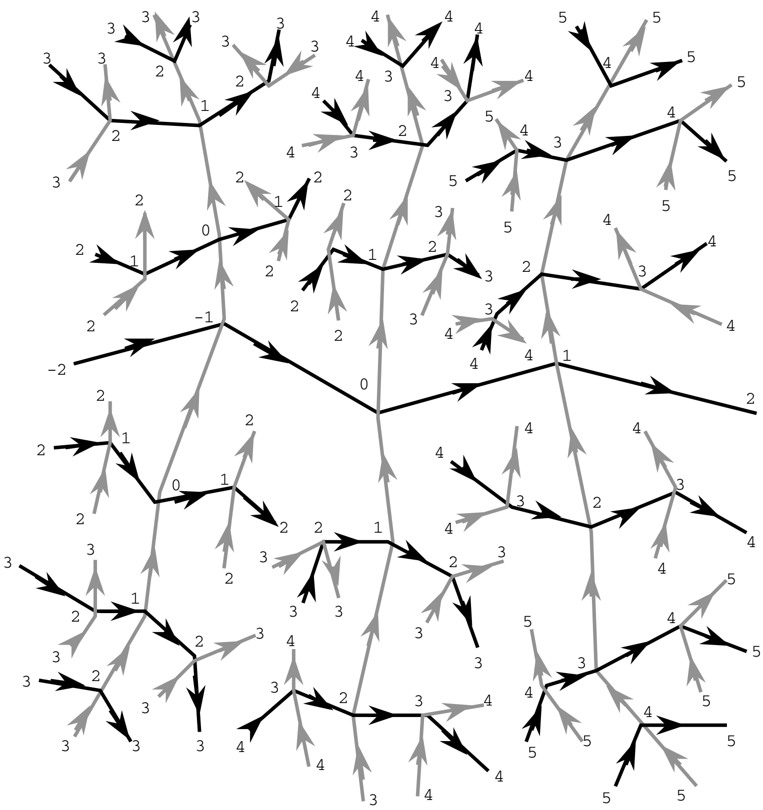

FIGURE 4. The Cayley graph of G with an induced height function.

Assume that v is not on the boundary of F and that there exists a fixed label g of $L_C \setminus \{1_G\} = \{c_1, c_2, \dots, c_p, c_1^{-1}, c_2^{-1}, \dots, c_p^{-1}\}$ such that the set of edges with v as endpoint can be divided into two sets:

- the set $E_{S,g}$ of the edges e such that $lab_{S,v}(e) = g$;
- the set $E_{S,1_G}$ of the edges e' such that $lab_{S,v}(e') = 1_G$.

Another solution S' satisfying the same boundary condition can be deduced from S, by changing only the labels of the edges of $E_{S,g} \cup E_{S,1_G}$ (see Fig. 5). Precisely, S' is defined as follows:

- for each edge e of the set $E_{S,g}$, $lab_{S',v}(e) = 1_G$;
- for each edge e of the set $E_{S,1_G}$, $lab_{S',v}(e) = g^{-1}$;
- for any other edge e, $lab_{S',v}(e) = lab_{S,v}(e)$.

The transformation described above is called a *flip* centered in vertex v. Notice that $|h_S(v) - h_{S'}(v)| = 1$, and, for each vertex w such that $w \neq v$, we have: $h_S(w) = h_{S'}(w)$. This yields that S and S' are consecutive elements of the order defined above (if, for example, $h_S(v) = h_{S'}(v)+1$, then S is an immediate successor

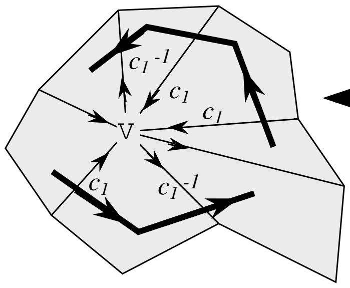

FIGURE 5. An example of flip.

of S'). Notice that, conversely, S can also be obtained from S' by a flip centered at v.

3.2. Key-lemma and consequences

Since the set of solutions is finite, there exists at least one minimal solution S_0 (*i.e.* a solution without predecessor). We now study the properties of such a solution.

Lemma 3.1 (key-lemma)**.** *Let S_0 be a minimal solution, let h_0 denote the height function induced by S_0 and let w be an interior vertex of F. There exists a neighbor w' of w such that $h_0(w') = h_0(w) + 1$.*

Proof. Let $func_{S_0}$ denote the group function induced by S_0 and assume that the lemma is false for a vertex w. Let $fath_0(w)$ denote the father of $func_{S_0}(w)$ and let $g_0(w)$ denote the element of G such that $fath_0(w)g_0(w) = func_{S_0}(w)$ (which yields that either $g_0(w) = c_i$ or $g_0(w) = c_i^{-1}$ for some integer i).

Let w' be a neighbor of w and let e denote the edge linking w' and w. Since, by hypothesis, $h_0(w') \leq h_0(w)$, we have two alternatives:

- $h_0(w') = h_0(w)$. This yields that $func_{S_0}(w') = func_{S_0}(w)$. Thus we have: $lab_{s_0,v}(e) = 1_G$;
- $h_0(w') = h_0(w) - 1$. Thus, necessarily, $func_{S_0}(w') = fath_0(w)$, whence $lab_{s_0,v}(e) = g_0(w)$.

Thus a flip centered in w can be executed. A solution S_1, which is lower than S_0, is created in this way, which contradicts the minimality of S_0. □

Note that the key-lemma implies that for a minimal solution all the vertices of F of maximal height are on the boundary of F.

Corollary 3.2. *Let M_0 denote the maximal height reached by h_0 on the boundary of F. For each vertex v of a cell of F, $h_0(v) \leq M_0$. Moreover, if $h_0(v) = M_0$, then v is on the boundary of F.*

Proof. If this corollary is false, then there exists a interior vertex v_0 such that $h_0(v_0) \geq M_0$. From the key-lemma, the vertex v_0 has a neighbor v_1 such that

$h_0(v_1) \geq M_0 + 1$. Repeating this argument, one constructs an infinite sequence $(v_i)_{i\in\mathbb{N}}$ of pairwise distinct vertices, which contradicts the fact that the cellular figure is finite. $\square$

Definition 3.3. Let M_0 denote the maximal height reached by h_0 on the boundary of F. We define recursively the sequence $(\phi_k)_{k\in\mathbb{N}}$ of sets of vertices by: $\phi_0 = \{v|h_0(v) = M_0\}$ and, for $k > 0$, v is element of ϕ_k if and only if v is not an element of $\cup_{i=0}^{k-1}\phi_i$ and one the following alternatives holds:

- the vertex v is on the boundary of F and $h_0(v) = M_0 - k$;
- there exists a neighbor v' of v in ϕ_{k-1}.

Corollary 3.4. *For each integer k, we have $\phi_k = \{v|h_0(v) = M_0 - k\}$.*

Proof. By induction on k. The result is true for $k = 0$, by definition.

Now, fix a positive integer k and assume that the result is true for each integer k' such that $0 \leq k' < k$. Let v be a vertex of ϕ_k such that there exists a neighbor v' of v in ϕ_{k-1}. This yields that $|h_0(v) - h_0(v')| \leq 1$. Since v is not element of $\cup_{i=0}^{k-1}\phi_i$, we have $h_0(v) < M_0 - k + 1$. Moreover, by the induction hypothesis, $h_0(v') = M_0 - k + 1$. Thus, we necessarily have: $h_0(v) = M_0 - k$.

Conversely, if v is an interior vertex of F such that $h_0(v) = M_0 - k$, then, from the key-lemma and the induction hypothesis, there exists a neighbor v' of v in ϕ_{k-1}. $\square$

4. Algorithm

We can now give an algorithm which either gives a solution of the CIO problem (if such a solution exists) or indicates that there is no solution. When a solution exists, this algorithm constructs a minimal solution S_0.

4.1. Description

- **Input:** A simply connected cellular figure with a boundary condition.
- **Initialization:** Arbitrarily assign a direction to each edge of F and fix a vertex v_0 of the boundary of F. Then, from v_0, compute $func_{S_0}(v)$ and $h_0(v)$ for each vertex v of the boundary of F, with respect to the boundary conditions.

  Let M be an integer variable which, for initialization, is equal to the highest value M_0 obtained for h_0 on the boundary of F.
- **Main loop:** If there exists a vertex v, for which the group function and the height function are not previously defined, such that v has a neighbor w for which those values have been defined and $h_0(w) = M$, then define $func_{S_0}(v)$ as the father of $func_{s_0}(w)$. Otherwise, decrease M of one unit.

  These instructions are repeated until the group function and the height function are defined for each vertex of F.

- **Control:** For each edge e of a cell of F, compute the value $g = (func_{S_0}(o(e))^{-1} func_{S_0}(t(e))$ and check that g is a label of L_C (otherwise, there is no solution).
- **Finalization:** Using the routine described at the end of Section 2, for each cell of F, construct an extended solution.

If we want a classical solution (with no cycle), it suffices to destroy the cycles as follows:

- **Deletion of cycles:** For each edge e on the boundary such that $lab_{S_0}(e) \neq 1_G$, successively confirm the labels of all the edges of the path of edges induced by S_0 beginning (or finishing) in e.
  Afterwards, change all the non-confirmed inputs into 1_G.

4.2. Analysis

4.2.1. *Correctness*

First, notice that, from the corollaries of the key-lemma, if a solution exists then, for each vertex v, the value given in the loop is the only possible for $func_{S_0}(v)$ (which yields that there exists a unique minimal solution).

If no contradiction is detected during the control, we can claim that the value given to each vertex induces a solution, from Proposition 2.3.

4.2.2. *Time complexity*

Let n denote the number of vertices and let m denote the number of edges of F.

The initialization costs at most $O(m)$ time units. The execution of the main loop costs $O(m)$ time units: the neighborhood of each vertex of height M has to be explored to define new values of the group function, thus each edge is used twice (once in each direction).

The execution of the control costs $O(m)$ time units.

The execution of the routine of Section 2 for each cell of F costs $O(m)$ time units, since each interior edge is used twice and each edge of the boundary is used once, according to the number of cells which share the edge.

The deletion of cycles costs $O(m)$ time units since each edge is used at most once during the confirmation process.

Thus, the algorithm has a time complexity in $O(m)$. We recall that, from planarity, $m \leq 3n + 6$, thus the algorithm also has a time complexity in $O(n)$.

In the following sections we give some examples of similar problems which can be solved in a very similar way to the CIO problem, with an $O(n)$ algorithm. We give the principal elements, which allows the reader to precisely construct algorithms and their proofs of correctness.

5. Colored inputs problem

The colored inputs (CI for short) problem is the same as the CIO problem, ignoring directions: given a simply connected cellular figure F with colored line segments which cross the boundary, can we match these inputs joining them by non-intersecting (undirected) colored lines in the interior of F, with respect of the colors, in such a way that each edge is crossed by at most one line, and, when it is possible, how can a solution be exhibited?

5.1. Algebraic interpretation

This problem can be solved using the group $G' =< c_1, c_2, ...c_p | c_1^2 = c_2^2 = ... = c_p^2 = 1 >$ (*i.e.* the group generated by the colors such that any equality true in G' can be deduced from $c_1^2 = c_2^2 = ... = c_p^2 = 1_{G'}$). This group permits to translate the CI problem in a problem of labeling of the edges of F, similarly as for the CIO problem (notice that, for this problem, the edges do not have to be directed).

5.2. Group function and height function

Let g and g' be two elements of G'. If there exists an integer i such that $g' = gc_i$, then (multiplying by c_i) $g = g'c_i$. In other words, if there exists a directed edge of the Cayley graph of G', from g to g', labeled by c_i, then there exists a directed edge from g' to g labeled by c_i.

If the opposite edges with the same label of the Cayley graph of G' are replaced by an undirected edge with this label, then a tree $T_{G'}$ (with labeled edges) is constructed. This structure of tree permits to construct a height on G'. Thus, from Theorem 2.2, one can successively construct a group function and a height function on the vertices of F.

5.3. Order on the solutions

We canonically define a (partial) order on the set of the solutions of the CI problem as follows: a solution S is lower than a solution S' if each vertex v of F is lower in S than in S'. Moreover, we can define the same kind of flips.

Let v denote a vertex of $T_{G'}$. Assume that for a fixed label g, the set of the edges of which an endpoint is v can be divided into two sets:

- the set $E_{S,g}$ of the edges e labeled by g;
- the set $E_{S,1_{G'}}$ of the edges e labeled by $1_{G'}$.

As for the CIO problem, another solution S' satisfying the same boundary condition can be deduced from S by exchanging the labels. This type of flip allows to prove the key-lemma and, therefore, to obtain the algorithm.

6. Saturated problems

The saturated colored-inputs outputs (SCIO for short) problem is defined as follows: given a simply connected cellular figure F with a (coming in or going out) arrow on each edge of the boundary, can we draw non-intersecting colored directed lines and cycles in such a way that each interior edge is crossed exactly once, and arrows are joined by directed lines with respect to colors?

Notice that a solution of the SCIO problem is also an extended solution of the CIO problem. Thus the algebraic interpretation, group function and height function are those of the CIO problem. The main difference arises for the flips, since those of Section 4 are not allowed to pass from a solution of the SCIO problem to another one.

As for the examples before, we introduce some flips. Let S be a solution, g be a label and v be an interior vertex such that, for each edge of which an endpoint is v, $lab_{S,v}(e) = g$. For each other label g', another solution $S_{g'}$ can be constructed only changing the labels of the edges with v as endpoint as follows: for each edge e of this type, $lab_{S_{g'},v}(e) = g'$. This defines a flip for the SCIO problem. By extension, we state $S = S_g$, and we similarly define the auxiliary solution S_{1_G}, which is an extended solution of the CIO problem but not of the SCIO problem.

In the Cayley graph of the free group G, the elements $func_{S_{g''}}(v)$ (for g'' being either a color or the inverse of a color) are the neighbors of $func_{S_{1_G}}(v)$, which yields that there exists a unique label g_0 such that $func_{S_{g_0}}(v)$ is the grandfather of all the other elements $func_{S_{g''}}(v)$. Thus $h_{S_{g_0}}(v) = h_{S_{g''}}(v) - 2$.

This last equality guarantees that a key-lemma can be proved for the SCIO problem. Thus the algorithm of Section 5 gives a solution to this problem, if such a solution exists. It suffices to add a control to be sure that each edge is really crossed.

Note that in the particular case when the figure is a piece of the square lattice of the plane and only one color is allowed, a solution of the saturated inputs outputs problem is a solution of the ice model (or six-vertex model) of physicists [6]: an atom of oxygen is in each cell, an atom of hydrogen is on each edge and the arrows indicate the inter-atomic connections in a crystal of ice.

One can also define the saturated colored inputs problem (for undirected lines and cycles) which can be treated in a very similar way.

7. Concluding remarks

We finish this paper by pointing out three facts which seem to be the most important contributions of this paper.

1) In the special case when only one color is used, the problems seen above are flow problems (in the classical way), and the group function and height function here are the "potential function" introduced by Hassin [3,4]) to produce an algorithm of maximum flow on a planar network. Thus the algebraic notions developed in this paper are generalizations of the potential function studied before.

2) The algorithms of this paper prove that the main reason for which Thurston's method [11] holds is not the regularity of the lattice used, since in our problems one can use an irregular figure.

3) This paper makes a bridge between flows problems and some tilings problems. They both appear as problems of labeling the edges of a cellular figure by elements of a group, with local constraints.

Acknowledgements. The authors thank Sébastien Desreux for his helpful comments and improvements of the manuscript.

References

[1] T. Chaboud, *Pavages et Graphes de Cayley.* Ph.D. Thesis, École Normale Supérieure de Lyon (1995).
[2] J.H. Conway and J.C. Lagarias, Tiling with Polyominoes and Combinatorial Group Theory. *J. Combin. Theory A* **53** (1990) 183–208.
[3] R. Hassin, Maximum flows in (s, t) planar networks. *Inform. Proc. Lett.* **13** (1981) 107.
[4] R. Hassin and D.B. Johnson, An $O(nlog^2n)$ algorithm for maximum flow in undirected planar networks. *SIAM J. Comput.* **14** (1985) 612–624.
[5] C. Kenyon and R. Kenyon, Tiling a polygon with rectangles. *Proc. 33rd FOCS* (1992) 610–619.
[6] J. Kondev and Ch.L. Henley, Kac-Moody symmetries of critical ground states. *Nuclear Phys. B* **464** (1996) 540–575.
[7] J.C. Lagarias and D.S. Romano, A Polyomino Tiling of Thurston and its Configurational Entropy. *J. Combin. Theory A* **63** (1993) 338–358.
[8] W. Magnus, A. Karass and D. Solitar, *Combinatorial Group Theory.* Dover Publications, Inc. (1976).
[9] J. Propp, *A pedestrian approach to a method of Conway, or a tale of two cities.* Internal Report, Massachusetts Institute of Technology (1993).
[10] E. Rémila, Tiling a figure using a height in a tree, in *Proc. of the 7th annual ACM-SIAM Symposium On Discrete Algorithms (SODA).* SIAM eds, Philadelphia (1996) 168–174.
[11] W.P. Thurston, Conway's tiling group. *Amer. Math. Monthly* (1990) 757–773.

Communicated by Y. Métivier.
Received November 26, 2003. Accepted March 2, 2004.