On a Graph Coloring Problem Arising From Discrete Tomography

C. Bentz and M. C. Costa
CEDRIC, CNAM, Paris, France

D. de Werra
IMA–EPFL, Lausanne, Switzerland

C. Picouleau
CEDRIC, CNAM, Paris, France

B. Ries
IMA–EPFL, Lausanne, Switzerland

An extension of the basic image reconstruction problem in discrete tomography is considered: given a graph $G = (V, E)$ and a family of chains P_i together with vectors $h(P_i) = (h_i^1, \ldots, h_i^k)$, one wants to find a partition $V^1, \ldots, V^k$ of V such that for each P_i and each color j, $|V^j \cap P_i| = h_i^j$. An interpretation in terms of scheduling is presented. We consider special cases of graphs and identify polynomially solvable cases; general complexity results are established in this case and also in the case where $V^1, \ldots, V^k$ is required to be a proper vertex k-coloring of G. Finally, we examine also the case of (proper) edge k-colorings and determine its complexity status.

Keywords: discrete tomography; vertex coloring; edge coloring; family of chains; scheduling

1. INTRODUCTION

Discrete tomography deals with the reconstruction of discrete objects from their projections.

The reader is referred to the book of Hermann and Kuba [12] for an overview of problems in discrete tomography.

Here we shall consider a graph coloring problem which generalizes a basic image reconstruction problem in discrete tomography defined below.

We are given a connected graph $G = (V, E)$ and a collection $\mathcal{P}$ of p subsets P_i of vertices of G. We are also given a set of colors $1, 2, \ldots, k$ as well as a collection H of p vectors $h(P_i) = (h_i^1, \ldots, h_i^k) \in \mathbb{N}^k$ $(i = 1, \ldots p)$.

We have to find a k-partition $V^1, V^2, \ldots, V^k$ of V such that

$$\left|P_i \bigcap V^j\right| = h_i^j \quad \text{for all } i \leq p \text{ and all } j \leq k. \qquad (1)$$

This problem will be called $\Lambda(G, k, \mathcal{P}, H)$. It is clear that in this formulation the structure of G plays no role.

We shall from now on consider a family of chains μ_i in G; we will denote by P_i the (ordered) set of vertices in μ_i and the length of μ_i will be $|P_i|$. Whenever no confusion may arise, we shall identify μ_i with its vertex set P_i. We will then call $|P_i|$ the length of P_i. In the case where the structure of G plays no role, it is not restrictive to start from chains μ_i (instead of arbitrary subsets P_i of vertices as above): we can indeed link the vertices of a P_i to form a chain μ_i.

The k-partition need not be a coloring of G where adjacent vertices have different colors. We will talk indifferently of *k-partition* or *k-coloring* to describe a partition of the vertex set into k subsets (color classes); whenever we will have the usual requirement of having different colors on adjacent nodes, we will call this a *proper* k-coloring. The corresponding reconstruction problem associated to proper k-colorings will be denoted $\Lambda^*(G, k, \mathcal{P}, H)$.

Let us now consider the special case where $G = (V, E)$ is a grid graph; its vertex set is $V = \{x_{rs} | r = 1, \ldots, m;\ s = 1, \ldots, n\}$ and its edge set is

$$E = \{[x_{rs}, x_{r,s+1}] |\ s = 1, \ldots, n-1;\ r = 1, \ldots, m\} \bigcup$$
$$\{[x_{rs}, x_{r+1,s}] |\ r = 1, \ldots, m-1;\ s = 1, \ldots, n\}$$

If x_{rs} is located in row r and column s of the grid, then by taking for $\mathcal{P}$ the collection of chains $P_r = \{x_{r1}, \ldots, x_{rn}\}$ for $r = 1, \ldots, m$ and $P_{m+s} = \{x_{1s}, \ldots, x_{ms}\}$ for $s = 1, \ldots, n$, $\Lambda(G, k, \mathcal{P}, H)$ is exactly the basic image reconstruction problem in discrete tomography; here h_r^j (resp. h_{m+s}^j) is the number of occurrences of color j in row r (resp. in column s) (i.e. $(h_r^1, \ldots, h_r^k)$ and $(h_{m+s}^1, \ldots, h_{m+s}^k)$ are the horizontal and the vertical projections, respectively). This problem is also known as colored matrix reconstruction problem.

For $k = 2$ the problem consists of reconstructing a $(0, 1)$-matrix from its vertical and horizontal projections, i.e., number of occurrences of 1 in each row and in each column; this case is solved in polynomial time [15].

For $k = 4$, this problem is NP-complete [4]; for $k = 3$ the complexity status is open but some special cases were solved in polynomial time [5, 6].

In this paper we will consider some extensions and variations of this basic problem by taking more general classes of graphs G such as trees, bipartite graphs, planar graphs, cacti.

As an application of $\Lambda(G, k, \mathcal{P}, H)$ let us mention the following problem consisting in scheduling the refurbishment of the stations in a city subway network. The network is represented by a graph $G = (V, E)$ where the vertices are the stations. Each metro line is associated with a chain P_i. Assuming that the renovation operation of every single station takes 1 month, we want to schedule these operations while taking into account the following requirements: in month j, the number of stations in metro line P_i which will be closed for renovation is h_i^j. The problem of assigning a date (month) for the renovation of every station with the above constraints is precisely $\Lambda(G, k, \mathcal{P}, H)$ if the whole refurbishment has to take place in a period of k months. In some cases, it is desired to avoid closing two consecutive stations along the same metro line; the assignment of dates is then a proper k-coloring of the underlying graph G and the problem is $\Lambda^*(G, k, \mathcal{P}, H)$.

In addition to the aforementioned application, our problem may be viewed in a different context related to constraint satisfaction in logic. Essentially we are given a collection of n Boolean variables as well as a collection of clauses P_i (each one of them involves a subset of the Boolean variables). It is required to find an assignment of values True or False to each Boolean variable in such a way that in each clause P_i the number of variables with value False is exactly (or at most) a given number h_i^F. Notice that here we have a number k of colors which is 2. The general k-coloring case would then correspond to k-valued logical variables.

After preliminaries given in Section 2, we will consider the basic problem $\Lambda(G, k, \mathcal{P}, H)$ in Section 3 with the case $k = 2$ (difficult and easy cases) and the general case $k \geq 3$. Then Section 4 will be dedicated to the case of proper colorings, i.e. to $\Lambda^*(G, k, \mathcal{P}, H)$. In Section 5 we will consider line graphs. This amounts to replacing the vertex colorings by edge colorings. Again we will consider general k-colorings and also proper k-colorings. Finally, Section 6 will present a summary of the results obtained in this paper.

2. PRELIMINARIES

For graph theoretical terms not defined here, the reader is referred to [3].

In the following we assume that several basic conditions for a solution to exist are verified, in particular, $\sum_j h_i^j = |P_i|$, for all $i = 1, \ldots, p$. In addition, if we want to determine proper colorings, we have to assume that $h_i^j \leq \lceil \frac{|P_i|}{2} \rceil$ for all i, j. It follows that there is at most one color such that $h_i^j = \lceil \frac{|P_i|}{2} \rceil$ if $|P_i|$ is odd and at most two colors such that $h_i^j = \frac{|P_i|}{2}$ if $|P_i|$ is even. These colors will be called *saturating* for P_i.

We need some more definitions and notations for $\mathcal{P}$. For a family $\mathcal{P} = (P_i \mid i = 1, \ldots, p)$ of subsets P_i of a set V, we call *cover index* of $\mathcal{P}$ and denote by $c(\mathcal{P})$ the maximum number of members of $\mathcal{P}$ which may cover a single element of V (i.e. which have a nonempty intersection).

For instance, in the basic reconstruction problem of discrete tomography, we have $c(\mathcal{P}) = 2$.

A family $\mathcal{P} = (P_i \mid i = 1, \ldots, p)$ of subsets P_i of a set V is called *nested* if for any $P_i, P_f \in \mathcal{P}$, we have either $P_i \subseteq P_f$ or $P_f \subseteq P_i$ or $P_i \cap P_f = \emptyset$.

Consider now a partition of $\mathcal{P}$ into nested families. One can look for a partition into the smallest possible number of nested families. This number, denoted by $\text{Nest}(\mathcal{P})$, is called the *nesticity* of $\mathcal{P}$.

Fact 2.1. [9] *One can determine in polynomial time if for a family $\mathcal{P}$ we have $\text{Nest}(\mathcal{P}) \leq 2$.*

Proof of fact 2.1. Assign a vertex to each $P_i \in \mathcal{P}$ and link by an edge P_i and P_f whenever $P_i \cap P_f \neq \emptyset$, $P_i \not\subseteq P_f$ and $P_f \not\subseteq P_i$. The resulting graph is bipartite if and only if $\text{Nest}(\mathcal{P}) \leq 2$. ∎

Observe that $c(\mathcal{P})$ and Nest $(\mathcal{P})$ are unrelated: we may have $c(\mathcal{P}) > \text{Nest}(\mathcal{P})$ or $c(\mathcal{P}) < \text{Nest}(\mathcal{P})$. In fact, for $\mathcal{P} = (\{a, b\}, \{a, c\}, \{b, c\})$, we have $c(\mathcal{P}) = 2$, $\text{Nest}(\mathcal{P}) = 3$ and for $\mathcal{P}' = (\{a, b, c\}, \{a, b\})$, we have $c(\mathcal{P}') = 2, \text{Nest}(\mathcal{P}') = 1$.

3. ARBITRARY COLORINGS

In this section we establish some complexity results and we exhibit some cases which can be solved in polynomial time for $\Lambda(G, k, \mathcal{P}, H)$.

Notice that whenever the k-colorings are not required to be proper, we can assume that for each edge e there is at least one chain μ_i which uses e; otherwise the edge can be removed.

Notice that it may happen that we get a disconnected graph; in such a case the problem is decomposed.

We shall start with the case where we have $k = 2$ colors.

3.1. *Difficult Problems for $k = 2$*

Let us first give two statements which do not refer to the nature of the underlying graph G.

Theorem 3.1. *$\Lambda(G, 2, \mathcal{P}, H)$ is NP-complete if $\mathcal{P}$ is a 3-uniform family ($|P_i| = 3$ for $i = 1, \ldots, p$) which is 3-regular (each vertex is in exactly three P_i's).*

Proof. We use a reduction from *X3C* (exact cover by 3-sets) which is known to be *NP*-complete [8]. In $\mathcal{P}$ one has to find a set of disjoint P_i's which cover exactly the ground set. Or equivalently, we have to find a set S of vertices such that each P_i contains exactly one vertex of S.

Let $h_i^1 = 1, h_i^2 = 2$ for each P_i; then there is a partition V^1, V^2 of the ground set satisfying (1) if and only if there is a set S of vertices containing exactly one vertex of each P_i. In this case $S = V^1$ and the other vertices form V^2. ∎

Theorem 3.2. *$\Lambda(G, 2, \mathcal{P}, H)$ is NP-complete if $Nest(\mathcal{P}) = 3$.*

Proof. We use a transformation of the three-dimensional matching problem which is known to be NP-complete [8]. To state a three-dimensional matching problem, we introduce a collection of points with coordinates (α, β, γ) with $\alpha, \beta, \gamma \in \{1, 2, \ldots, q\}$ and three families formed by all disjoint chains parallel to the coordinate axes; this gives $\mathcal{P}$ with $\text{Nest}(\mathcal{P}) = 3 = c(\mathcal{P})$.

We set $h_i^1 = 1, h_i^2 = |P_i| - 1$ for each P_i in $\mathcal{P}$. Then there exists a matching of size q if and only if there exists a partition V^1, V^2 of the set of points which satisfies (1). ∎

Notice that it follows from this transformation that $\Lambda(G, k, \mathcal{P}, H)$ remains NP-complete for $k = 2$ and $c(\mathcal{P}) = 3$.

Theorem 3.3. *$\Lambda(G, 2, \mathcal{P}, H)$ is NP-complete when G is bipartite of maximum degree ≤ 4 and each color occurs at most three times in each P_i ($h_i^j \leq 3, \forall i = 1, \ldots, p, \forall j = 1, 2$) and $|P_i \cap P_f| \leq 1$ for all $1 \leq i, f \leq p$ ($i \neq f$).*

Proof. The transformation is from the NP-complete problem ONE-IN-THREE 3SAT which is defined as follows [8]:

INSTANCE: A set U of variables, a collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$ variables.

QUESTION: Is there a truth assignment for U such that each clause in C has exactly one true literal?

This problem is also NP-complete in the case where there is no negated literal.

We build a graph by associating with each variable x occurring s times vertices $x_1, x_{12}, x_2, x_{23}, x_3, \ldots, x_{s-1,s}, x_s$ and edges $[x_1, x_{12}], [x_{12}, x_2], [x_2, x_{23}], \ldots, [x_{s-1,s}, x_s]$. For each clause $c_l = \{x, y, z\}$ we know the number of occurrences of its variables in clauses $c_1, \ldots, c_{l-1}$; so assume $c_l = \{x_d, y_e, z_f\}$ which means that in c_l x has its dth occurrence, y its eth occurrence and z its fth. We introduce vertices u_l and w_l with edges $[x_d, u_l], [u_l, y_e], [y_e, w_l], [w_l, z_f]$. Clearly the graph obtained is bipartite. Now we define $\mathcal{P}$.

For each variable x, each edge $[x_1, x_{12}], [x_{12}, x_2], \ldots, [x_{s-1,s}, x_s]$ becomes a chain P_i' with $h(P_i') = (1, 1)$. For each clause $c_l = \{x_d, y_e, z_f\}$ we introduce a chain $P_i'' = \{x_d, u_l, y_e, w_l, z_f\}$ with $h(P_i'') = (3, 2)$ and also chains $P_l^* = \{u_l\}, P_l^{**} = \{w_l\}$ with $h(P_l^*) = h(P_l^{**}) = (1, 0)$.

The family $\mathcal{P}$ of chains obtained verifies clearly $|P_i \cap P_f| \leq 1$ for all $i, f \leq p$ ($i \neq f$). Furthermore, no vertex of G has degree more than 4.

If an instance of ONE-IN-THREE 3SAT has answer "yes", then assigning color 1 to vertices u_l, w_l (for all l) and to $x_1, x_2, \ldots, x_s$ if variable x is "true" or to $x_{12}, x_{23}, \ldots, x_{s-1,s}$ otherwise and giving color 2 to the remaining vertices gives a positive answer to the corresponding instance $\Lambda(G, 2, \mathcal{P}, H)$. Conversely if an instance of $\Lambda(G, 2, \mathcal{P}, H)$ is positive, then all vertices u_l, w_l (for all l) have color 1, so for each chain $P_i'' = \{x_d, u_l, y_e, w_l, z_f\}$ there is exactly one vertex in $\{x_d, y_e, z_f\}$ with color 1. Furthermore from the requirements on the chains P_i', for each variable x, all vertices $x_1, x_2, \ldots, x_s$ have the same color. So assigning the value 'true' to x if $x_1, x_2, \ldots, x_s$ have color 1 or value 'false' otherwise we get a positive answer to ONE-IN-THREE 3SAT. ∎

Theorem 3.4. *$\Lambda(G, 2, \mathcal{P}, H)$ is NP-complete when G is a tree with maximum degree* 3.

Proof. Again, we reduce from the NP-complete problem ONE-IN-THREE 3SAT with no negated literal, already defined. We denote by $x_1, \ldots, x_\nu$ the variables and by $c_1, \ldots, c_\alpha$ the clauses. We construct a tree as follows: there is a main path Π with $\nu + \alpha$ vertices. Each one of the ν first vertices of Π is linked by an edge to a leaf, the ith leaf being labeled by x_i (we shall speak of a *variable leaf*). Each one of the α next vertices of Π is linked to a *clause gadget* (so that, in our tree, there is one gadget for each clause): the gadget for a clause $c_h = x_i \vee x_j \vee x_k$ is a tree with five vertices (labeled a_h, b_h, x_i, x_j, and x_k), x_i, x_j, and x_k being the 3 leaves, and a_h being linked to Π by an edge. The edges inside the gadget are $[a_h, x_i], [a_h, b_h], [b_h, x_j]$, and $[b_h, x_k]$ (see Fig. 1 for an example). Note that the tree constructed so far has maximum degree 3.

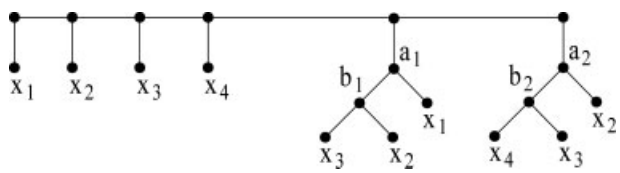

FIG. 1. The tree constructed for the instance $(x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3 \vee x_4)$.

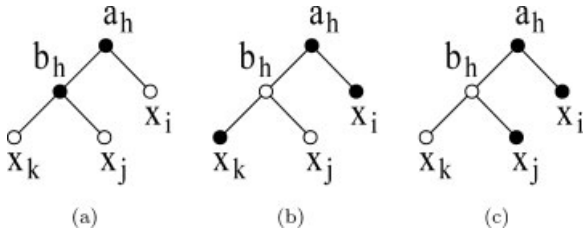

FIG. 2. The three possible colorings for the clause gadget of $c_h = x_i \vee x_j \vee x_k$.

It remains to describe the collection $\mathcal{P}$. First, in the gadget of clause $c_h = x_i \vee x_j \vee x_k$, there is a chain $P_h = \{a_h\}$ with $h(P_h) = (1, 0)$, a chain $P'_h = \{x_i, a_h, b_h\}$ with $h(P'_h) = (2, 1)$, and a chain $P''_h = \{x_j, b_h, x_k\}$ with $h(P''_h) = (1, 2)$. Then, the path Π is a chain in $\mathcal{P}$ with $h(\Pi) = (v+\alpha, 0)$. Eventually, for each occurrence of a variable x_i in a clause c_r there is a chain from the variable leaf i to the leaf x_i in the clause gadget of c_r. Let us denote by P_i^r this chain. If the leaf x_i in the clause gadget of c_r is linked to a_r then we have $h(P_i^r) = (|P_i^r| - 1, 1)$ else we have $h(P_i^r) = (|P_i^r| - 2, 2)$.

Now, the important point is that, because of all the chains of the form P_h, P'_h, and P''_h, there are only three ways of coloring each clause gadget (see Fig. 2: black vertices have color 1, white vertices have color 2).

Moreover, because of all the chains of the form P_i^r, given one of the 3 possible colorings of the clause gadget of $c_h = x_i \vee x_j \vee x_k$, one and only one of the variable leaves labeled x_i, x_j and x_k has color 1: x_i in the coloring of Figure 2a, x_j in the coloring of Figure 2b, x_k in the coloring of Figure 2c. Hence, given a solution for $\Lambda(G, 2, \mathcal{P}, H)$ on this instance, we can easily obtain a solution for the associated satisfiability instance, by assigning *true* to variables whose variable leaves have color 1 and *false* to the others. Conversely, given a truth assignment, assign color 1 to variable leaves associated with *true* variables and color 2 to the others, and color each clause gadget with respect to the *only* variable equal to *true* in the associated clause. It follows from the above discussion that we obtain a valid coloring. ∎

In the above construction, by contracting Π into a single vertex v, and all the a_h into v (i.e., $a_1 = \cdots = a_\alpha = v$) we obtain:

Theorem 3.5. *$\Lambda(G, 2, \mathcal{P}, H)$ is NP-complete in trees of diameter at most 4 when $|P_i| \leq 4$ for each P_i in $\mathcal{P}$, $h_i^j \leq 3$ $(i \leq p,\ j = 1, 2)$ and $|P_i \cap P_f| \leq 2$ for each P_i and P_f $(P_i \neq P_f)$ in $\mathcal{P}$ $(i, f \leq p)$.*

3.2. Polynomially Solvable Cases with $k = 2$

We recall that the basic image reconstruction problem in discrete tomography is polynomially solvable for $k = 2$ when the $P'_i s$ are the rows and the columns of the associated grid graph G. Remember that in this special case we have $c(P) = 2$.

More generally, we can state:

Theorem 3.6. *$\Lambda(G, 2, \mathcal{P}, H)$ is polynomially solvable if $c(\mathcal{P}) = 2$.*

Proof. We construct a multigraph G' as follows: Assign a vertex P_i to each chain P_i in $\mathcal{P}$. Each vertex of G, which is in P_i and in P_f is represented by an edge in G' between P_i and P_f. Each vertex, which is covered by a unique P_i is associated to an edge in G' between vertex P_i and a new vertex P'_i. So there is a one-to-one correspondence between the vertices of G and the edges of G'.

Then a solution, if there is one, will correspond to a subset F of edges of G' such that for each vertex P_i, F has h_i^1 edges adjacent to P_i (there is no restriction for the vertices P'_i).

In G', the edges of F will give V^1 in G and the edges not in F will correspond to V^2 in G. There are polynomial algorithms (see [13]) to construct such subsets F if they exist or to decide that there is no solution. ∎

One can derive the following from results in [9].

Theorem 3.7. *$\Lambda(G, 2, \mathcal{P}, H)$ is polynomially solvable if $Nest(\mathcal{P}) = 2$.*

Proof. Starting from the inclusion tree of each one of the two nested families covering $\mathcal{P}$, one can build a network flow model where a compatible integral flow will define the subset $V^1 \subseteq V$ and $V^2 = V - V^1$ will be obtained immediately as shown in [9].

Assume $\mathcal{P}$ can be decomposed into nested subfamilies A and B. We represent both families by the inclusion tree of their subsets P_i. A source a (resp. a sink b) is linked to all maximal (inclusionwise) subsets of A (resp. B). We link each $l \in V$ to the unique minimal subset A_r of A (resp. B_s of B) which contains l by an arc (A_r, l) (resp. (l, B_s)). The network is obtained by orienting all remaining edges from a to b. The arc entering (resp. leaving) each P_i in A (resp. B) has a capacity and a lower bound of flow equal to h_i^1. The arcs adjacent to the vertices corresponding to the elements of V have capacity 1 and a lower bound of flow equal to 0.

In Figure 3 an example is given for a set $V = \{1, 2, \ldots, 7\}$ and a family $\mathcal{P}$ with $Nest(\mathcal{P}) = 2$. Here

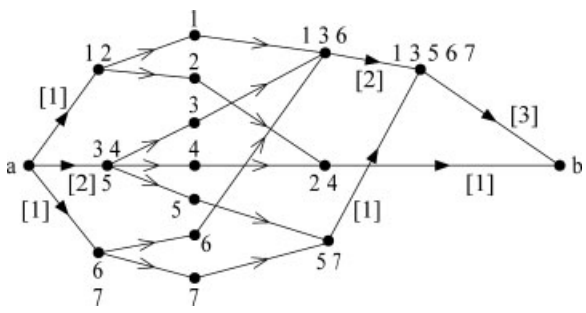

FIG. 3. The network associated with a family $\mathcal{P}$ with Nest$(\mathcal{P}) = 2$.

$A = (\{1,2\},\{3,4,5\},\{6,7\})$ and $B = (\{1,3,6\},\{2,4\},\{5,7\},\{1,3,5,6,7\})$. The values h_i^1 are shown in brackets.

There is a one-to-one correspondence between the feasible integral flows from a to b and the subset V^1 of vertices in a coloring (V^1, V^2) satisfying the requirements. ∎

Theorem 3.8. *Let G be an arbitrary graph and $\mathcal{P}$ a family of chains P_i such that any P_i has at most two vertices belonging to some other chains of $\mathcal{P}$. Then $\Lambda(G,2,\mathcal{P},H)$ can be solved in polynomial time.*

Proof. We shall transform the problem into a 2SAT problem which is known to be polynomially solvable [2].

We associate a binary variable x to every vertex of G which belongs to at least two chains P_i. Notice that we may assume that $\min\{h_i^1, h_i^2\} \geq 1$, $i \leq p$, otherwise there is only one color occurring in P_i and the problem can be reduced. We first remove all vertices which belong to exactly one P_i (these will be considered later). Now each P_i contains one or two vertices. For each P_i which has exactly two vertices, say x, y, which belong to other chains, we write a clause c_i as follows.

If $h_i^1 = 2, h_i^2 = 1$, we set $c_i = x \vee y$ (this means that at least one of the vertices x, y must have color one) and if $h_i^1 = 1, h_i^2 = 2$, we set $c_i = \bar{x} \vee \bar{y}$ (at least one of x, y must have color 2). If $\min\{h_i^1, h_i^2\} \geq 2$, we do nothing (since x and y can get any color). Finally, when $h_i^1 = h_i^2 = 1$, we introduce a constraint $x = \bar{y}$ (because x and y must get different colors). For any P_i which has exactly one vertex belonging to more than one chain in $\mathcal{P}$, we do nothing since by assumption $(\min\{h_i^1, h_i^2\} \geq 1)$ this vertex can have any color. We define $\mathcal{C} = \wedge_{i=1}^{q} c_i$ and using the equality constraints $x = \bar{y}$ we may substitute variable $\bar{y}$ to variable x. We are left with a 2SAT instance. It has a solution if and only if $\Lambda(G,2,\mathcal{P},H)$ has a solution.

From a solution of 2SAT, we derive a partition V^1, V^2 of the vertices associated to the binary variables. The bicoloring V^1, V^2 of the vertices of G belonging to more than one chain of $\mathcal{P}$ is given by $V^1 = \{v|\ v \text{ is true}\}$, $V^2 = \{v|\ v \text{ is false}\}$.

For each P_i it is possible to assign color 1 or 2 to the uncolored yet vertices so that the number of occurrences of color j is h_i^j (for $j = 1, 2$). This will provide the required coloring of G.

Conversely if $\Lambda(G,2,\mathcal{P},H)$ has a solution, then by setting $x =$ true (resp. $x =$ false) for all variables corresponding to the vertices x which are in more than one chain and have color 1 (resp. color 2), we will satisfy all clauses in $\mathcal{C}$ (as well as the equality constraints). ∎

Theorem 3.9. *If $G = (V, U)$ is a tree where all arcs have an orientation and each $P_i \in \mathcal{P}$ is an oriented path, then $\Lambda(G,2,\mathcal{P},H)$ can be solved in polynomial time.*

Proof. Notice that the incidence matrix (paths × vertices) of such a graph is totally unimodular. So if we write the system $A\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{1}$ where $a_{iv} = 1$ if path P_i contains vertex v (or $a_{iv} = 0$ else) and $b_i = h_i^1$, then we may check in polynomial time with a linear programming solver whether the system has a solution; if it is the case there is an integral solution (since A is totally unimodular) which gives V^1, and $V^2 = V - V^1$ which form a partition of V satisfying all requirements. ∎

3.3. The Case $k \geq 3$

Let us first consider the special case where all P_i's have size $|P_i| \leq 2$.

Theorem 3.10. *For any graph G and any $\mathcal{P}$ such that every $|P_i| \leq 2$, $\Lambda(G,k,\mathcal{P},H)$ can be solved in polynomial time.*

Proof. Consider $\Lambda(G,k,\mathcal{P},H)$. Eliminate all P_i's such that $h_i^j = 2$ for some color $j \leq k$ (these have a unique coloring) and apply the reductions implied by these eliminations. We also apply the reductions due to chains P_i with $|P_i| = 1$. Consider a pair P_i, P_j with $|P_i \cap P_j| = 1$. Let Π_l be the set of colors j with $h_l^j > 0$. If $\Pi_i \cap \Pi_j = \emptyset$, there is no solution; if $|\Pi_i \cap \Pi_j| = 1$, then assign this color to the vertex in $P_i \cap P_j$ and the rest of P_i, P_j is also determined. We apply these reductions until either we get a contradiction or we have a collection of connected components $C_1, \ldots, C_r$ where in each connected component all P_i's have the same set Π_i of possible colors (remember that $|P_i| = 2$ and $|\Pi_i| = 2$). Then our problem has a solution if and only if every connected component is bipartite. ∎

For the case where the number of colors is $k = 3$, we have the following:

Theorem 3.11. *$\Lambda(G,3,\mathcal{P},H)$ is NP-complete when $|P_i| = 3$, $h_i^j = 1$ for $i = 1, \ldots, p$, $j = 1,2,3$ and $c(\mathcal{P}) = 2$.*

Proof. We use a transformation from edge three-coloring of a three-regular graph G'. This problem is known to be NP-complete [10].

We will construct a graph G and a family $\mathcal{P}$ of chains in G. We will associate a chain P_i in G to each vertex w_i of G'; each edge $[w_i, w_f]$ of G' is associated with a vertex $v_{if} \equiv v_{fi}$ of $V(G)$. P_i will be a chain in G containing the three vertices corresponding to the three edges adjacent to w_i in G'. If in G' vertex w_i is adjacent to w_r, w_s, w_t $(r < s < t)$ then in G $P_i = \{v_{ir}, v_{is}, v_{it}\}$ and the corresponding chain will be formed by edges $[v_{ir}, v_{is}], [v_{is}, v_{it}]$.

We set $h_i^j = 1$ for $i = 1, \ldots, p$ and $j = 1,2,3$. Then there is an edge three-coloring of G' if there is a partition V^1, V^2, V^3 of $V(G)$ such that for each P_i, $|P_i \cap V^j| = 1 = h_i^j$ for any i, j. ∎

Theorem 3.11 is best possible since from Theorem 3.10 the problem is easy when $|P_i| \leq 2$ for all $i \leq p$.

Remark 3.1. *According to Brooks theorem (see* [3]*), the chromatic number $\chi(G)$ of a three-regular connected graph G is 3 unless G is either a clique on four nodes (in which case $\chi(G) = 4$) or a bipartite graph (in which case $\chi(G) = 2$).*

Since edge three-coloring is NP-complete in three-regular graphs [10]*, we can state: edge 3-coloring in a three-regular graph G is NP-complete even if* $\chi(G) = 3$.

Conversely, note that if a connected graph G is edge three-colorable then $\Delta(G) \leq 3$ *and thus either G is a clique on four nodes or* $\chi(G) \leq 3$.

3.4. The Case Where G is a Chain or a Tree and $k > 2$

We will now consider $\Lambda(G, k, \mathcal{P}, H)$ where G is a tree, each P_i is a chain of G and furthermore for any two chains P_i, P_f in $\mathcal{P}$ we have $|P_i \cap P_f| \leq 1$. In such a case we have the following:

Lemma 3.1. *If G is a tree and if the family* $\mathcal{P}$ *of chains of G satisfies* $|P_i \cap P_f| \leq 1$ *for all* $i,f \leq p$*, then there is an order (which we call canonical order) of chains such that for any* $q > 1$

$$\left| P_q \bigcap \left(\bigcup_{i=1}^{q-1} P_i \right) \right| \leq 1$$

Proof. Notice first that we can assume $|P_i| \geq 2$ for each $i \leq p$. This implies that we cannot have $P_i \subset P_f$ for any $i,f \leq p$ $(i \neq f)$. Now G has a pendent vertex contained in exactly one chain P_i of $\mathcal{P}$. This chain will be called P_1; we remove it from $\mathcal{P}$ as well as all vertices belonging to P_1 only. Now we can find another pendent vertex of the remaining tree G' and this determines P_2. We will thus find a numbering of the chains of $\mathcal{P}$ which satisfies the requirements. ∎

We will describe below an algorithm for solving $\Lambda(G, k, \mathcal{P}, H)$ in a tree $G = T$; in this procedure (called FFC) we will have to determine for each vertex of T the "forced" colors as well as the "forbidden" colors; such a procedure will also be able to detect contradictions in the data which imply that no solution exists. A color c is said to be *forced* (resp. *forbidden*) for a vertex v if there exists no feasible solution where v has a color $c' \neq c$ (resp. where v has color c).

The procedure FFC which makes a repeated use of a maximum flow in a bipartite graph can be sketched as follows.

Procedure FFC (Forced and Forbidden Colors). Let us consider a chain P_i and let us denote by $x_1, \ldots, x_v$ the vertices in P_i. Let Π_i be the set of colors required in P_i : $\Pi_i = \{j \mid h_i^j > 0\}$. For each vertex x_l, $l = 1, \ldots, v$, π_l denotes the set of possible colors for x_l, i.e. $\pi_l = \cap_{i|x_l \in P_i} \Pi_i$.

We construct the following bipartite graph $G = (X, Y, E)$ with $X = \{x_1, \ldots, x_v\}$, $Y = \Pi_i$, and $[x_l, j] \in E$ if $j \in \pi_l$; the capacity of $[x_l, j]$ is equal to 1. To get a network N, we add a source s with an arc of capacity 1 from s to each vertex in X and a sink t with an arc from each vertex j in Y to t; the capacity of (j, t) is equal to h_i^j for all $j \in Y$. Any integral flow from s to t saturating the arcs out of s gives a possible coloring of the vertices in P_i. To any edge $[x_l, j] \in E$ which is saturated in every maximum flow corresponds a forced color j for x_l. To any edge $[x_l, j] \in E$ with a flow equal to 0 in every maximum flow corresponds a color j forbidden for x_l.

Note that it is easy to determine all the edges saturated (resp. with no flow) in every maximum flow. For each edge $[x_l, j]$ in E, suppress $[x_l, j]$ (resp. force a flow from s to t through $[x_l, j]$) and compute a new maximum flow in the obtained network. If the value of this flow is lower than the original maximum flow, then $[x_l, j]$ is saturated (resp. with no flow) in every maximum flow.

Procedure FFC either finds the forbidden colors or a forced color for a vertex v or concludes that there is no more forbidden color nor forced color. If the set π_v of possible colors for v is $\pi_v = \{1, \ldots, k\}$ initially for each vertex v, we notice that finding a forced color c for v reduces π_v to a set $\pi_v = \{c\}$ and finding the forbidden colors $c_{i_1}, \ldots, c_{i_q}$ for v replaces π_v by $\pi_v = \pi_v - \{c_{i_1}, \ldots, c_{i_q}\}$.

Since we will apply FFC as long as forbidden or forced colors can be found, it will be called at most $|V|k$ times.

Clearly we remove all vertices which have a forced color and we update the values h_i^j accordingly as well as the sets Π_i.

At the end of the repeated applications of FFC we will either have discovered a contradiction ($\pi_v = \emptyset$ for some vertex v) or obtained for each vertex v a set π_v with $| \pi_v | \geq 2$.

Theorem 3.12. *If G is a tree and* $\mathcal{P}$ *a family of chains of G satisfying* $|P_i \cap P_f| \leq 1$ *for any* $i,f \leq p$ $(i \neq f)$*, then* $\Lambda(G, k, \mathcal{P}, H)$ *can be solved in polynomial time.*

Proof. We start by applying the FFC procedure; it may happen that one has to remove some vertices with forced colors; in such a case we get a forest and we apply the procedure on each connected component separately.

Wlog we consider a tree G and we construct a canonical order $P_1, \ldots, P_p$ of the chains of $\mathcal{P}$. Since we apply procedure FFC until there are no more forced colors and neither forbidden colors, we have the following:

Fact 3.1. *If in a chain* P_i *a single arbitrary vertex v has been given a possible color* $c \in \pi_v$*, there exists an assignment of possible colors* $c(w) \in \pi_w$ *to all remaining vertices w of* P_i *such that* P_i *has exactly* h_i^j *vertices of color j* $(1 \leq j \leq k)$.

It is then possible to color the vertices of G by considering the chains $P_1, \ldots, P_p$ in the canonical order (starting from any vertex of P_1). Clearly we will be able to extend the coloring to all vertices of G since, having colored the vertices of $P_1, \ldots, P_i$, the chain P_{i+1} has exactly one vertex which is already colored (with a color in π_v).

The whole procedure is polynomial:

FFC consists of applying for each chain P_i a maximum flow algorithm in a bipartite network with $|P_i|$ vertices on the left and k vertices on the right. To find the forbidden colors and the forced colors, we have to find at most $|P_i|k$ times an augmenting chain (this takes $O(|P_i|k)$ time); globally we have a complexity $O((|P_i|k)^2)$ for getting the forbidden colors and the forced colors. For a maximum flow we have $O((|P_i| + k)^3)$ (see [1]). Hence an application of FFC has a complexity $O((|P_i| + k)^3 + (|P_i|k)^2)$. Since we apply FFC at most $|V|k$

times, we have $O(((|P_i| + k)^3 + (|P_i|k)^2)|V|k)$ and since $|P_i| \leq |V|$ we have finally $O(((|V|+k)^3+(|V|k)^2)|V|k)$. ∎

Proposition 3.1. *If G is a cycle and if the family $\mathcal{P}$ is such that $|P_i \cap P_f| \leq 1$ for any $i,f \leq p$ with $i \neq f$, then $\Lambda(G,k,\mathcal{P},H)$ can be solved in polynomial time.*

Proof. We take a consecutive numbering of the chains P_i as in the case where G is a tree so that $|P_i \cap P_{i+1}| = 1$ for all $i \leq p-1$ and in addition $|P_p \cap P_1| = 1$; let $v_0 \in P_p \cap P_1$.

We simply consider the following problems O_j (for $j = 1,\ldots,k$): find a feasible coloring such that v_0 has color j.

This amounts to removing v_0 and updating the h_i^j accordingly; this is simply $\Lambda(G - v_0, k, \mathcal{P}', H')$ where $G - v_0$ is a chain. ∎

More generally if G is a *cactus*, i.e., a connected graph where any two cycles have at most one common vertex, then we can proceed as for a tree in the following special case: let us assume that each P_i belongs to exactly one cycle (or to a chain not contained in a cycle). Each cycle C has some vertices which may belong to other cycles or to external chains; we shall assume that all these vertices are necessarily endpoints of chains P_i.

It is not difficult to see that we can number the P_i's in $\mathcal{P}$ in such a way that for all $f \leq p$ $|P_f \cap (\cup_{i=1}^{f-1} P_i)| \leq 1$ (except for the last P_i's which "close" a cycle in G).

We can work separatly on each cycle C and determine the possible colors for the last vertex, i.e. the vertex connecting C to some cycle or some external chain covered by chains P_i with smaller indices.

We proceed as in the case of trees by applying an FFC procedure first and then, in case no contradiction has occurred, we will be in the situation where we have either a single P_i (contained in an external chain) to color where exactly one vertex is already colored or we reach a cycle (with exactly one vertex already colored). In the first case we proceed as before and in the second one, we have that the cycle can be colored by extending the coloring from the vertex which has been colored and we continue. This will finally color the whole graph.

As in the case of trees, the procedure will give a feasible coloring or exhibit a contradiction.

4. PROPER COLORINGS

Having discussed $\Lambda(G,k,\mathcal{P},H)$ we shall examine the case where the k-partition is a proper k-coloring ($\Lambda^*(G,k,\mathcal{P},H)$).

Here we shall assume that for every edge $e = [x,y]$ in G, there is a chain μ_i which uses e; this implies in particular $x,y \in P_i$. This assumption is not restrictive: let $e = [x,y]$ be an edge which is not covered by any μ_i in the collection defined in $\Lambda^*(G,k,\mathcal{P},H)$. We replace e by a chain $\mu_e = (x_e^1 = x, u_e^1, x_e^2, u_e^2, \ldots, x_e^{k-1}, u_e^{k-1}, x_e^k = y)$ where $x_e^2,\ldots,x_e^{k-1}$ are new vertices and $u_e^1,\ldots,u_e^{k-1}$ are new edges; we set $P_e = \{x_e^1,\ldots,x_e^k\}$ and $h_e^j = 1$ for $j = 1,\ldots,k$.

Clearly there is a proper k-coloring of the resulting graph G^* which is solution of $\Lambda^*(G^*,k,\mathcal{P},H)$ if there is a proper k-coloring which is solution of $\Lambda^*(G,k,\mathcal{P},H)$ because x and y will necessarily get different colors in any feasible coloring of G^*.

4.1. Solvable Cases of Proper Colorings

Let us now consider some cases for which polynomial time algorithms can be found.

Fact 4.1. *$\Lambda^*(G,2,\mathcal{P},H)$ is polynomially solvable.*

Justification. Notice that in each P_i with odd $|P_i|$, the vertices have necessarily forced colors. So we can assume that there are only chains of even length and each vertex may be colored with color 1 or 2. The problem then consists in verifying whether the graph is bipartite or not, which can be done in polynomial time.

We obtain from Theorem 3.12 and its proof:

Corollary 4.1. *$\Lambda^*(G,k,\mathcal{P},H)$ can be solved in polynomial time if G is a tree, $\mathcal{P}$ is such that $|P_i \cap P_f| \leq 1$ for $i,f \leq p$ ($i \neq f$) and $h_i^j \leq 1$ for all $i \leq p$, $j \leq k$.*

From now on we will have to consider repeatedly proper k-colorings of chains P_i of G (with possibly $k > 2$ and with h_i^j occurrences of color j in chain P_i). So we will start by stating some elementary properties of such colorings.

We recall that a color j is saturating in a chain P if $h^j = \lceil \frac{|P|}{2} \rceil$. The set of colors j such that $h^j > 0$ will be denoted by Π.

Remark 4.1. *If P is an odd chain with a saturating color a, then a occurs necessarily at both endpoints of P in any coloring.*

Remark 4.2. *If P is an even chain with a saturating color a, then a occurs necessarily at least at one endpoint of P in any coloring.*

Lemma 4.1. *Let P be a chain to be colored and assume there is no saturating color in Π. For any two colors e,d in Π, one can find a proper k-coloring of P where e and d occur at the endpoints of P. In case $h^d \geq 2$, we can have a coloring with d occurring at both endpoints.*

Proof. Let $P = \{1,2,\ldots,n\}$ and let d,e be the colors which have to occur at the ends. Assume first that n is even. Start from the left, assigning h^d times color d to vertices $1,3\ldots,2h^d-1$ and from the right, assign h^e times color e to vertices $n, n-2, \ldots, n-2(h^e-1)$. It remains $\max\{0, n-2h^e-2h^d+2\}$ adjacent vertices in the center. We can find $\max\{0, \frac{n}{2} - h^e - h^d + 1\}$ nonadjacent vertices among them. Together with the vertices $2,4,\ldots,2h^d-2$ and $n-1, n-3, n-2h^e+3$, this gives $\frac{n}{2}-1$ nonadjacent vertices.

If n is odd, we choose a color $f \neq d, e$ (which exists since there is no saturating color). We color vertex n with e and we decrease h^e by one. Then we apply the previous coloring, with color f replacing color e, to $P' = P - \{n\}$; this will give a proper coloring of P since vertex $n-1$ has color f and vertex n has color e.

Finally we start by coloring the nonadjacent vertices with the remaining colors. If $h^e + h^d - 1 \geq \frac{n}{2}$, then all uncolored vertices are nonadjacent and the coloring can be completed. In the other case ($h^e + h^d - 1 < \frac{n}{2}$), we have an interval I of $n - 2h^e - 2h^d + 2$ consecutive uncolored vertices in the center. We color the remaining vertices in the order $2h^d, 2h^d + 2, \ldots, n-2h^e, n-2h^e+3, \ldots, n-1, 2, 4, \ldots, 2h^d-2, 2h^d+1, 2h^d+3, \ldots, n-2h^e+1$ exhausting one color before taking the next one. Since there is no saturating color we will get a proper coloring of the chain.

To obtain a coloring with d occurring on 1 and n, consider $P' = P - \{n\}$ and $(h^d)' = h^d - 1$. Apply the coloring algorithm to P' with colors d, e. Clearly vertex $n-1$ will not have color d and we can color vertex n with d to get the required proper coloring of P. ∎

Lemma 4.2. *If P is an even chain with exactly one saturating color a, one can choose any color b and construct a coloring of P such that a and b are occurring at the endpoints.*

Proof. Assume first $b \neq a$. Color the nodes $1, 3, 5, \ldots, |P| - 1$ with color a and color the nodes $|P|, |P| - 2, \ldots, 2$ with the remaining colors starting with color b.

If $b = a$, then color a occurs at both ends: we color nodes $1, 3, 5, \ldots, |P| - 3$ and $|P|$ with color a. Since there are no other saturating color, we can color the nodes $|P| - 2, |P| - 4, \ldots, 2, |P| - 1$ with the remaining colors and no conflict will occur. ∎

We shall say that the singletons P_i in $\mathcal{P}$ have the *CS* property (Consecutive Singletons) if the following holds: if a singleton P_i is an intermediate vertex of some $P_c = \{x_c^1, x_c^2, \ldots, x_c^r = P_i, x_c^{r+1}, \ldots, x_c^s\}$ then either $x_c^1, \ldots, x_c^{r-1}$ or $x_c^{r+1}, \ldots, x_c^s$ are also singletons in $\mathcal{P}$.

Proposition 4.1. *Let G be a chain. For any $\Lambda^*(G, k, \mathcal{P}, H)$ with $|P_i \cap P_f| \leq 1 \ \forall i \neq f (i, f \leq p)$ and where all singletons P_i have the CS property, there is an equivalent problem $\Lambda^*(G^*, k, \mathcal{P}^*, H^*)$ where the family $\mathcal{P}^*$ satisfies:*

(a) $|P_i^*| \geq 2$
(b) $|P_i^* \cap P_{i+1}^*| \leq 1$ $(i < p^*)$ *and* $P_i^* \cap P_f^* = \emptyset$ $(i \notin \{f - 1, f, f+1\})$

Here "equivalent" means that one problem has a solution if and only if the other one has a solution.

Proof. Assuming that the vertices are given in increasing order of numbering along the chain, we can say that a chain P_i starts at some vertex x_d (or ends at some vertex x_e) if d is the smallest (e is the largest) index in P_i.

Now consider a chain $P_c = \{x_c^1, \ldots, x_c^r, x_c^{r+1}, \ldots, x_c^s\}$ where $x_c^1, \ldots, x_c^r$ are singletons $P_c^1, \ldots, P_c^r$ in $\mathcal{P}$. We remove $x_c^1, \ldots, x_c^r$ and replace P_c by $P_c^* = \{x_c^{r+1}, y^1, \ldots, y^{k-2}\}$ with $h(P_c^*) = (1, \ldots, 1, 0, 1, \ldots, 1)$ where the missing color is the color of x_c^r and $y^1, \ldots, y^{k-2}$ are new vertices. We also introduce $P_c^{**} = \{x_c^{r+1}, \ldots, x_c^s\}$ with updated values of h_i^j according to the colors already assigned to $x_c^1, \ldots, x_c^r$. Similarly, if there is a chain $P_d = \{x_d^1, \ldots, x_d^t = x_c^1\}$ ending at vertex x_c^1 we replace it by a chain $P_d^* = P_d - x_c^1 = \{x_d^1, \ldots, x_d^{t-1}\}$ and introduce $P_d' = \{x_d^{t-1}, z^1, \ldots, z^{k-2}\}$ with $h(P_d' = (1, \ldots, 1, 0, 1, \ldots, 1)$ where the missing color is the color of x_c^1. We update the values h_i^j accordingly. Then we have an equivalent problem since x_d^t will not get the color of x_c^1 and x_c^{r+1} will not get the color of x_c^r. So we have cut the problem into two subchains and singletons in P_c have been removed. By repeating this we get an equivalent problem with all P_i's verifying $|P_i| \geq 2$. ∎

Theorem 4.1. *If G is a chain where the singletons P_i have the CS property and $\mathcal{P}$ is such that $|P_i \cap P_f| \leq 1 \ \forall i \neq f$ $(i, f \leq p)$, then $\Lambda^*(G, k, \mathcal{P}, H)$ can be solved in polynomial time.*

Proof. As already remarked, we can assume that $k \geq 3$.

Wlog we can assume that $\mathcal{P}$ has the properties (a) and (b) given in Proposition 4.1. Consider now the problem $\Lambda^*(G, k, P, H)$. To solve it we use a procedure similar to the one used for $\Lambda(G, k, P, H)$. If any contradiction occurs during the following forced assignments then there is no solution.

- Whenever a vertex $v \in P_i \cap P_{i+1}$ is assigned some color j we update the parameters as follows: $h_i^j \leftarrow h_i^j - 1$; $h_{i+1}^j \leftarrow h_{i+1}^j - 1$; if $h_i^j = 0$ then set $\Pi_i \leftarrow \Pi_i - \{j\}$; if $h_i^{j+1} = 0$ then set $\Pi_{i+1} \leftarrow \Pi_{i+1} - \{j\}$.
- If there exists $1 \leq i < p$ such that $\Pi_i \cap \Pi_{i+1} = \emptyset$, then there is no solution.
- If there exists $1 \leq i < p$ such that $|\Pi_i \cap \Pi_{i+1}| = 1$ then color $P_i \cap P_{i+1}$ with the common color.
- For each odd P_i with a saturating color, say j, assign color j to both endpoints of P_i.
- For each even P_i with a saturating color, say j, j must be assigned to one of the endpoints of P_i. For $1 < i < p$,
  if $j \notin \Pi_{i-1} \cup \Pi_{i+1}$ then there is no solution
  if $j \notin \Pi_{i-1}$ then assign color j to $P_i \cap P_{i+1}$
  if $j \notin \Pi_{i+1}$ then assign color j to $P_i \cap P_{i-1}$.

For any colored vertex, propagate the possible implications of this coloring to the previous and next intersections in the following way; if any contradiction occurs, there is no solution.

Assume that $v \in P_i \cap P_{i+1}$ has been colored with j:

- if $l \neq j$ is a saturating color of P_i (resp. P_{i+1}) then color the left (resp. right) endpoint of P_i (resp. P_{i+1}) with l,
- if $|\Pi_i \cap \Pi_{i-1}| = 1$ $(i > 1)$ (resp. $|\Pi_{i+1} \cap \Pi_{i+2}| = 1$ $(i < p-2)$) assign the unique color l such that $h_i^l \geq 1$ and $h_{i-1}^l \geq 1$ (resp. $h_{i+1}^l \geq 1$ and $h_{i+2}^l \geq 1$) to $P_i \cap P_{i-1}$ (resp. $P_{i+1} \cap P_{i+2}$).

At this step, if no contradiction occurred, we have a set of colored vertices located at intersections of chains P_i. In addition, any pair $\{P_i, P_{i+1}\}$ $(i < p)$ such that $P_i \cap P_{i+1}$ is uncolored verifies $|\Pi_i \cap \Pi_{i+1}| \geq 2$ and if j is a color saturating P_i then $j \in \Pi_{i-1} \cap \Pi_i \cap \Pi_{i+1}$ $1 < i < p$.

Moreover, if one endpoint of P_i $(1 \leq i \leq p)$ is already colored, any color remaining in Π_i is compatible with it and can be used to color the other endpoint; if P_i has a saturating color it is the one already assigned.

The problem has a solution which can be obtained in two more steps:

(A) First we assign a color to all uncolored intersection $P_i \cap P_{i+1}(i < p)$, in the following way: Let $P_i \cap P_{i+1}$ be the first uncolored intersection in G; color $P_i \cap P_{i+1}$ with any color $j \in \Pi_i \cap \Pi_{i+1}$. If $i = 1$, color the first endpoint of P_i with any allowed color. If $P_{i+1} \cap P_{i+2}$ is uncolored $(i+1 < p)$ then there is at least one color different from j in $\Pi_{i+1} \cap \Pi_{i+2}$; if there is a saturating color l in P_{i+1} and if $l \neq j$ then assign color l to $P_{i+1} \cap P_{i+2}$ (we are sure that $l \in \Pi_{i+1} \cap \Pi_{i+2}$) otherwise choose any color in $\Pi_{i+1} \cap \Pi_{i+2}$. Propagate the implications of each coloring until we reach a vertex already colored. Then search for the following uncolored intersection and continue the process until the end of G.

(B) Clearly the partial coloring obtained so far is such that for every chain the saturating colors are assigned to endpoints in such a way that according to Lemmas 4.1 and 4.2, the coloring can be extended to all yet uncolored vertices.

∎

Remark 4.3. *One should mention that Theorem 4.1 can be extended to trees where $\mathcal{P}$ is such that in every P_i only the "first" and "last" vertices may belong to another P_j.*

Remark 4.4. *$\Lambda^*(G, k, \mathcal{P}, H)$ can be solved in polynomial time if $|P_i| = 2$ for all $i = 1, \ldots, p$. Since $|P_i| = 2$ for each i, each edge is a P_i and there are exactly two possible colorings for each P_i. We take the first coloring of P_1; we propagate this coloring and if we obtain a proper coloring of G, we are done. Else we have a conflict; we then reverse the coloring of P_1 and propagate this coloring as before and we will find a coloring of G or a conflict. In the last case, there is no solution.*

4.2. Difficult Cases of Proper Colorings

Theorem 4.2. *$\Lambda^*(G, 3, \mathcal{P}, H)$ is NP-complete in trees with maximum degree 3.*

Proof. We use the construction in the proof of Theorem 3.4 and introduce a new vertex on each edge of the tree; we force these new vertices to have color 3. ∎

Theorem 4.3. *$\Lambda^*(G, 3, \mathcal{P}, H)$ is NP-complete even if G is planar bipartite, $|P_i \cap P_f| \leq 1$ $(i,f \leq p,\ i \neq f)$, $|P_i| \leq 3$ $(i \leq p)$ and $h_i^j \leq 1$, $i = 1, \ldots, p$, $j = 1, 2, 3$.*

Proof. We use a transformation from the NP-complete problem PrExt which is defined as follows:

INSTANCE: A positive integer q and a graph G in which some vertices are precolored using at most q colors.

QUESTION: Can the precoloring of G be extended to a proper coloring of G using at most q colors?

This problem is proven to be NP-complete even if $q = 3$ and G is planar bipartite (see [11]).

Consider a planar bipartite graph $G = (X, Y, E)$. Suppose that some of its vertices are precolored using colors 1, 2, and 3. For each precolored vertex x, we set $P_x = \{x\}$ and $h(P_x) = (1, 0, 0)$ if x has color 1, $h(P_x) = (0, 1, 0)$ if x has color 2, and $h(P_x) = (0, 0, 1)$ if x has color 3. For each edge $e = [x, y]$ in G, we add a new vertex z_e and a new edge $[x, z_e]$. We set $P_e = \{x, y, z_e\}$ and $h(P_e) = (1, 1, 1)$.

Clearly our new graph G' is still planar bipartite. Furthermore $|P_i \cap P_f| \leq 1$ $(i,f \leq p,\ i \neq f)$, $|P_i| \leq 3$ $(i \leq p)$, and $h_i^j \leq 1$, $i = 1, \ldots, p$, $j = 1, 2, 3$.

It is easy to see that PrExt has a solution in G if and only if $\Lambda^*(G', 3, \mathcal{P}, H)$ has a solution in G'. ∎

5. EDGE COLORINGS

We now consider edge colorings instead of vertex colorings; we may in a similar way define problem $\Psi(G, k, \mathcal{P}, H)$ where $\mathcal{P}$ is a collection of p subsets P_i of edges of G and we want to find a k-partition $E^1, E^2, \ldots, E^k$ of E such that

$$\left|P_i \bigcap E^j\right| = h_i^j \quad \text{for all } i \leq p \text{ and all } j \leq k. \tag{2}$$

If we want to find a proper edge k-coloring then the problem will be denoted by $\Psi^*(G, k, \mathcal{P}, H)$.

In general, the subsets P_i of edges will be chains (open or closed). $|P_i|$ will be the number of edges in chain P_i.

Clearly problems Ψ and Ψ^* in a graph G are equivalent to problems Λ and Λ^* in $L(G)$ where $L(G)$ is the line graph of G (edges of G become vertices of $L(G)$).

It follows that when G itself is a chain, then $L(G)$ is also a chain and the results for Λ and Λ^* also apply to the edge coloring case.

5.1. Arbitrary Colorings

In this situation every edge e which is not included in some P_i may clearly be removed from G. So we can assume wlog that every e is in some P_i of $\mathcal{P}$.

Theorem 5.1. *$\Psi(G, k, \mathcal{P}, H)$ can be solved in polynomial time if $|P_i| \leq 2$ for each chain $P_i \in \mathcal{P}$.*

Proof. This follows directly from the proof of Theorem 3.10. After reduction we transform the graph as follows: each edge becomes a vertex and we link two vertices if there is a P_i containing the corresponding edges. The problem has a solution if and only if there is no odd cycle in this graph. ∎

Theorem 5.2. *$\Psi(G, 2, \mathcal{P}, H)$ is NP-complete even if G is a tree T with maximum degree 3 and the P_i's are chains or bundles.*

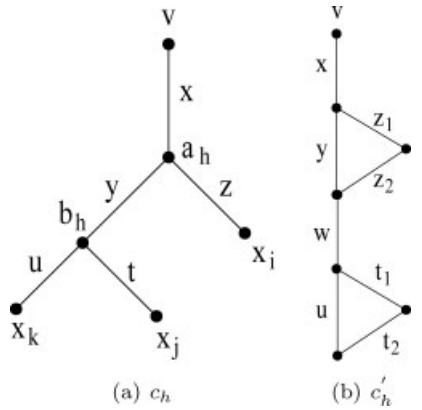

FIG. 4. Transformation of bundle constraints for a tree into chain constraints in a cactus.

Proof. We use the same reduction from ONE-IN-THREE 3SAT as in Theorem 3.4.

We have to color edges instead of vertices; the leaf variables now correspond to leaf edges and for each clause c_h we now have for the sets P'_h bundles of edges y, z and x (see Fig. 4a) if the clause is given by $c_h = x_i \vee x_j \vee x_k$. We set $h(P'_h) = (2,1)$. The set P''_h is now the bundle y, u, t with $h(P''_h) = (1,2)$ and the set $P_h = \{x\}$ with $h(P_h) = (1,0)$. The other chains P_i^r are defined similarly. ∎

If we require all P_i's to be exclusively chains in G (but not bundles) we can derive the following for a special cactus in which no two cycles have a common vertex [see *Le cactus* (G. Bitton and M. Munz, Private communication, Paris, December 14, 2006) for additional properties of cacti].

Theorem 5.3. *$\Psi(G, 2, \mathcal{P}, H)$ is NP-complete even if G is a triangulated cactus with maximum degree 3 and where all the P_i's are chains.*

Proof. We just have to show how the bundle requirements can be transformed into constraints related to chains.

We transform the clause gadget c_h as shown in Figure 4b. The cactus obtained in this way is triangulated (its cycles are triangles).

The bundle $P' = \{x, y, z\}$ with $h(P') = (2,1)$ becomes chains $P'_* = \{x, y, z_2, z_1\}$, $P'_{**} = \{z_2\}$ with $h(P'_*) = (2,2)$, $h(P'_{**}) = (0,1)$.

The bundle $P'' = \{y, u, t\}$ with $h(P'') = (1,2)$ becomes chains $P''_* = \{y, w, u, t_2, t_1\}$, $P''_{**} = \{w, t_1\}$ with $h(P''_*) = (1,4)$ and $h(P''_{**}) = (0,2)$.

Finally, the P_i's using chains between v and edges u and t can also be replaced by chains in the new gadget c'_h with appropriate modifications of the values h_i^j. ∎

5.2. Proper Colorings

Theorem 5.4. *$\Psi^*(G, 3, \mathcal{P}, H)$ is NP-complete when G is 3-regular, $\mathcal{P}$ is a collection of vertex disjoint triangles P_i considered as sets of edges (i.e. $|P_i| = 3, \forall i = 1, \ldots, p$, $P_i \cap P_f = \emptyset$ for all $i, f, i \neq f$) and $h_i^j = 1, \forall i = 1, \ldots, p, \forall j = 1, 2, 3$.*

Proof. We use a transformation from edge three-coloring of a three-regular graph G'. This problem is known to be NP-complete [10].

For each vertex i adjacent to vertices f, l, p, we introduce in G the vertices v_{if}, v_{il}, v_{ip}. These three vertices are pairwise linked forming a triangle which will correspond to P_i. Thus G will have $3|V|$ vertices. For each edge $[i, f]$ in G' we introduce an edge $[v_{if}, v_{fi}]$ in G.

We take $p = |V(G')|$ and $\mathcal{P} = (P_1, \ldots, P_p)$ with $h_i^j = 1$ for $i = 1, \ldots, p$ and $j = 1, 2, 3$. Notice that the P_i's form closed chains.

There is an edge three-coloring of G' if and only if there is an edge three-coloring of G. The edges of $E(G)$ are colored as follows:

1. for each edge $[i, f]$ of G' with color k, the corresponding edge $[v_{if}, v_{fi}]$ in G has color k;
2. the three edges forming a triangle P_i can be colored with three colors by extending the coloring obtained after the previous stage.

Finally, note that any edge three-coloring of G will satisfy the requirements on the sets P_i. ∎

Theorem 5.5. *$\Psi^*(G, 3, \mathcal{P}, H)$ is NP-complete when G is a bipartite three-regular graph and $\mathcal{P}$ is a family of chains P_i of length two which are pairwise nonadjacent.*

Proof. Let us call SIM (for simultaneity requirements) the following problem: we are given a three-regular bipartite simple graph G with two subsets $\mathcal{S}_1$, $\mathcal{S}_2$ of edges such that $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$ and the edges of $\mathcal{S}_i$ are pairwise nonadjacent for $i = 1, 2$.

Does there exist an edge three-coloring (M_1, M_2, M_3) of G such that $M_1 \supseteq \mathcal{S}_1$, $M_2 \supseteq \mathcal{S}_2$?

SIM was shown to be NP-complete in [7]. We use a reduction from SIM as follows: from $G = (V, E)$ with subsets $\mathcal{S}_1, \mathcal{S}_2$ we construct a simple graph G^* by replacing each edge $e = [x, y]$ in $\mathcal{S}_i$ by the graph given in Figure 5. We set $P_e = \{[x''_e, y'_e], [y'_e, x'''_e]\}$ with $h_e^1 = 0, h_e^2 = h_e^3 = 1$ for $i = 1$ or with $h_e^1 = 1, h_e^2 = 0, h_e^3 = 1$ for $i = 2$. Note that in any solution of $\Psi^*(G^*, 3, \mathcal{P}, H)$ the edge $[x'_e, y]$ will get the same color as $[x, y'_e]$.

G^* is a three-regular bipartite simple graph; it has an edge three-coloring satisfying the requirements on each P_e if and

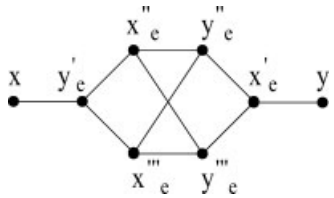

FIG. 5. Transformation of G where edge $e = [x, y]$ is precolored into G^*.

TABLE 1. Summary of the results for $\Lambda(G,k,\mathcal{P},H)$.

G	k	$\lvert P_i\rvert$	h_i^j	$\lvert P_i \cap P_f\rvert$		Status	Theorem
	2				$c(\mathcal{P}) = 2$	P	3.6
	2				Nest$(\mathcal{P}) = 2$	P	3.7
	2				$\lvert P_i \cap \bigcup P_f\rvert \le 2$ $i \neq f$	P	3.8
Dir. tree	2				P_i: oriented path	P	3.9
		≤ 2				P	3.10
Tree				≤ 1		P	3.12
Cactus				≤ 1		P	Prop. 3.1
	2	3			$\mathcal{P}$ 3-regular	NPC	3.1
	2				Nest$(\mathcal{P}) = 3$ $= c(\mathcal{P})$	NPC	3.2
Bipartite	2		≤ 3	≤ 1	$\Delta(G) \le 4$	NPC	3.3
Tree	2				$\Delta(G) \le 3$	NPC	3.4
Tree	2	≤ 4	≤ 3	≤ 2	Diameter ≤ 4	NPC	3.5
	3	3	1		$c(\mathcal{P}) = 2$	NPC	3.11

only if G has an edge three-coloring where each edge e in $\mathcal{S}_i$ has color i for $i = 1, 2$. ∎

Theorem 5.6. *$\Psi^*(G,3,\mathcal{P},H)$ is NP-complete when G is a planar bipartite graph with maximum degree $\Delta(G) \le 3$ and $\mathcal{P}$ is a family of chains P_i of length 2.*

Proof. We shall use a transformation from the precoloring extension problem on edges which is shown to be NP-complete even for planar, three-regular bipartite graphs [14].

Let $G' = (X \cup Y, E)$ be a planar three-regular bipartite graph in which some edges are precolored using colors 1, 2, and 3. For each vertex $i \in X \cup Y$ incident to two precolored edges, color the third edge with the remaining color (if there is a contradiction, the problem has no solution). For each vertex $i \in X \cup Y$ incident to one precolored edge $[i,f]$, take $P_i = \{[i,l],[i,p]\}$ where l, p are the endpoints of the two uncolored edges incident to i. If $[i,f]$ has color $j \in \{1,2,3\}$, take $h_i^j = 0, h_i^q = 1$, $q \neq j$, $q \in \{1,2,3\}$. Delete the precolored edges. We get a planar bipartite graph G with maximum degree $\Delta(G) \le 3$ and $\mathcal{P}$ is a family of chains P_i of length 2.

It is clear that the precoloring extension problem on the edges of G' has a positive answer if and only if $\Psi^*(G,3,\mathcal{P},H)$ has a positive answer. As G can be obtained from G' in polynomial time, we proved that our problem is NP-complete. ∎

TABLE 2. Summary of the results for $\Lambda^*(G,k,\mathcal{P},H)$.

G	k	$\lvert P_i\rvert$	h_i^j	$\lvert P_i \cap P_f\rvert$		Status	Theorem
	2					P	Fact 4.1
		2		≤ 1		P	Rem. 4.4
Tree			≤ 1	≤ 1		P	Cor. 4.1
Chain				≤ 1	CS property	P	4.1
Tree	3				$\Delta(G) \le 3$	NPC	4.2
Bipartite planar	3	≤ 3	≤ 1	≤ 1		NPC	4.3

TABLE 3. Summary of the results for $\Psi(G,k,\mathcal{P},H)$.

G	k	$\lvert P_i\rvert$	h_i^j	$\lvert P_i \cap P_f\rvert$		Status	Theorem
	≤ 2					P	5.1
Tree	2				P_i: chain or bundle; $\Delta(G) \le 3$	NPC	5.2
Cactus	2				$\Delta(G) = 3$; G triangulated	NPC	5.3

TABLE 4. Summary of the results for $\Psi^*(G,k,\mathcal{P},H)$.

G	k	$\lvert P_i\rvert$	h_i^j	$\lvert P_i \cap P_f\rvert$		Status	Theorem
	3	3	1	0	G 3-regular; P_i: triangle	NPC	5.4
Bipartite	3	2		0	G 3-regular	NPC	5.5
Bipartite planar	3	≤ 2			$\Delta(G) \le 3$	NPC	5.6

6. SUMMARY AND CONCLUSION

We have studied an extension of the basic image reconstruction problem of discrete tomography. The complexity status of some variations has been determined; the results are summarized in Table 1 for $\Lambda(G,k,\mathcal{P},H)$. Then Table 2 presents the results for the case of proper colorings ($\Lambda^*(G,k,\mathcal{P},H)$).

Finally for edge k-colorings, Table 3 (resp. Table 4) shows the status of some problems for arbitrary edge k-colorings, i.e. for $\Psi(G,k,\mathcal{P},H)$ (resp. for proper edge k-colorings, i.e. for $\Psi^*(G,k,\mathcal{P},H)$).

There are more cases to examine and it would in particular be interesting to consider a family $\mathcal{P}$ of chains with less restrictive hypotheses in some special classes of graphs. But the results obtained here seem to show that the problems become difficult even in very simple cases.

Acknowledgments

This work was completed in 2005–2006 when M. C. Costa and C. Picouleau were visiting the Institute IMA at the Ecole Polytechnique Fédérale de Lausanne and when D. de Werra was visiting the CEDRIC Laboratory at Conservatoire National des Arts et Métiers in Paris. The support of both institutions is gratefully acknowledged. The authors also thank an anonymous referee whose comments have contributed to improving the presentation of the paper.

REFERENCES

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, Network flows, Prentice-Hall, Englewood Cliffs, 1993.

[2] B. Aspvall, M.F. Plass, and R. Tarjan, A linear-time algorithm for testing the truth of certian quantified Boolean formulas, Informat Process Lett 8 (1979), 121–123.

[3] C. Berge, Graphes, Gauthier-Villars, Paris, 1983.

[4] M. Chrobak and C. Dürr, Reconstructing polyatomic structures from X-rays: NP-completness proof for three atoms, Theor Comput Sci 259 (2001), 81–98.

[5] M.-C. Costa, D. de Werra, and C. Picouleau, Using graphs for some discrete tomography problems, Discrete Appl Math 154 (2006), 35–46.

[6] M.-C. Costa, D. de Werra, C. Picouleau, and D. Schindl, A solvable case of image reconstruction in discrete tomography, Discrete Appl Math 148 (2005), 240–245.

[7] D. de Werra and J. Erschler, Open shop scheduling with some additional constraints, Graphs Combinat 12 (1996), 81–93.

[8] M.R. Garey and D.S. Johnson, Computers and intractability, a guide to the theory of NP-completeness, Freeman, New York, 1979.

[9] P. Hansen and D. de Werra, Nesticity, DIMACS Ser Discrete Math Theor Comput Sci 37 (1997), 225–232.

[10] I. Holyer, NP-completeness of edge-coloring, SIAM J Comput 10 (1981), 718–720.

[11] J. Kratochvil, Precoloring extension with fixed color bound, Acta Math Univ Comenianae LXIII 1 (1994), 139–153.

[12] A. Kuba and G.T. Hermann, Discrete tomography: Foundations, algorithms and applications, Birkhäuser, Boston, 1999.

[13] L. Lovasz and M. Plummer, Matching theory, North Holland, New York, 1986.

[14] D. Marx, NP-completeness of list coloring and precoloring extension on the edges of planar graphs, J Graph Theory 49 (2005), 312–324.

[15] H.J. Ryser, Combinatorial properties of matrices of zeros and ones, Can J Math 9 (1957), 371–377.