The complexity of restrictive H-coloring*

Josep Díaz[†] Maria Serna[†] Dimitrios M. Thilikos[†]

Abstract

We define a variant of the H-coloring problem where the number of preimages of certain vertices is predetermined as part of the problem input. We consider the decision and the counting version of the problem, namely the *restrictive H-coloring* and the *restrictive $\#H$-coloring* problems, and we provide a dichotomy theorem determining the H's for which the restrictive H-coloring problem is either NP-complete or polynomially solvable. Moreover, we prove that the same criterion discriminates the #P-complete and the polynomially solvable cases of the *restrictive $\#H$-coloring* problem. Finally, we prove that both our results apply also for the list versions and other extensions of the above problems.

1 Introduction

Let us consider the following processing setting: we have a host network H of processors with communication links between them, and a set of jobs with communication demands between them, where these jobs and their restrictions in their concurrent execution are modeled by a graph G. We may have further restrictions, for instance in many practical cases, several qualitative restrictions are imposed by the guest network concerning the types of processors that are able to carry out each of the jobs. For this reason, each job may be accompanied with a list of the processors that can perform the task. In real systems the host network wants to keep bounded (or fixed) the load of some of its processors. Thus, some processors may have as an additional quantitative restriction the *number* of jobs assigned to them. The goal is to make a suitable assignment of jobs to processors satisfying all the communication load and all the preference constrains. Historically, one good model to simulate these problems of assignation of interrelated jobs to interconnected processors has been the H-coloring problem [FHH99]. In this paper we propose a model for the full generality problem, incorporating all the above restrictions. It is the first time, such a model is proposed.

Given two graphs G and H, an *homomorphism* from G to H is any function mapping the vertices in G to vertices in H, in such a way that the image of an edge is also an edge. In the case that H is fixed, such an homomorphism is called an H-coloring of G. For a given graph H, the *H-coloring problem* asks if there exists an H-coloring of the input graph G, while the *$\#H$-coloring* asks for the number of the H-colorings of the input graph G. The complexity of both problems depends on the choice of the particular graph H. It is known

*The work of all the authors was supported by the IST Program of the EU under contract number IST-99-14186 (ALCOM-FT) and by the Spanish CYCIT projects TIC-2000-1970-CE and TIC-1999-0754-C03 (MALLBA). The work of the last author was partially supported by the Ministry of Education and Culture of Spain, Grant number MEC-DGES SB98 0K148809.

†Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. Campus Nord Mòdul C6. c/ Jordi Girona Salgado 1-3, 08034, Barcelona, Spain. E-mails: {diaz,mjserna,sedthilk}@lsi.upc.es

that the H-coloring problem is polynomially time solvable if H is bipartite or it contains a loop; otherwise it is NP-complete [HN90]. Its counting version, the $\#H$-coloring problem, is polynomially solvable if all the connected components of H are either complete reflexive graphs or complete irreflexive bipartite graphs [DG00], otherwise the problem is #P-complete.

The more general version in which a list of allowed processors is given for each job is known as the *list H-coloring* problem. The complexity of the *list H-coloring* problem has been studied in [FH98, FHH99, FHH01], and for its counting version, the *list $\#H$-coloring* problem, in [HN01, DST01b].

A variant of the H-coloring problem, the so called *(H,C,K)-coloring* problem and variations whose complexity was studied in [DST01c, DST01b], considers the case in which the quantitative restriction is fixed independently of the graph. See [DST01a] and [Día01] for surveys on different problems based on H-colorings.

In this paper we consider the case in which the additional restriction depend on the graph G, and thus form part of the input. We call this new problem the *restrictive H-coloring* problem. We examine the complexity of the restrictive H-coloring and its variants (see definitions later). We prove that all these problems are polynomial time solvable if all the connected components of the host graph H are either complete reflexive graphs or complete irreflexive bipartite graphs. Moreover, we prove that in any other case the decision problems are NP-complete and the counting problems are #P-complete. Observe that, in contrast to the non restrictive problems, the dichotomy result attained for this problem is the same for both list and non list problems, as well as, for counting an decision problems.

2 Definitions

All the graphs in this paper are finite, undirected, and cannot have multiple edges but can have loops. We call *reflexive* a graph with all its vertices looped. If none of the vertices of a graph is looped then we call it irreflexive. We use the notations $V(G)$ and $E(G)$ for the vertex and the edge set of a graph G respectively. Also we set $n = |V(G)|$. For a given graph G and a vertex subset $S \subseteq V(G)$, the subgraph induced by S is the graph $G[S] = (S, E(G) \cup S \times S)$. We use standard notation for graphs, K_n^r is a reflexive clique on n vertices and K_n, m is the complete (irreflexive) complete bipartite graph, with bipartitions of size n and m.

Given two graphs G and H, an *homomorphism* from G to H is any function $\sigma : V(G) \to V(H)$ where for any edge $\{v,u\} \in E(G)$, $\{\sigma(v), \sigma(u)\}$ is also an edge of H. For fixed H we say that σ is an *H-coloring* of G.

For a fixed graph H, the *H-coloring problem* asks for the existence of an H-coloring of the input graph G, while the *$\#H$-coloring* asks for the number of the H-colorings of the input graph G.

For a fixed graph H, and given a graph G, a *list of preferences* is a function $L : V(G) \to 2^{V(H)}$. Given the pair (G,L) a *list H-coloring* of (G,L) is an homomorphism σ from G to H such that for any $v \in V(G)$, $\sigma(v) \in L(v)$.

For a fixed graph H, and an input formed by a graph G and an associated list of preferences L, the *list H-coloring* problem asks for the existence of a list H-coloring of the input, while the *list $\#H$-coloring* asks for the number of list H-colorings of the input.

For a fixed graph H, and given a graph G with n vertices, a *weighting function* is a function, $\mathsf{w} : V(H) \to \{0, \ldots, n, \infty\}$. Given the pair (G, w), a *restrictive H-coloring* of (G, w) is an H-coloring σ of G such that for all $a \in V(H)$ with $\mathsf{w}(a) \neq \infty$, $|\sigma^{-1}(a)| = \mathsf{w}(a)$, when

$\mathsf{w}(a) \neq \infty$, $\sigma^{-1}(a)$ can have any number of vertices. Notice that ∞ is used to represent the lack of restrictions on the number of preimages of a vertex, as usual we will assume that for any natural n it holds $n < \infty$. Given a graph G, a preference list L, and a weighting function w a *restrictive list H-coloring* of the triple (G, L, w) is a list H-coloring σ of (G, L) such that σ is also a (G, w) restrictive H-coloring.

For a fixed graph H, and given a graph G with n vertices, a *weighting list* is a function $\mathsf{W} : V(H) \rightarrow 2^C\}$, where $C = \{(x, y) \mid x, y \in \mathbb{N}0 \leq x \leq y \leq n\}$, that assigns to each vertex in H a collection of interval ranges. Given the pair (G, W), a *loosely restrictive H-coloring* of (G, w) is an H-coloring σ of G such that for all $a \in V(H)$ there is an $(x, y) \in W(a)$ such that $|\sigma^{-1}(a)| \in [x, y]$. Given the triple (G, L, W), where G is a graph G, L is a preference list for G, and W is a weighting list for H, a *loosely restrictive list H-coloring* of the triple (G, L, W) is a list H-coloring σ of (G, L) such that σ is also a (G, W) loosely restrictive H-coloring.

Now we can introduce the new problems:

Name: Restrictive H-coloring problem
Input: A graph G and a weighting function w on H
Question: Does (G, w) have a restrictive H-coloring?

Name: Restrictive list H-coloring problem
Input: A graph G, a list L on G, and a weighting function w on H
Question: Does (G, L, w) have a restrictive list H-coloring?

Name: Loosely restrictive H-coloring problem
Input: A graph G and a weighting list W on H
Question: Does (G, W) have a loosely restrictive H-coloring?

Name: Loosely restrictive list H-coloring problem
Input: A graph G, a list L on G, and a weighting list H on H
Question: Does (G, L, W) have a loosely restrictive list H-coloring?

The counting versions of the four problems, the restrictive $\#H$-coloring, the restrictive list $\#H$-coloring, the loosely restrictive $\#H$-coloring and the loosely restrictive list $\#H$-coloring are defined as usual.

Notice that any given weighting function can be converted into a weighting list, therefore the loose versions are a generalization of the restrictive ones. Furthermore, by setting $\mathsf{w}(a) = \infty$ for all $a \in H$, the restrictive H-coloring problems solves the corresponding H-coloring problem, therefore we can translate all the hardness results to the restrictive problem versions. In particular, the #P-hardness results in [DG00, HN01, DST01b] translates in the following result.

Theorem 1. If H has a connected component that is not a complete (irreflexive) bipartite graph or a complete reflexive clique then the (loosely) restrictive $\#H$-coloring and the (loosely) restrictive list $\#H$-coloring problems are #P-hard.

In the remaining of the paper we will show that the above condition gives a dichotomy condition for the problem to be in P or to be hard for the eigth restrictive problems.

3 NP-completeness results

In this section we show that when H has a connected component that is not a complete (irreflexive) bipartite graph or a complete reflexive clique the restrictive H-coloring problem,

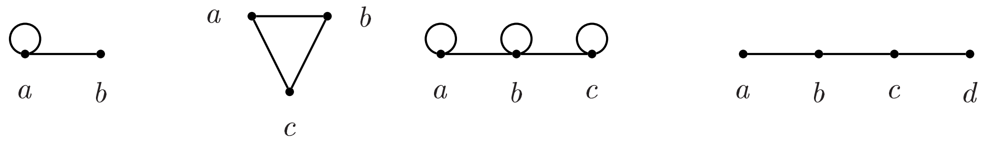

Figure 1: The four forbidden subgraphs of Lemma 1

and therefore the other three restrictive H-coloring decision problems are NP-complete. As this four problems are clearly in NP, we provide only the hardness proofs.

It is well known that a connected graph that excludes the four forbidden subgraphs given in Figure 1 characterize the following property [Neš01].

Lemma 1. All the connected components of a graph H are either a complete reflexive graph or a complete irreflexive bipartite graph iff H does not contain as induced subgraphs any of the graphs given in Figure 1.

Now we can state the NP-completeness result in this section.

Theorem 2. If H contains, as an induced subgraph, any of the graphs in Figure 1, then the restrictive H-coloring problem is NP-complete.

Proof. We will distinguish four cases depending on which of the graphs in Figure 1 appears as an induced subgraph of H. Observe that we can seelect a particular induced subgraph of H by setting to zero the number of tasks that a processor can perform.
Case 1. If $\{a, b\}$ is an edge in H where a is looped and b is unlooped then we define

$$\mathsf{w}(v) = \begin{cases} \infty & \text{if } v = a, \\ k & \text{if } v = b, \\ 0 & \text{otherwise.} \end{cases}$$

In this case (G, w) has a restrictive H-coloring iff G has an independent set of size at least k.
Case 2. If $\{a, b, c\}$ form a triangle in H then we set

$$\mathsf{w}(v) = \begin{cases} \infty & \text{if } v \in \{a, b, c\}, \\ 0 & \text{otherwise.} \end{cases}$$

In this case (G, w) has a restrictive H-coloring iff G is 3-colorable.
Case 3. Let now $\{a, b, c\}$ be an induced reflexive path in H. We will reduce the following problem to the restrictive H-coloring problem.

> **Name:** Balanced Separator
> **Input:** Graph G and positive integer $k \leq n$.
> **Question:** Is there a partition of $V(G)$ in three sets A, B, C, with $|C| = k$, and such that the removal of C leaves a graph with no edges between A and B, and $\max\{|A|, |B|\} \leq |V|/2$.

The above problem can be shown NP-complete by a slight variation of the NP-hardness proof given in [BJ92] for the *minimum B-vertex separator* problem.

Let G be an input of the above problem, we construct a new graph $\tilde{G}$ with $k+1$ new vertices, $V(\tilde{G}) = V(G) \cup \{u_0, \dots, u_k\}$, and with edge set $E(\tilde{G}) = E(G) \cup \{\{u_0, x\} \mid x \in V(G)\} \cup \{\{u_0, u_i\} \mid 1 \le i \le k\}$.

For any $v \in V(H)$, we set

$$\mathsf{w}(v) = \begin{cases} n/2 & \text{if } v = a, \\ n/2 & \text{if } v = b, \\ k+1 & \text{if } v = c, \\ 0 & \text{otherwise.} \end{cases}$$

and we claim that G has a balanced separator if and only if $(\tilde{G}, \mathsf{w})$ has a restrictive H-coloring.

For the if part, assume that G has a balanced separator, then map all the vertices in A to a, all the vertices in B to b and all the vertices in C to c. The vertex u_0 is mapped to c, an the remaining vertices are splitted between a, b and c to attain the demanded sizes.

For the only if part, in the case that $(\tilde{G}, \mathsf{w})$ has a restrictive H-coloring σ, by construction all the vertices must be mapped to a, b or c. Then, define $A = \sigma^{-1}(a) \cap V(G)$, $B = \sigma^{-1}(b) \cap V(G)$ and $C = \sigma^{-1}(c) \cap V(G)$. As a and b are not connected, we get that C is a balanced separator. However, C might have less than k vertices. In such a case we can move vertices from A and/or B to pad C to the demanded size.

Case 4. Let now $\{a, b, c, d\}$ be an induced irreflexive path in H. We consider the following NP-complete problem [GJ79]:

> **Name:** Balanced Complete Bipartite Subgraph
> **Input:** Bipartite connected graph $G = (V_1, V_2, E)$ and positive integer k, such that $k \le |V_1| + |V_2|$.
> **Question:** Does G contain $K_{k,k}$ as an induced subgraph?

Let (G, k) be an input of the above problem. Assume that G has bipartition V_1 and V_2. Let u_1, u_2 be two new vertices not in $V(G)$. We construct a new bipartite graph $\tilde{G} = (W, F)$ with bipartition $W_1 = V_1 \cup \{u_1\}$ and $W_2 = V_2 \cup \{u_2\}$, and with edge set

$$\begin{aligned} F = {} & \{\{u_1, x\} \mid x \in V_2\} \cup \{\{x, u_2\} \mid x \in V_1\} \\ & \cup \{\{u_1, u_2\}\} \\ & \cup \{\{x, y\} \mid x \in V_1, y \in V_2, \text{ and } \{x, y\} \notin E(G).\} \end{aligned}$$

Notice that $\tilde{G}$ is the bipartite complement of G with two new adjacent vertices v and u, such that v is connected with all the vertices in the one part and u with all the vertices in the other.

For all $v \in V(H)$, we set

$$\mathsf{w}(v) = \begin{cases} k & \text{if } v = a, \\ \infty & \text{if } v = b, \\ \infty & \text{if } v = c, \\ k & \text{if } v = d, \\ 0 & \text{otherwise.} \end{cases}$$

and we claim that G contains $K_{k,k}$ as a subgraph if and only if $\tilde{G}$ has a restrictive H-coloring.

For the if part, suppose that G contains $K_{k,k}$ as a subgraph and let, $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ be the bipartition subsets producing the $K_{k,k}$. Define the function $\sigma : V(G) \to V(H)$ as

$$\sigma(u) = \begin{cases} a & \text{if } u \in U_1, \\ d & \text{if } u \in U_2, \\ c & \text{if } u \in V_1 - U_1, \\ b & \text{if } u \in V_2 - U_2, \\ b & \text{if } u = u_1, \\ c & \text{if } u = u_2. \end{cases}$$

We prove that σ is an H-coloring of $\tilde{G}$. Assume that $\{x, y\} \in E(\tilde{G})$, we consider four cases:

- If $\{x, y\} = \{u_1, u_2\}$ then $\{\sigma(x), \sigma(y)\} = \{b, c\} \in E(H)$.
- If $x = u_1$ and $y \in V_2$ then $\sigma(x) = b$ and $\sigma(y) \in \{a, c\}$. Therefore $\{\sigma(x), \sigma(y)\} \in E(H)$.
- If $y = u_2$ and $x \in V_2$ then $\sigma(x) = c$ and $\sigma(y) \in \{b, d\}$. Therefore $\{\sigma(x), \sigma(y)\} \in E(H)$.
- If $x \in V_1$ and $x \in V_2$ then $x \notin U_1$ and $y \notin U_2$. Therefore $x \in V_1 - U_1$ and $y \in V_2 - U_2$. We conclude that $\{\sigma(x), \sigma(y)\} = \{b, c\} \in E(H)$

Therefore, in any case, we have $\{\sigma(x), \sigma(y)\} \in E(H)$and conclude that σ is an H-coloring of $\tilde{G}$. Moreoever, as $|\sigma^{-1}(a)| = |U_1| = |\sigma^{-1}(d)| = |U_2| = k$ then σ is a restrictive H-coloring of $(\tilde{G}, \mathsf{w})$.

For the only if part, suppose now that $\sigma : V(G) \to V(H)$ is a restrictive H-coloring of $\tilde{G}$. We first prove that $\sigma(\{u_1, u_2\}) = \{b, c\}$. Indeed, if one, say u_1, of u_1, u_2, is mapped to one, say a, of a, d, then none of the vertices of $\tilde{G}$ can be mapped to d as all the vertices of $\tilde{G}$ are of distance at most 2 from t_1. W.l.o.g. we assume that $\sigma(u_1) = b$ and $\sigma(u_2) = c$.

Let U_2 be the set formed with the neighbors of u_1 that are mapped to a. Notice that $U_2 \subseteq V_2$ and $|U_2| = k$. Similarly let U_1 be the subset with the neighbors of u_2 that are mapped to b. Observe that $U_1 \subseteq V_1$ and $|U_1| = k$. As σ is a H-coloring and $\{a, d\}$ is not an edge of H, there is no edge in $\tilde{G}$ connecting a vertex in U_1 with a vertex in U_2. Therefore, in G all the vertices of U_1 are connected with all the vertices in U_2 which implies that $G[U_1 \cup U_2]$ is K_{kk}. □

By using the facts that the restrictive list H coloring problem can solve the restrictive H-coloring problem, and that the loosely version are generalizations of the restrictive and the restrictive list problems, we obtain the NP-hardness.

Theorem 3. If H has a connected component that is not a complete (irreflexive) bipartite graph or a complete reflexive clique then the (loosely) restrictive H-coloring and the (loosely) restrictive list H-coloring problems are NP-complete.

4 Restrictive H-coloring: the connected case

In this section we solve in polynomial time the counting version of the restrictive H-coloring problem in the case that H does not contain as a subgraph any of the forbidden subgraphs in Figure 1 and, furthermore, G is connected.

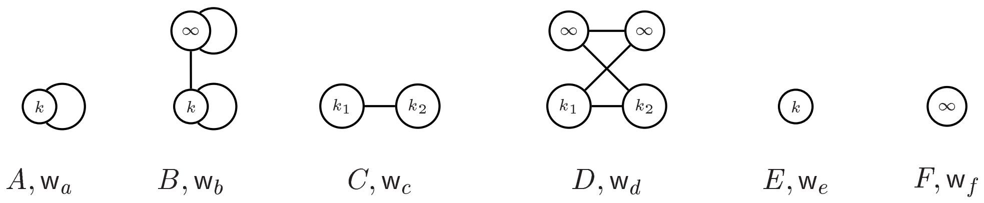

Figure 2: The six basic cases for counting H-colorings

Let us first show that for any of the dif and only iferent graphs and weighting assignments shown in Figure 2, the number of restrictive H-colorings of a graph G can be computed in polynomial time.

Given two graphs G, H and a weighting assignment w on $V(H)$, $\mathsf{H}(G, H, \mathsf{w})$ will denote the number of restrictive H-colorings of (G, w). We set $n = |V(G)|$, and for a connected bipartite graph G, we set n_1, n_2 to be the sizes of the two bipartitions. We start solving the counting problem for the six graphs depicted in Figure 2. For each one of them we provide a polynomial time algorim that correctly computes the number of restrictive colorings.

For the graph A, given the pair (G, w_a), G has only one A-coloring that map all the vertices of G to the vertex in A. The unique restriction is in the number of allowed preimages, so either there are one or zero restrictive colorings. Therefore the following algorithm computes in polynomial time the number of restrictive A-colorings of (G, w_a).

function $\mathsf{H}(G, A, \mathsf{w}_a)$
  if $n = k$
    then return(1)
    else return(0)
  end if
end

For the graph B, given the pair (G, w_b), the situation is similar to the previous one. The B-colorings must map k vertices to the vertex with weight k and the remaining vertices to the other vertex. Therefore, the following algorithm computes in polynomial time the number of restrictive B-colorings of (G, w_b).

function $\mathsf{H}(G, B, \mathsf{w}_b)$
  if $n \leq k$
    then return$(\binom{n}{k})$
    else return(0)
  end if
end

For the graph C, given the pair (G, w_c), as C is bipartite it is required that G is bipartite and that $n = k_1 + k_2$. To accomodate G, we have to control the sizes of its bipartitions. Then, the following algorithm computes in polynomial time the number of restrictive C-colorings of (G, w_c).

function $\mathsf{H}(G, C, \mathsf{w}_c)$

if G is not bipartite and $n \neq k_1 + k_2$
then return(0)
elsif $(n_1 \neq k_1$ and $n_1 \neq k_2)$ or $(n_2 \neq k_1$ and $n_2 \neq k_2)$ **then return**(0)
elsif $(n_1 = k_1$ and $n_1 \neq k_2$ and $n_2 = k_2)$ **then return**(1)
elsif $(n_1 \neq k_1$ and $n_1 = k_2$ and $n_2 = k_1)$ **then return**(1)
else return(2)
end if
end

For the graph D, given the pair (G, w_d), the situation is similar to the previous one. The D-colorings must map k_1 vertices in one bipartition to the vertex with weight k_i and the remaining vertices in the bipartition to the unbounded vertex. Therefore, the following algorithm computes in polynomial time the number of restrictive D-colorings of (G, w_d).

function $\mathsf{H}(G, D, \mathsf{w}_d)$
if G is not bipartite and $n \neq k_1 + k_2$
then return(0)
elsif $(n_1 < k_1$ and $n_1 < k_2)$ or $(n_2 < k_1$ and $n_2 < k_2)$ **then return**(0)
elsif $(n_1 \geq k_1$ and $n_1 < k_2$ and $n_2 \geq k_2)$ **then return** $\left(\binom{n_1}{k_1} \cdot \binom{n_2}{k_2}\right)$
elsif $(n_1 < k_1$ and $n_1 \geq k_2$ and $n_2 \geq k_1)$ **then return** $\left(\binom{n_2}{k_1} \cdot \binom{n_1}{k_2}\right)$
else return $\left(\binom{n_1}{k_1} \cdot \binom{n_2}{k_2} + \binom{n_2}{k_1} \cdot \binom{n_1}{k_2}\right)$
end if
end

For the graphs E and F the situation is simpler. For the existence of a coloring the graph G must be an isolated vertex. Thus, the following algorithms compute in polynomial time the number of restrictive E-colorings of (G, w_e) and F-colorings of (G, w_f), respectively.

function $\mathsf{H}(G, E, \mathsf{w}_e)$
if G has no edges and $k = 1$
then return(1)
else return(0)
end if
end

function $\mathsf{H}(G, F, \mathsf{w}_f)$
if G has no edges
then return(1)
else return(0)
end if
end

Therefore, we can compute in polynomial time the number of restrictive H-colorings of a given pair (G, w_h), for any $H \in \{A, B, C, D, E, F\}$ (the given in Figure 2). These particular cases are the main ingredient in the polynomial time algorithm to compute the number of restrictive H colorings when G is a connected graph.

Lemma 2. Let H be a reflexive clique, given a connected graph G and a weighting function w on H, then $\mathsf{H}(G, H, \mathsf{w})$ can be computed in polynomial time.

Proof. Let $C = \{a \in V(H) \mid \mathsf{w}(a) \neq \infty\}$, let $k = \sum_{a \in C} \mathsf{w}(a)$, and let $\alpha = |V(H) - C|$. We will consider two cases.
Case 1. $C = V(H)$. In this case collapsing all the vertices in H into a single vertex and assigning to it weight k we get the graph A in Figure 2 with a weighting assignment w_a. Observe that any restrictive A-coloring of (G, w_a) can be extended into $k!$ ways to obtain a valid restrictive H-coloring of (G, w), and any valid H-coloring of (G, w) can be contracted to provide a valid restrictive A-coloring of (G, w_a).
Case 2. $C \neq V(H)$. In this case by collapsing all the vertices in C to a vertex with weight k and all the remaining vertices vertices in $V(H) - C$ to a vertex with weight ∞, we obtain the graph B in Figure 2 with a weight assignment w_b. Observe that any restrictive B-coloring of (G, w_b) can be extended into $k!\alpha^{n-k}$ ways to obtain a valid restrictive H-coloring of (G, w), and that any valid H-coloring of (G, w) can be contracted to provide a valid restrictive B-coloring of (G, w_b).

Therefore, the following polynomial time algorithm computes $\mathsf{H}(G, H, \mathsf{w})$.

```
function H(G, H, w)
    Compute n, k and α;
    if α = 0
      then if n = k
             then return(k!)
             else return(0)
               end if
             elsif n ≤ k
               then return (k! α^(n−k) (n choose k))
               else return(0)
          end if
      end
```

□

Lemma 3. Let H be a complete (irreflexive) bipartite graph with more than one vertex, given a connected graph G and a weighting function w on H, $\mathsf{H}(G, H, \mathsf{w})$ can be computed in polynomial time.

Proof. Let $H = (V_1, V_2, E)$. For $i = 1, 2$, let $C_i = \{a \in V_i \mid \mathsf{w}(a) \neq \infty\}$, let $k_i = \sum_{a \in C_i} \mathsf{w}(a)$, and let $\alpha_i = |V_i - C_i|$. We will consider two cases.
Case 1. $C_1 = V_1$ and $C_2 = V_2$. In this case collapsing all the vertices in V_1 to a vertex with weight k_1 and collapsing all the vertices in V_2 to a vertex with weight k_2 we obtain the graph C in Figure 2 and a weight assignment w_c. Observe that any restrictive C-coloring of (G, w_c) can be extended int $k_1!\, k_2!$ ways to obtain a valid restrictive H-coloring of (G, w), and any valid H-coloring of (G, w) can be contracted to provide a valid restrictive C-coloring of (G, w_c).
Case 2. $C_1 \neq V_1$ or $C_2 \neq V_2$. In this case by collapsing all the vertices in C_i to a vertex with weight k_i and all the remaining vertices in V_i to an unbounded vertex, we obtain the

graph D in Figure 2 with a weight assignment w_d. Observe that any restrictive D-coloring of (G, w_d) can be extended in $k_1!\, k_2!\, \alpha_1^{n_1-k_1}\, \alpha_2^{n_2-k_2}$ ways to obtain a valid restrictive H-coloring of (G, w), and that any valid H-coloring of (G, w) can be contracted to provide a valid restrictive D-coloring of (G, w_d).

Therefore, the following polynomial time algorithm computes $\mathsf{H}(G, H, \mathsf{w})$.

```
function H(G, H, w)
    if G is not bipartite
      then return 0
      else
        Compute n1, n2, k1, k2, α1 and α2;
        if α1 = α2 = 0
          then
            if (n1 ≠ k1 and n1 ≠ k2) or (n2 ≠ k1 and n2 ≠ k2)
              then return(0)
              elsif (n1 = k1 and n1 ≠ k2 and n2 = k2) then return(k1! k2!)
              elsif (n1 ≠ k1 and n1 = k2 and n2 = k1) then return(k1! k2!)
              else return(2 k1! k2!)
            end if
          elsif (n1 < k1 and n1 < k2) or (n2 < k1 and n2 < k2)
            then return(0)
            elsif (n1 ≥ k1 and n1 < k2 and n2 ≥ k2)
              then return (k1! k2! α1^(n1−k1) α2^(n2−k2) C(n1,k1) · C(n2,k2))
              elsif (n1 < k1 and n1 ≥ k2 and n2 ≥ k1)
                then return (k1! k2! α1^(n1−k1) α2^(n2−k2) C(n2,k1) · C(n1,k2))
                else return (k1! k2! α1^(n1−k1) α2^(n2−k2) (C(n1,k1) · C(n2,k2) + C(n2,k1) · C(n1,k2)))
            end if
        end if
    end if
end
```

$\square$

In the case that H is an isolated vertex, G must also be an isolated vertex and we can compute $\mathsf{H}(G, E, \mathsf{w}_e)$ and $\mathsf{H}(G, F, \mathsf{w}_f)$ in polynomial time.

Now we are ready to prove the main result in this section.

Theorem 4. If all the connected components of H are either a complete (irreflexive) bipartite graph or a complete reflexive clique, then the restrictive $\#H$-coloring problem can be solved in polynomial time.

Proof. Assume that H has l connected components. Given a weighting function w on H, let w_j denote the restriction of w to the vertices in H_j. As the given graph G is connected, it can be mapped only to one connected component of H, therefore we only have to count the number of restrictive H_j colorings of (G, w_j) that fulfill the weight bounds with an empty assignment of vertices in G to the remaining components.

We classify the connected components of H as follows: H_j is *free* if $\mathsf{w}(H_j) = \{\infty\}$; H_j is *forbidden* if $\mathsf{w}(H_j) = \{0\}$; otherwise H_j is *restricted*. Therefore, we have

$$\mathsf{H}(G, H, \mathsf{w}) = \begin{cases} \sum_{1 \leq j \leq l} \mathsf{H}(G, H_j, \mathsf{w}_j) & \text{if all the components are free or forbidden,} \\ \mathsf{H}(G, H_j, \mathsf{w}_j) & \text{if } H_j \text{ is the unique restricted component,} \\ 0 & \text{if more than one component is restricted.} \end{cases}$$

By Lemmas 2 and 3 the last formula can be evaluated in polynomial time. □

Notice that counting in polynomial time implies deciding in polynomial time, so we get the same result for the decision versions.

Corollary 1. If all the connected components of H are either a complete (irreflexive) bipartite graph or a complete reflexive clique, then the restrictive H-coloring problem can be solved in polynomial time.

5 H-coloring: the general case

Now we show how to compute the number of restrictive H-colorings in the general case in which the graph G might not be connected. Observe that in a restrictive H-coloring of G a connected component of G may only proporcionat a part of the demanded number of preimages. Due to this fact we are forced to take into consideration H-colorings of components of G that are H-colorings but that fill only part of the number of preimages required by w.

Theorem 5. If all the connected components of H are either a complete (irreflexive) bipartite graph or a complete reflexive clique, then the restrictive $\#H$-coloring problem can be solved in polynomial time.

Proof. Given (G, w), we consider the set

$$\mathrm{B}(\mathsf{w}) = \{f : V(H) \to \{0, \dots, n\} \mid \text{ for all } a \in V(H) \ f(a) \leq \mathsf{w}(a)\}$$

Given two functions $\mathsf{w}_1, \mathsf{w}_2 \in \mathrm{B}(\mathsf{w})$, as usual, the function $\mathsf{w}_1 + \mathsf{w}_2$ is defined, for any $a \in V(H)$, as $(\mathsf{w}_1 + \mathsf{w}_2)(a) = \mathsf{w}_1(a) + \mathsf{w}_2(a)$.

To keep an uniform notation we will assume that all the weighting functions are defined over $V(H)$. To fullfil this goal any weighting function of a connected component H_j is extended to H by assigning the weight 0 to all the vertices outside $V(H_j)$. We say that a weighting function w defined over H is *proper* for H_j if for all $u \in V(H) - V(H_j)$, $\mathsf{w}(u) = 0$. We will represent by $\mathrm{P}(j)$ the set of proper functions for the component G_j, $1 \leq j \leq l$.

We also assume that G has m connected components $G_1, \dots, G_m$, and use the notation G^i to denote the graph formed by the disjoint union of $G_1, \dots, G_i$. To compute, for given G and w, $\mathsf{H}(G, H, \mathsf{w})$, we compute initially a table, $T[i, j, m]$, such that for any $1 \leq i \leq m$, $1 \leq j \leq l$ and $f \in \mathrm{B}(\mathsf{w}))$ we have

$$T[i, j, f] = \begin{cases} \mathsf{H}(G_i, H_j, f) & f \in \mathrm{P}(j) \\ 0 & \text{otherwise.} \end{cases}$$

By Theorem 4, $T[i, j, f]$ can be computed in polynomial time, for any f. As G has n vertices, the size of $\mathrm{B}(\mathsf{w})$ is at most n^h and therefore polynomial in the input size, so the whole table can be computed in polynomial time.

Now we can compute using dynamic programming techniques a table $S[i, f]$, for $1 \leq i \leq m$ and $f \in E(\mathsf{w})$, such that $S[i, f]$ keeps the number of restrictive H-colorings of (G^i, f). To get the formula we have only to take into account that a connected component of G must be mapped entirely to a unique connected component of H. For any $f \in E(\mathsf{w})$

$$S[1, f] = \sum_{1 \leq j \leq l} T[1, j, f]$$

and for any $1 < j \leq m$

$$S[j, f] = \sum_{1 \leq j \leq l, f_1 + f_2 = f} S[j-1, f_1] \cdot T[i, j, f_2].$$

As the size of $E(\mathsf{w})$ is polynomial, table S can be computed in polynomial time.

To finish, let $A(\mathsf{w}) = \{f \in E(\mathsf{w}) \mid \mathsf{w}(a) = f(a) \text{ for all } a \in H \text{ with } \mathsf{w}(a) \neq \infty\}$ be the set of weighting bounded functions fullfiling the restrictions, then

$$\mathsf{H}(G, H, \mathsf{w}) = \sum_{\{f \in A(\mathsf{w})\}} S[m, f],$$

which again can be computed in polynomial time. □

Putting together Theorems 1, 3 and 5, we get the dichotomy result.

Theorem 6. If all connected components of H are either a complete (irreflexive) bipartite graph or a complete reflexive clique, then the restrictive H-coloring and the restrictive $\#H$-coloring) problems can be solved in polynomial time, otherwise they are NP-complete or #P-complete, respectively.

6 The restrictive list H-coloring problem

Now we will show how to extend the previous result to counting restrictive list H-colorings. The main difficulty here is that the vertices in a connected component of H cannot be collapsed to a single vertex, because this may put together vertices that are not in the same vertex list. Once we have solved the connected case the second step is identical to the non connected case for the restrictive H-coloring.

We will consider the two main types of connected components and show that a dynamic programming approach allow us to compute the number of restrictive list H-colorings. Making an abuse of notation we will represent by $\mathsf{H}(G, H, \mathsf{w}, L)$ the number of restrictive list H-colorings of a triple (G, w, L).

Lemma 4. Let H be a reflexive clique, given a connected graph G a weighting function w on H and a list selection for G, then $\mathsf{H}(G, H, \mathsf{w}, L)$ can be computed in polynomial time.

Proof. As H is a reflexive clique we can assign a vertex of G to any vertex in H provided the additional restrictions are fulfilled. Let $V(G) = \{u_1, \ldots, u_n\}$, and define the set

$$\mathrm{B}(\mathsf{w}) = \{f : V(H) \to \{0, \ldots, n\} \mid \text{ for all } a \in V(H)\ f(a) \leq \mathsf{w}(a)\}.$$

For any $a \in H$ define f_a by

$$f_a(b) = \begin{cases} 1 & \text{if } b = a \\ 0 & \text{otherwise.} \end{cases}$$

We want to compute a table $R[i, f]$, $1 \leq i \leq n$, $f \in \mathrm{B}(\mathsf{w})$, that counts the number of restrictive list H-colorings for the triple $(G[\{u_1, \dots, u_i\}], f, L)$. The recurrence is the following: for any $f \in \mathrm{B}(\mathsf{w})$

$$R[1, f] = \begin{cases} 1 & \text{if } f = f_a \text{ for some } a \in L(u_1), \\ 0 & \text{otherwise.} \end{cases}$$

and, for any $1 < j \leq m$

$$R[i, f] = \sum_{f_1+f_2=f, f_2=f_a \text{ for some } a \in L(u_i)} R[j-1, f_1].$$

As the size of $\mathrm{B}(\mathsf{w})$ is polynomial, we can fill the table R in polynomial time.

Setting $\mathrm{A}(\mathsf{w}) = \{f \in E(\mathsf{w}) \mid \mathsf{w}(a) = f(a) \text{ for all } a \in H \text{ with } \mathsf{w}(a) \neq \infty\}$ we have

$$\mathsf{H}(G, H, \mathsf{w}, L) = \sum_{f \in \mathrm{A}(\mathsf{w})} R[n, f].$$

□

Lemma 5. Let H be a complete (irreflexive) bipartite graph with more than one vertex, given a connected graph G, a weighting function w on H, and a list L for G, then $\mathsf{H}(G, H, \mathsf{w}, L)$ can be computed in polynomial time.

Proof. If H is bipartite, then G must be bipartite. In this case, we can work separately with the two possible assignments of bipartitions of G with bipartitions of H. Notice that once the global assignment is set, any vertex can be mapped to any one in the assigned partition, thus working as in the previous lemma we can compute $\mathsf{H}(G, H, \mathsf{w})$ in polynomial time. □

Using the same techniques of Section 5, we can obtain the polynomial time result, this together with Theorems 1 and 3 give the dichotomy for the list version of the problem.

Theorem 7. If all the connected components of H are either a complete (irreflexive) bipartite graph or a complete reflexive clique, then the restrictive list H-coloring and the restrictive list $\#H$-coloring problems can be solved in polynomial time, otherwise they are NP-complete and $\#\mathsf{P}$-complete, respectively.

7 Loosely restrictive H-coloring

In this section we consider the general version of the restrictive H-coloring problem.

Theorem 8. If all the connected components of H are either a complete (irreflexive) bipartite graph or a complete reflexive clique, then the loosely restrictive $\#H$-coloring and the loosely restrictive list $\#H$-coloring problems can be solved in polynomial time, otherwise they are $\#\mathsf{P}$-complete, respectively.

Proof. The negative part follows from Theorem 1. Assume that all the connected components of H are either a complete (irreflexive) bipartite graph or a complete reflexive clique, and that the input is a graph G together with a weighting list W. Recall that in a loosely restrictive H-coloring σ of G, it is required that for any $a \in H$, $|\sigma^{-1}(a)|$ falls inside one of the prescribed integer ranges for a in W.

Let u_n be the weigthing assignment that assigns weight n to any $a \in V(H)$, define the set

$$\mathrm{B}(\mathsf{u}_n) = \{f : V(H) \to \{0, \dots, n\} \mid f(a) \leq n \text{ for any } a \in V(H)\}.$$

Using the algorithm described in the proof of Theorem 5 we can compute in polynomial the values $S[m, f]$ holding the number of restrictive H-colorings of (G, f), for any function $f \in E(\mathsf{u})$.

Define the set of allowed functions

$$\mathrm{A}(\mathsf{W}) = \{f \in \mathrm{B}(\mathsf{u}) \mid \forall a \in V(H)\ \exists (x, y) \in \mathsf{W}(a)\ x \leq f(a) \leq y\}$$

Then, the number of loosely restrictive H-colorings is

$$\sum_{f \in \mathrm{A}(\mathsf{W})} S[m, f]$$

and this sum can also be computed in polynomial time.

For the list version, using the algorithm described in the proof of Theorem 7 we can compute in polynomial the values $R[n, f]$ holding the number of restrictive list H-colorings of (G, f), for any function $f \in E(\mathsf{u})$. Then, the number of loosely restrictive list H-colorings is

$$\sum_{f \in \mathrm{A}(\mathsf{W})} R[n, f]$$

and this sum can also be computed in polynomial time. □

As a corollary of the previous result and of Theorem 3 we get the dichotomy for the decisonal version of the problems.

Theorem 9. If all the connected components of H are either a complete (irreflexive) bipartite graph or a complete reflexive clique, then the loosely restrictive H-coloring and the loosely restrictive list H-coloring problems can be solved in polynomial time, otherwise they are NP-complete.

8 Further variations and conclusions

Two variants of the list H-coloring problem have been considered in the literature: the connected list H-coloring problem, in which any set in the list must induce a connected subgraph; and the one-all list H-coloring, in which all the sets in the list either contain all the vertices in G or exactly one vertex. The two problems were introduced in [FH98]. The results presented in this paper also hold for the restrictive and loosely restrictive decision and counting version of both problems.

The running time of the algorithms in this has the form $O(n^{h+c})$, where c is a constant independent of h. It is an open problem to characterize the family of graphs for which a time bound of the form $O(f(h)n^c)$ can be achieved, where f depends on h but it is independent of n.

References

[BJ92] Thang Nguyen Bui and Curt Jones. Finding good approximate vertex and edge partitions is NP-hard. *Information Processing Letters*, 42(3):153–159, 1992.

[DG00] Martin Dyer and Catherine Greenhill. The complexity of counting graph homomorphisms. *Random Structures Algorithms*, 17:260–289, 2000.

[Día01] Josep Díaz. H-colorings of graphs. *Bulletin of the European Association for Theoretical Computer Science*, (75):82–92, 2001.

[DST01a] Josep Díaz, Maria Serna, and Dimitrios M. Thilikos. The complexity of parameterized H-colorings: A survey. In J. Nešetřil and P. Winkler, editors, *Graphs, Morphisms and Statistical Physics*, DIMACS series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 2001. To appear.

[DST01b] Josep Díaz, Maria Serna, and Dimitrios M. Thilikos. Counting list H-colorings and variants. Technical Report LSI-01-27-R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, 2001.

[DST01c] Josep Díaz, Maria Serna, and Dimitrios M. Thilikos. (H, C, K)-colorings: Fast, easy and hard cases. In *Mathematical Foundations of Computer Science 2001, MFCS-2001*, volume 2136 of *Lecture Notes in Computer Science*, pages 304–315. Springer-Verlag, 2001.

[FH98] Tomas Feder and Pavol Hell. List homomorphisms to reflexive graphs. *Journal of Combinatorial Theory (series B)*, 72(2):236–250, 1998.

[FHH99] Tomas Feder, Pavol Hell, and Jing Huang. List homomorphisms and circular arc graphs. *Combinatorica*, 19:487–505, 1999.

[FHH01] Tomas Feder, Pavol Hell, and Jing Huang. Bi-arc graphs and the complexity of list homomorphism. manuscript, 2001.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

[HN90] P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory (series B)*, 48:92–110, 1990.

[HN01] Pavol Hell and J. Nešetřil. Counting list homomorphisms and graphs with bounded degrees. In J. Nešetřil and P. Winkler, editors, *Graphs, Morphisms and Statistical Physics*, DIMACS series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 2001. To appear.

[Neš01] Jaroslav Nešetřil, 2001. personal communication.