Complexity Issues on Bounded Restrictive H-coloring*

Josep Díaz† Maria Serna† Dimitrios M. Thilikos†

Abstract

We consider a bounded version of the restrictive and the restrictive list H-coloring problem in which the number of preimages of certain vertices of H is taken as parameter. We consider the decision and the counting versions, as well as, further variations of those problems. We provide complexity results identifying the cases when the problems are NP-complete or #P-complete or polynomial time solvable. We conclude stating some open problems.

1 Introduction

Suppose that we have a graph G whose vertices represent jobs to be processed and its edges represent communication demands between two jobs, assume that we also have a computer network represented as a graph H whose vertices represent the processors and its edges the communication links between them. We would like to assign the jobs to the processors so that all the communication demands between jobs are satisfied. This question can be modeled by the problem of asking whether there exists an homomorphism from G to H, the *H-coloring* problem [6]. Hell and Nešetřil proved a dichotomy theorem on the complexity of the H-coloring problem asserting that when H contains a loop or is bipartite the problem is in P, otherwise the problem is NP-complete [8].

An extension of the basic H-coloring problem is the problem of counting the number of H-colorings for a given a graph G, which is denoted as the *#H-coloring* problem. This problem has received a lot of attention in recent times, partially motivated by its connections with several models of statistical physics (see for example [5]). Dyer and Greenhill have given a complete characterization of the cases where the #H-coloring problem is #P-complete and the cases where the counting problem is polynomially solvable [5].

Let us assume that we impose some incompatibility restrictions between processes, in the sense that each process can be executed only in some specified processors. This problem variation is modeled by the *list H-coloring* problem, where each vertex in the input graph has associated a list with the vertices that can be mapped to. Feder, Hell, and Huang provided a dichotomy theorem for the complexity of the list H-coloring problem. They proved that if H belongs to the class of bi-arc graphs, the list H-coloring problem is in P, otherwise the problem is NP-complete [7].

*The work of the first author was partially supported by the Distinció per a la Recerca of the Generalitat de Catalunya, 2002. The work of the first and third authors was partially supported by the Spanish project TIN-2004-07925-C03-01. The work of the second and third authors was partially supported by the Spanish project TIC-2002-04498-C05-03.

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

The *list* $\#H$*-coloring* problem is the counting version of the list H-coloring problem. The dichotomy criterion for the $\#H$-coloring problem applies also for the list $\#H$-coloring problem [9].

Variants of the H-coloring problem in which some quantitative restrictions are fixed independently of the graph have been studied. Bačík considers the *equitable* H*-coloring* problem. An H-coloring is equitable if all the vertices in H have approximately the same number of preimages. The problem was also extended by pre-fixing the minimum proportion of vertices to be map into a given vertex. In the case that the graph H is *irreflexive*, without loops, it is shown that the equitable coloring problem can be solved in polynomial time when all the connected components of H are complete bipartite graphs, otherwise the problem is NP-complete [1]. We refer the reader to [10] for further variations on H-colorings.

In the network embedding problem, we can further restrict the number of jobs that may be assigned to a processor. This is a quite natural restriction for many practical instances. The general setting requires that together with the input graph it is provided a subset C of the vertices of H and a weight assignment K on C. Given C and K, a graph G has a restrictive H*-coloring* if there exists an H-coloring of G such that the number of preimages of any vertex in C equals its weight. The restrictive H*-coloring* problem asks for the existence of such a coloring, given G, C and K. In a similar way we can consider both bounds on the number of preimages together with preference lists, giving rise to the *restrictive list* H*-coloring* problem. The *restrictive* $\#H$*-coloring* and the *restrictive list* $\#H$*-coloring* are the counting version of the restrictive H-coloring problems. The authors proved a dichotomy for the restrictive H-coloring problems: If all the connected components of H are either a complete irreflexive bipartite graph or a complete reflexive clique, then the restrictive H-coloring and the restrictive list H-coloring problems can be solved in polynomial time, otherwise they are hard [2]. Observe that the dichotomy criterion for the $\#H$-coloring problem applies also for the restrictive versions, both counting and decision, with or without lists.

Recall that many NP-complete problems can be associated with one or more parameters. In many cases, when the parameter is considered to be a constant, the problem becomes polynomially solvable. However, this is not always the case. For instance, if in the 4-coloring problem we demand that one of the color classes has size at most a fixed constant k then the problem of finding such 4-colorings remains NP-complete. The restrictive H-coloring is amenable to parameterization by fixing a part of the input: C and K, and consider them as pre-fixed, and therefore bounded independently of the input size. We consider those bounded restrictive H-coloring problems denoted as the (H, C, K)*-coloring*, the *list* (H, C, K)*-coloring*, the $\#(H, C, K)$*-coloring*, and the *list* $\#(H, C, K)$*-coloring* problem, and initiate a complexity classification of them.

To study the complexity of the bounded versions of restrictive H-coloring problems, we address first the question of knowing which triples (H, C, K) lead to NP or #P-complete problems and which ones lead to polynomially solvable problems. We provide partial answers to the previous questions. We prove that the (H, C, K)-coloring problem remains NP-complete when the graph $H - C$, obtained by removing from H the vertices of C, is loopless and non-bipartite. On the other hand, we prove that, if $H - C$ is a bi-arc graph, the problem is in P. In the case of the list (H, C, K)-coloring problem, we prove that it is in P if $H - C$ is a bi-arc graph, otherwise the problem remains NP-complete. For the $\#(H, C, K)$-coloring problem, we give the necessary conditions to (H, C, K) for the problem to be #P-complete. We also provide further conditions on $H-C$ which classify the problem in P. In the case of the list $\#(H, C, K)$-coloring problem, we prove that the above conditions constitute a dichotomy theorem. The

Problem	P	NP-complete/#P-complete
list (H,C,K)-coloring	dichotomy (Theorem 3.1-3)	
(H,C,K)-coloring	list $(H-C)$-coloring in P	$(H-C)$-coloring NP-hard
list $\#(H,C,K)$-coloring	dichotomy (Theorem 3.1-1)	
$\#(H,C,K)$-coloring	list $\#(H-C)$-coloring in P	(H,C,K) irreducible
All results above hold also for the "$\leq$"-versions of the corresponding problems		

Table 1: Complexity of the parameterized H-coloring problems

results on the classification of the parameterized coloring problems are summarized in Table 1. Further results on the polynomially solvable cases having additionally a fixed parameter tractable algorithm can be found in [4].

Finally, we introduce a variant of the restrictive H-coloring, in which the number of preimages of each vertex in C is *at most* its weight. Working as before we consider the bounded version of this problem taking as parameter (C,K). We denote the so obtained problems using the term $(H,C,\leq K)$-coloring. We show that all the results in this paper also hold for this variant.

The paper is organized as follows: In Section 2 we give the basic definitions and some preliminary results. In Section 3 we prove the results of Table 1. Finally, in Section 4 we describe the consequence of our results to the other parameterization. We conclude in Section 5 stating some open problems.

2 Definitions and previous results

All the graphs in this paper are undirected, without multiedges, but can have loops. Following the terminology in [6, 7] we say that a graph is *reflexive* if all its vertices have a loop, a graph is *irreflexive* when none of its vertices have a loop, and a graph is *general* if it has some looped vertices and some unlooped ones. We denote a loop at vertex v as $\{v,v\}$. We assume that all the bipartite graphs in this paper are irreflexive. By K_n^r we denote the reflexive clique with n vertices and by C_n we denote the irreflexive cycle with n vertices.

Given a graph G, let $V(G)$ denote its vertex set and $E(G)$ denote its edge set. If $S\subseteq V(G)$, we call the graph $(S, E(G)\cap\{\{x,y\}\mid x,y\in S\})$ the *subgraph of G induced by S* and we denote it by $G[S]$. We denote by $\gamma(G)$ the number of connected components of G. As mentioned in the introduction we use $G-S$ as a notation for the graph $G[V(G)-S]$. Define the neighborhood of a vertex v in G by $N_G(v)=\{u\in V(G)|(u,v)\in E(G)\}$.

Given two graphs G and G', with no vertices in common, its *disjoint union* is the graph $G\cup G' = (V(G)\cup V(G'), E(G)\cup E(G'))$, and its *join* is the graph $G\oplus G' = (V(G)\cup V(G'), E(G)\cup E(G')\cup\{\{u,v\}\mid u\in V(G), v\in V(G')\}$.

For any function $\varphi: A\to B$ and any subset $C\subseteq A$, we define the *restriction of φ to C* by $\varphi|_C=\{(a,b)\in\varphi \mid a\in C\}$. Given two functions $\varphi: A\to\mathbb{N}$ and $\psi: A'\to\mathbb{N}$, with $A'\cap A=\emptyset$, the *disjoint union* of φ and ψ, is a function from $A\cup A'$ to $\mathbb{N}$ defined by:

$$(\varphi\cup\psi)(x)=\begin{cases}\varphi(x) & \text{if } x\in A,\\ \psi(x) & \text{if } x\in A'.\end{cases}$$

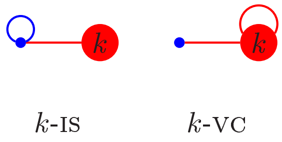

Figure 1: The partial weight assignments (H, C, K) for the parameterized independent set and vertex cover as parameterized restrictive colorings (the labeled vertices represent the set C and its label the associated weight)

In case that $A = A'$, for any $x \in A$, we define $(\varphi + \psi)(x) = \varphi(x) + \psi(x)$. Given two functions $\varphi, \psi : A \to \mathbb{N}$ we say that $\phi \leq \psi$ if, for any $x \in A$, $\phi(x) \leq \psi(x)$. We use $\varnothing$ to denote the empty function.

Given two graphs G and H we say that a function $\chi : V(G) \to V(H)$ is an *H-coloring* of G if for any edge $\{x, y\}$ of G, $\{\chi(x), \chi(y)\}$ is also an edge of H.

A partial weight assignment on H is a pair (C, K) where $C \subseteq V(H)$ and $K : C \to \mathbb{N}$. A *restrictive H-coloring* of G and (C, K) is a partial weight assignment on H is an H-coloring of G such that the number of vertices of G mapped to $c \in C$ is $K(c)$.

The triple (H, C, K), where H is a graph and (C, K) is partial weight assignment on H, is called a *partial weight assignment*. All the vertices of C are the *weighted vertices* of H. For each partial weight assignment (H, C, K), we use the notation $h = |V(H)|, r = |C|$ and $k = \sum_{c \in C} K(c)$. We say that a partial weight assignment (H, C, K) is *positive* if, for any $c \in C$, $K(c) \neq 0$. A partial weight assignment (H, C, K) is said to be a *weighted extension* of a graph F if $H - C = F$. Finally, for any $S \subseteq C$, we define $(H, C, K)[S] = (H - (C - S), S, K|_S)$.

Let (H, C, K) be a partial weight assignment and let c be a vertex in C. We call (H, C, K) *c-reducible* if H has an $(H - \{c\})$-coloring χ such that $\chi(c) \in V(H) - C$. We say that (H, C, K) is *reducible* if it is c-reducible for some $c \in C$, otherwise (H, C, K) is said to be *irreducible*.

Given a partial weight assignment (H, C, K), a mapping $\chi : V(G) \to V(H)$ is a *(H, C, K)-coloring* of G if χ is a restrictive H-coloring of G and (C, K). Notice in this case, for all $c \in C$, we have $|\chi^{-1}(c)| = K(c)$. If in the above definition we replace "$=$" with "$\leq$", we say that χ is a *$(H, C, \leq K)$-coloring* of G.

For fixed H, C, and K, given an input graph G, the *(H, C, K)-coloring* problem checks whether there exists an (H, C, K)-coloring of G. In the same way, the *$(H, C, \leq K)$-coloring* problem checks whether G has an $(H, C, \leq K)$-coloring.

In Figure 1 we give two examples where the (H, C, K)-coloring problem corresponds to the parameterized independent set and the parameterized vertex cover problem.

Two partial weight assignments (H, C, K) and (H', C', K') are said to be *equivalent*, and we denote it as $(H, C, K) \sim (H', C', K')$ if, for any graph G, G has an (H, C, K)-coloring if and only if G has an (H', C', K')-coloring. In a similar way, two partial weight assignments are said *$\leq$-equivalent* and we denote it as $(H, C, K) \sim_{\leq} (H', C', K')$, if, for any graph G, G has an $(H, C, \leq K)$-coloring if and only if G has an $(H', C', \leq K')$-coloring.

For a fixed graph H and a given a graph G, an *(H, G)-list* is a function $L : V(G) \to 2^{V(H)}$ mapping any vertex of G to a subset of $V(H)$. For any vertex $v \in V(G)$, the *list* of v is the set $L(v)$. The *list H-coloring* problem asks whether, given a graph G and an (H, G)-list L, there is an H-coloring χ of G so that for every $u \in V(G)$ we have $\chi(u) \in L(u)$, we will refer to such χ as a list H-coloring of (G, L). The *restrictive list H-coloring* problem asks whether, given a graph G, an (H,G)-list L, and a partial weight assignment on H (C,K) there is an H-coloring χ of G so that for every $u \in V(G)$ we have $\chi(u) \in L(u)$, and further, for any $c \in C$ $|\chi^{-1}(c)| = K(c)$. This problem can be parameterized in the same way as the restrictive H-coloring to define the *list* (H,C,K)*-coloring* and the *list* $(H,C,\leq K)$*-coloring* problems.

The bi-arc graphs are useful for characterizing the complexity of the list H-coloring problem and they are defined using the following geometric representation.

Let C be a cycle and two specific points p and q on it. A *bi-arc* is an ordered pair of arcs (N,S) on C such that N contains p but not q, and S contains q but not p. A graph is a *bi-arc graph* if there is a family of bi-arcs $\{(N_x,S_x) \ : \ x \in V(H)\}$ such that, for any $x, y \in V(H)$, not necessary distinct, the following hold:

- if x and y are adjacent, then neither N_x intersects S_y nor N_y intersects S_x;
- if x and y are not adjacent, then both N_x intersects S_y and N_y intersects S_x.

As usual, the $\#(H,C,K)$-coloring, the list $\#(H,C,K)$-coloring, the $\#(H,C,\leq K)$-coloring and the the list $\#(H,C,\leq K)$-coloring problems denote the counting versions of the corresponding decision problems. The complexity of the H-coloring, the list H-coloring, and the restrictive H-coloring problems and their corresponding counting versions is described by the following theorem.

Theorem 2.1.

1. [8] The H-coloring problem is in P if H contains a loop or it is bipartite; otherwise it is NP-complete.
2. [7] The list H-coloring problems is in P if H is a bi-arc graph; otherwise it is NP-complete.
3. [5] The $\#H$-coloring problem is in P if all the connected components of H are either complete reflexive graphs or complete irreflexive bipartite graphs. Otherwise, it is $\#\mathsf{P}$-complete.
4. [9] The list $\#H$-coloring problem is in P if all the connected components of H are either complete reflexive graphs or complete irreflexive bipartite graphs. Otherwise the counting problem is $\#\mathsf{P}$-complete. The same result holds even if G is a graph of degree bounded by a constant [9].
5. [2] The restrictive H-coloring, the restrictive list H-coloring, the restrictive $\#H$-coloring and the restrictive list $\#H$-coloring problems are in P if all the connected components of H are either complete reflexive graphs or complete irreflexive bipartite graphs. Otherwise, the decision versions are NP-complete and the counting version $\#\mathsf{P}$-complete.

3 Complexity issues

In this section we provide a classification, for each of the introduced coloring problems, separating weight assignments (H,C,K) for which the corresponding problem belongs to P from those for which the problem is hard (NP-complete or $\#\mathsf{P}$-complete). For a summary of the results mentioned in this section, see Table 1.

We first state the theorem,

$$
\begin{tikzcd}
(H, C, \leq K) \arrow[r, "(1)"] \arrow[d, "(7)"] & \#(H, C, \leq K) \arrow[d, shift left, "(8)"] \\
(H, C, K) \arrow[r, "(2)"] \arrow[d, "(5)"] & \#(H, C, K) \arrow[u, shift left, "(13)"] \arrow[d, "(6)"] \\
\text{list } (H, C, K) \arrow[r, "(3)"] \arrow[d, shift left, "(11)"] & \text{list } \#(H, C, K) \arrow[d, shift left, "(12)"] \\
\text{list } (H, C, \leq K) \arrow[r, "(4)"] \arrow[u, shift left, "(10)"] & \text{list } \#(H, C, \leq K) \arrow[u, shift left, "(9)"]
\end{tikzcd}
$$

Figure 2: Reductions between problems. If Π_1 and Π_2 are problems, the notation $\Pi_1 \longrightarrow \Pi_2$ means that Π_2 can be solved using an algorithm for Π_1 as subroutine.

Theorem 3.1.

1. The list $\#(H, C, K)$-coloring problem is in P if all the connected components of $H - C$ are either reflexive cliques or complete irreflexive bipartite graphs; otherwise it is #P-complete.

2. The $\#(H, C, K)$-coloring problem is in P if all the connected components of $H - C$ are either complete reflexive graphs or complete irreflexive bipartite graphs. If this condition is not satisfied and (H, C, K) is irreducible, then the $\#(H, C, K)$-coloring problem is #P-complete.

3. The list (H, C, K)-coloring problem is in P if $H - C$ is a bi-arc graph; otherwise it is NP-complete.

4. The (H, C, K)-coloring problem is in P if $H - C$ is a bi-arc graph. The problem is NP-complete if $H - C$ is not bipartite and does not contain a loop.

Before actually proving the theorem, we need to build up some basic structure. We start by showing the reducibility relationship between the problems under consideration. The reductions are summarized in Figure 2. We are going to sketch the proof of the correctness for the reductions.

Reductions (1) to (4): Any algorithm for a counting problem can also be used for its decision version.

Reductions (5) and (6): Any H-coloring problem instance can be seen as a list H-coloring instance in which the list for each vertex is $V(H)$.

Reductions (7) to (10): The $(H, C, \leq K)$-coloring problem can be reduced to the computation of $\leq k^r$ instances of the (H, C, K')-coloring problem, where $K' \leq K$.

Reduction (11): Let (G, L) be an instance of the list (H, C, K)-coloring problem. For each $D \subseteq V(G)$ with k vertices and each mapping $\chi : D \to C$, where $\forall_{v \in D}\ \chi(v) \in L(v)$ and $\forall_{c \in C}\ |\chi^{-1}(c)| = K(c)$, we construct an instance of the list $(H, C, \leq K)$-coloring problem where the input is the graph G and the list L' defined by

$$L'(v) = \begin{cases} L(v) & \text{if } v \notin D \\ \chi(v) & \text{otherwise.} \end{cases}$$

Then, G has an (H, C, K)-coloring problem if there is an $(H, C, \leq K)$-coloring for some constructed instance for the list $(H, C, \leq K)$-coloring problem. As the number of instances for the list $(H, C, \leq K)$-coloring problem is at most $k!|V(G)|^k$, the correctness of the reduction follows.

Reductions (12) and (13): The number of (H, C, K)-colorings is the difference of the number of $(H, C, \leq K)$-colorings minus the number of $(H, C, \leq K^-)$-colorings, where K^- is obtained by selecting arbitrarily a non-zero component of K, and decreasing it by one. The same arguments hold for the "listed" versions of the corresponding problems.

It is interesting to notice that the reduction landscape in Figure 2 can not be extended, other than by transitivity, with further arrows, without implying either $\mathsf{P} = \mathsf{NP}$ or $\mathsf{P} = \#\mathsf{P}$. The irreversibility of arrows (1) to (4) follows from the fact that the $(H, \emptyset, \varnothing)$-coloring problem is equivalent to the H-coloring problem and that from Theorem 2.1.1 we can find for each case an H where the decision version is in P and the corresponding counting version is $\#\mathsf{P}$-complete. Using the same idea and Theorem 2.1.1 one can find suitable H's justifying the irreversibility of arrows (5) and (6), under the hypothesis $\mathsf{P} \neq \mathsf{NP}$. The most interesting case, and rather asymmetric, is the irreversibility of arrow (7). To justify this case, it is enough to verify that if $H - C$ contains a looped vertex then an $(H, C, \leq K)$-coloring always exist, so the $(H, C, \leq K)$-coloring problem belongs to P. On the other hand, as we will prove in Theorem 3.4 of Section 3.1, there are triples (H, C, K) where $H - C$ contains a looped vertex and such that the $(H, C, \leq K)$-coloring problem is NP-complete. Therefore, unless $\mathsf{P} = \mathsf{NP}$, the (H, C, K)-coloring problem is not reducible to the $(H, C, \leq K)$-coloring problem.

We present some positive results for the list $(H - C)$-coloring and its counting version. The following straightforward Lemma gives the conditions under which a partial list coloring can be extended to a total list coloring.

Lemma 3.1. Let G, H be graphs and L an (H, G)-list. Let $D \subseteq V(G)$ and $\chi^* : D \to V(H)$ be a list $(H[C], C, K)$-coloring of $(G[D], L|_D)$. Finally let

$$L'(v) = \begin{cases} L(v) \cap \left(\bigcap_{u \in N_G(v) \cap D} (N_H(\chi(u)) - C) \right) & \text{if } N_G(v) \cap D \neq \emptyset, \\ L(v) & \text{otherwise.} \end{cases}$$

Then the following hold:

- If χ' is a list $(H - C)$-coloring of $(G - D, L')$ then $\chi = \chi' \cup \chi^*$ is an (H, C, K)-coloring of G.
- If χ is an (H, C, K)-coloring of G where $\chi \supseteq \chi^*$ then $\chi - \chi^*$ is an $(H - C)$-coloring of $(G - D, L')$.

Let us begin by giving a sufficient condition for the $\#(H, C, K)$-coloring problem to be in P.

Lemma 3.2. The list $\#(H, C, K)$-coloring problem can be solved in n^{k+c} steps, whenever the list $\#(H - C)$-coloring can be solved in $O(n^c)$ steps.

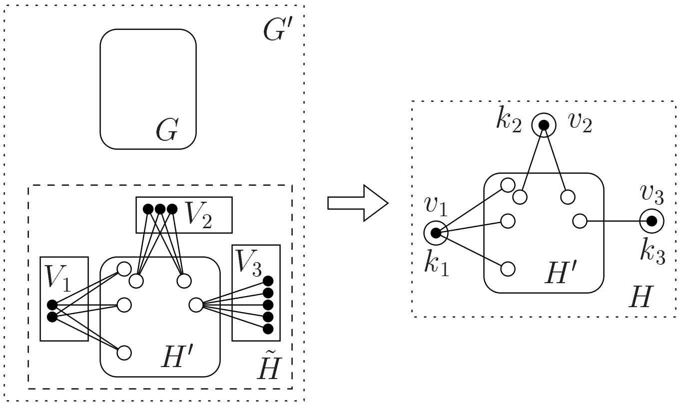

Figure 3: The reduction in the proof of Theorems 3.5 and 3.8

Proof. Let $\mathcal{X}$ be the set of all list $(H[C], C, K)$-colorings of each of the subgraphs of G with at most k vertices. If there is an algorithm solving the list $\#(H-C)$-coloring in $O(n^c)$ steps we can use this algorithm for each $(H[C], C, K)$-coloring $\chi^* \in \mathcal{X}$ and, according to the previous lemma, compute the number of the $(H-C)$-colorings χ' of $(G-D, L')$ that form an extension of χ^* to a list (H, C, K)-coloring of G. If we sum up these numbers we have the number of list (H, C, K)-colorings of G. As $|\mathcal{X}| \leq n^k$, the whole procedure needs $O(n^{c+k})$ steps. □

The same technique can be used to extend the previous result to the list (H, C, K)-coloring problem.

Lemma 3.3. The list (H, C, K)-coloring problem can be solved in n^{k+c} steps, whenever the list $(H-C)$-coloring can be solved in $O(n^c)$ steps.

Let us turn to proving conditions for which the problems are hard. We first prove that if $H-C$ is not bipartite and does not contain a loop then the $(H, C, \leq K)$-coloring problem is NP-complete. As we will see later, it suffices to consider the cases for which (H, C, K) is positive, as these cases imply the general case. Notice, that all the problems considered here belong to the class NP, therefore, we will only prove the hardness.

The following lemma indicates that for any reducible partial weight assignment there exists an equivalent irreducible partial weight assignment.

Lemma 3.4. If (H, C, K) is c-reducible then $(H, C, \leq K) \sim_{\leq} (H, C, \leq K)[V(H) - \{c\}]$.

Proof. Let $H' = H - \{c\}$. As c is a reducible vertex, there exists some H'-coloring φ of H such that $\varphi(c) = a \in V(H) - C$.

Given a graph G, let χ be an $(H, C, \leq K)$-coloring of G, then $\chi' = \rho \circ \chi$ is an H'-coloring of G. Moreover, by construction of χ' together with the fact that $\chi'(c) \notin C$, we have that for any $b \in C - \{c\}$, $|\chi'^{-1}(b)| \leq K(b)$. Therefore χ' is an $(H, C, \leq K)[V(H) - \{c\}]$-coloring of G.

Suppose now that χ is an $(H, C, \leq K)[V(H) - \{c\}]$-coloring of G. As $H' = H - \{c\}$, then χ is also an $(H, C, \leq K)$-coloring of G, and the lemma follows. □

Therefore, we can remove all the reducible weighted vertices while maintaining the problem equivalence.

Lemma 3.5. For any partial weight assignment (H, C, K), where the $(H-C)$-coloring problem is NP-complete, the $(H, C, \leq K)$-coloring problem is also NP-complete.

Proof. From the previous lemma, we can assume that (H, C, K) is an irreducible partial weight assignment. We set $H' = H - C$ and reduce the H'-coloring problem to the $(H, C, \leq K)$-coloring problem.

Define the graph $\tilde{H}$ constructed from H in such a way that each vertex $c \in C$ is replaced by a set of $K(c)$ vertices $V_c = \{v_1^c, \dots, v_{K(c)}^c\}$. Any edge connecting two vertices in $V(H) - C$ is maintained, an edge connecting $u \in V(H) - C$ with $c \in C$ is replaced by a set of edges $\{(u, v_j^c) \mid 1 \leq j \leq K(c)\}$, an edge connecting c and c' in C is replaced by $\{(v_j^c, v_{j'}^{c'}) \mid 1 \leq j \leq K(c), 1 \leq j' \leq K(c')\}$. (See Figure 3.)

Notice that by projecting all the copies to their pre-image, $\tilde{H}$ has an $(H, C, \leq K)$-coloring. Moreover, for any $(H, C, \leq K)$-coloring ρ of $\tilde{H}$ and for any $c \in C$ we have $|\rho^{-1}(c)| = K(c)$.

Given a graph G, consider the graph $G' = G \cap \tilde{H}$. We claim that G has an H'-coloring iff G' has an $(H, C, \leq K)$-coloring. If G has an H'-coloring, then the union of such an H'-coloring with an $(H, C, \leq K)$-coloring of $\tilde{H}$ gives an $(H, C, \leq K)$-coloring of G'. On the other direction, let χ be an $(H, C, \leq K)$-coloring of G'. As $\chi|_{V(\tilde{H})}$ is an $(H, C, \leq K)$-coloring of H', all the vertices of G must be mapped to $V(H')$, but then $\chi|_{V(G)}$ is an H' coloring of G. □

The "list" version of the above lemma is much easier, as for a given graph G and an $(H - C, G)$-list L, G has a list $(H - C)$-coloring iff G has a list $(H, C, \leq K)$-coloring with the same $(H - C, G)$-list. Therefore,

Lemma 3.6. For any partial weight assignment (H, C, K), where the list $(H - C)$-coloring problem is NP-complete, the $(H, C, \leq K)$-coloring problem is also NP-complete.

Let us prove the hardness results for the for the counting versions of the problem.

Lemma 3.7. For any partial weight assignment (H, C, K), the list $\#(H, C, K)$-coloring problem is #P-complete, whenever the $\#(H - C)$-coloring problem is #P-complete.

Proof. We set $H' = H - C$ and, in what follows, we reduce the $\#H'$-coloring problem to the list $\#(H, C, K)$-coloring problem. Let G be an instance of the H'-coloring problem. We construct an instance G' of the (H, C, K)-coloring problem as follows: Take G' as the disjoint union of G and a collection F_c of graphs, for $c \in C$, such that $F_c = (\{f_1^1, \dots, f_{K(c)}^1\}, \emptyset)$. Associated to every $v \in V(G)$, the list $L(v) = V(H')$. For any $c \in C$ and any $v \in V(F_c)$, make $L(v) = \{c\}$. Then, the H'-colorings of G are in one to one correspondence with the list H-coloring of (G', L), which give us the reduction between the $\#H'$-coloring problem and the list $\#(H, C, K)$-coloring problem. □

We can strengthened the hypothesis in the previous result as stated in the following lemma.

Lemma 3.8. For any irreducible parameter assignment (H, C, K), the $\#(H, C, K)$-coloring problem is #P-complete whenever the $\#(H - C)$-coloring problem is #P-complete.

Proof. Let (H, C, K) be an irreducible parameter assignment and set $H' = H - C$. We reduce the $\#H'$-coloring problem to the $\#(H, C, K)$-coloring problem:

Define the graph $\tilde{H}$ defined in the proof of Lemma 3.5 (see Figure 3). Recall that $\tilde{H}$ has an (H, C, K)-coloring, where $\chi(v) = v$ for $v \in V(\tilde{H}) - C'$ and $\chi(v) = c$ for $v \in V_c$. Furthermore, notice that a vertex in any V_c can only be mapped to C, therefore (H, C, K) is irreducible.

Let G be a graph, and let G' be the disjoint union of G and $\tilde{H}$. We claim that G has an H'-coloring iff G' has an (H, C, K)-coloring.

If G has an H'-coloring, then G' has also an (H,C,K)-coloring, because $\tilde{H}$ has an (H,C,K)-coloring. On the contrary, if G' is (H,C,K)-colorable then, all the vertices of G must be mapped to vertices of H', and G has an H'-coloring.

Any (H,C,K)-coloring of G' is formed by an (H,C,K)-coloring of $\tilde{H}$ and an H' coloring of G. Furthermore the disjointness of G and $\tilde{H}$ guarantees that the number of (H,C,K) coloring of G' is the product of the number of H' coloring of G and the number of (H,C,K)-coloring of $\tilde{H}$. This quantity can be computed in constant time and the statement of the Lemma follows. □

Now we are ready to finish the proof of Theorem 3.1.

. **Proof of Theorem 3.1**. During the proof we refer to the reductions summarized in Figure 2. The positive part of 3.1(1) follows from Lemma 3.2 and the positive part of Theorem 2.1(4). The positive part of Theorem 3.1(2) is a consequence of the positive part of Theorem 3.1.(1) and reduction (6). The negative part is a direct consequence of Lemma 3.8 and Theorem 2.1(3).

The positive part of 3.1(3) follows from Lemma 3.3 and the positive part of Theorem 2.1(2). The negative part follows from reduction (9) and 3.6 together with the negative part of Theorem 2.1(2). □

The positive part of 3.1.4 is derived by the positive part of Theorem 2.1(3) and reduction (5). Using Lemma 3.5 and reduction (7) along with Theorem 2.1(1) we have the negative part. That concludes the proof of Theorem 3.1.

Notice, finally, that Theorem 3.1 hold also for the "$\leq$"-versions of the problems. The parts 1, 2 and 3 come directly from the double reductions between the problems (see Figure 2). For the '$\leq$"-version of Theorem 3.1(4), the negative part follows directly from Lemma 3.5 and Theorem 2.1(1), while the positive part is derived using reduction (7).

3.1 The (H,C,K)-coloring gap

In this section we prove that the hypothesis of the positive part in Theorem 3.1(4) is not a necessary condition. In particular we show that the (H,C,K)-coloring remains NP-compete for certain weighted extensions of graphs F that are either bipartite or that contain at least one looped vertex. Observe that in the case where $H-C$ contains a looped vertex, the $(H,C,\leq K)$-coloring is trivially in P.

The list H-coloring problem on bipartite graphs, which that contain a chordless cycle of length $2k-2$ is NP-complete, for $k>3$. The proof is given in [7] and it consists of the following reduction from the k-coloring problem: given a graph G construct a new graph G' together with an (H,G') list L in the following way. The graph G' is obtained from G by replacing each edge by the gadget given in Figure 4, making the cycle C_0 common to all the edges. The list L fixes a unique possible image of the vertices in C_0 in the chordless cycle of H. The list for all the he remaining vertices is $V(H)$. The previous reduction proves the following theorem.

Theorem 3.2 ([7]). Let H be a bipartite graph that contain a chordless cycle of length $2k-2$, for some $k\geq 3$. Then, given a graph G, G has a k-coloring if (G',L) has a list H-coloring.

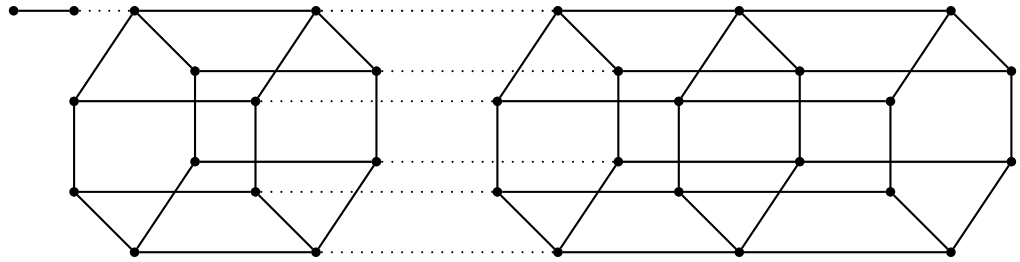

Figure 4: The gadget from [7] replacing an edge (x, y), C_0 is common to all the edges

We can achieve an analogous result for (H, C, K)-coloring, replacing the additional list by an suitable parametrized extension (H', C, K) of H. To explain this idea, we introduce the concept of a *rigid* extension of a cycle. Given a cycle C_α, we define a graph $\tilde{C}_\alpha$ by adding a set E_α with α additional weighted vertices, each new vertex corresponds to an edge in C_α and it is connected to the end-points of the associated edge (see Figure 5). Rigid extensions of reflexive or irreflexive cycles verify the following,

Lemma 3.9. Let C_α be a cycle with $\alpha \geq 4$ vertices, let $\tilde{C}_\alpha$ be its rigid extension and let 1_{E_α} the function constant 1 on E_α. Then any $(\tilde{C}_\alpha, E\alpha, 1_{E_\alpha})$-coloring of $\tilde{C}_k$, preserves the cyclical ordering of the vertices in C_k.

Proof. Observe that the triangles must be mapped to triangles, but each triangle in C_α can get at most one triangle. □

Let H be a bipartite graph that contain a chordless cycle of length $k \geq 4$. We reduce the k-coloring problem to the (H', C, K)-coloring problem, where H' is obtained from H making a rigid extension of the chordless cycle, setting C to the vertices added and making $K = 1_C$. The reduction is a modification of the construction in [7]: replace C_0 by its rigid extension. The graph G' is obtained from G by replacing each edge by the gadget given in Figure 6. Finally make $\tilde{C}_0$ common to all the edges. From Lemma 3.9 we have that the vertices in $\tilde{C}_0$ must be mapped to the unique triangles in H', and From Lemma 3.9 they must preserve the cyclic ordering. Therefore, using Theorem 3.2 we have proved the following theorem.

Theorem 3.3. If F is a bipartite graph that contains a chordless cycle of length greater than four, then F has a weighted extension that is NP-complete.

We can work in the same way with the construction given as reduction in [6] for the case that H is a bipartite reflexive graph that contains a chordless cycle with length 4 or more. Therefore, we have

Theorem 3.4. If F is a reflexive bipartite graph that contains a chordless cycle of length greater than four, then F has a weighted extension that is NP-complete.

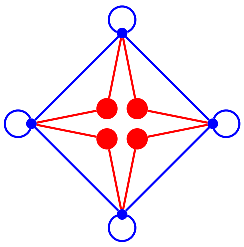

Figure 5: The rigid cycle extensions for irreflexive and reflexive cycles

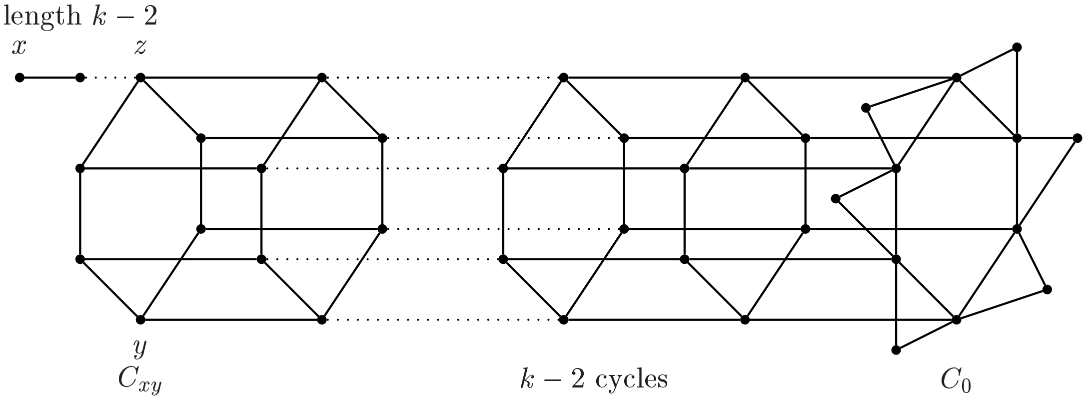

Figure 6: The gadget replacing an edge (x, y), C_0 is common to all the edges

In contraposition with the above results we can show tractability of a weighted extension of any graph F for which the list F-coloring is still NP-complete, but the F-coloring is in P.

Theorem 3.5. Let F_1 be either a bipartite graph with or without loops or a graph with at least one loop, and let F_2 be any graph. Then, for any K, the $(F_1 \oplus F_2, V(F_2), K)$-coloring can be solved in polynomial time.

Proof. As we have all the edges between $V(F_1)$ and $V(F_2)$, the lists associated to each vertex are just all the vertices in F_1. Therefore, we have to solve a F_1-coloring problem, which can be solved in polynomial time. $\square$

Finally we mention that any weighted extension of a graph F for which the F-coloring is in P, is also polynomially solvable in the trivial case when $K = (0, \ldots, 0)$.

In conclusion, we observe that to obtain, if exists, a dichotomy for the (H, C, K)-coloring problem is a hard problem as the corresponding characterization cannot only depend on $H - C$.

4 Extensions and further results

In this section we examine the complexity of two variants of the list coloring problem where some restrictions on the lists are imposed. We also extend some of our results to bounded degree graphs.

4.1 Variants of list-$\#H$-coloring.

We consider two variants of the list H-coloring: the *OA-list H-coloring*, where the list are either $V(H)$ or a single vertex; and the *C-list H-coloring*, where for every vertex $v \in V(G)$ the graph $H[L(v)]$ is connected. Both variants were introduced in [6]. The *OA-list (H,C,K)-coloring* and the *C-list (H,C,K)-coloring* are defined analogously. In this subsection we will study the complexity of the counting versions of all the variants defined before, namely the *OA-list $\#H$-coloring*, the *C-list $\#H$-coloring*, the *OA-list $\#(H,C,K)$-coloring*, and the *C-list $\#(H,C,K)$-coloring*.

We will first prove that Theorem 4 can be rewritten for both *OA-list $\#H$-coloring* and *C-list $\#H$-coloring*. In all the proofs of this section we use the fact that the $\#H$-coloring remains #P when the input graph is connected, which is a correct assertion as the number of H-colorings of G is equal to the product of the H-colorings of its connected components.

Lemma 4.1. If the OA-list $\#H$-coloring is in P, then the $\#H$-coloring problem is in P.

Proof. Let G be a graph, take a vertex $u \in V(G)$. The number of H-colorings of G can be computed as the sum of the solutions of h instances of OA-list $\#H$-coloring. An instance is generated by selecting a vertex $x \in H$ and letting $L(u) = \{x\}$ and, for all $v \in V(G)$, $u \neq v$, $L(v) = V(H)$. □

Lemma 4.2. If the C-list $\#H$-coloring is in P then the $\#H$-coloring problem is in P.

Proof. Let G be a connected graph. The number of H-colorings of G can be computed as the sum of the solutions of $\gamma(H)$ instances of the C-list $\#H$-coloring problem. An instance is generated by selecting one connected component F of H and letting $L(u) = F$ for all $u \in V(G)$. □

The positive part holds for both variants, so we have the following theorem.

Theorem 4.1. The C-list $\#H$-coloring and the OA-list $\#H$-coloring problems are in P if all the connected components of H are either complete reflexive graph or a complete irreflexive bipartite graphs. Otherwise the counting problems are #P-complete.

Notice that the reduction in the proof of Lemma 3.7 generates an instance in which the list for each vertex v of G', $L(v)$ is either $V(H)$ or a single vertex Therefore, we have the following:

Lemma 4.3. If the OA-list $\#(H,C,K)$-coloring is in P, then the $\#(H-C)$-coloring problem is in P.

The reduction provided in Lemma 3.7 does not generate a connected list instance. However, we can apply the same reduction for a connected graph G and create α copies of the pair (G', L), $(G'_1, L_1), \ldots, (G'_\alpha, L_\alpha)$, where α is the number of the connected components

$C_1, \ldots, C_\alpha$ of $H' = H - C$ that are not complete reflexive graphs neither complete irreflexive bipartite graphs. For each copy G'_i, L_i we modify L_i only for the vertices of G and we set $L_i(v) = V(C)$. As all the remaining components of $H - C$ are either complete reflexive graphs or complete irreflexive bipartite graphs, it is possible to compute the number n^* of C^*-colorings of G where C^* is their disjoint union. Notice also that by the construction of (G', L), the number n_i of C-list $\#(H, C, K)$-coloring problem of (G'_i, L_i) is equal to the number of the C_i-colorings of G. If there exists a polynomial algorithm for the C-list $\#(H, C, K)$-coloring problem then it is possible to compute the number $n_1 + n_2 + \cdots + n_\alpha + n^*$ that is the number of H'-colorings of G. Therefore we proved the following lemma.

Lemma 4.4. If the C-list $\#(H, C, K)$-coloring is in P, then the $\#(H - C)$-coloring problem is in P.

As the positive part holds for both variants, Theorem 1 can be rewritten as follows.

Theorem 4.2. The C-list $\#(H, C, K)$-coloring and the OA-list $\#(H, C, K)$-coloring problems are in P if all the connected components of H are either complete reflexive graph or a complete irreflexive bipartite graphs. Otherwise the counting problems are #P-complete.

Lemmata 4.1, 4.2, 4.3, and 4.4 can be rewritten for the decision versions of the corresponding problems. The proofs are almost the same. The only difference is that the "sums" should be replaced by binary "OR"-operations. Therefore, we have the following analogous of Theorems 4.1 and 4.2.

Theorem 4.3. The C-list H-coloring and the OA-list H-coloring problems are in P if H is a bi-arc graph. Otherwise the counting problems are NP-complete.

Theorem 4.4. The C-list (H, C, K)-coloring and the OA-list (H, C, K)-coloring problems are in P if $H - C$ is a bi-arc graph. Otherwise the counting problems are NP-complete.

The last question on the problems examined in this subsection is whether Theorems 4.2 and 4.4 hold also for the "$\leq$"-variants of the corresponding problems. The answers is positive as reductions (8) and (10) (see Figure 2) hold also for the "C-list" and "OA-list" versions of the list (H, C, K)-coloring and the list $\#(H, C, K)$-coloring (Notice that in the proof of reduction (10) the new lists preserve both "one-or-all" and "connectivity" requirements).

4.2 Bounded degree graphs

Recall that Theorems 2.1(3) and 2.1(4) hold even if the input graphs have bounded degree. Notice that the reduction used in the proof of Theorem 3.7, constructs a graph G' with the same max degree as G. This implies that Theorem 3.1(1) and its consequence Theorem 4.2 hold also for input graphs of bounded degree.

5 Remarks and open problems

Notice that, according to Theorem 3.1(4), for any weighted extension (H, C, K) of a bi-arc graph F the (H, C, K)-coloring problem is in P. On the other hand, if F is not bipartite and does not contain a loop then for any weighted extension (H, C, K) the (H, C, K)-coloring problem is NP-complete.

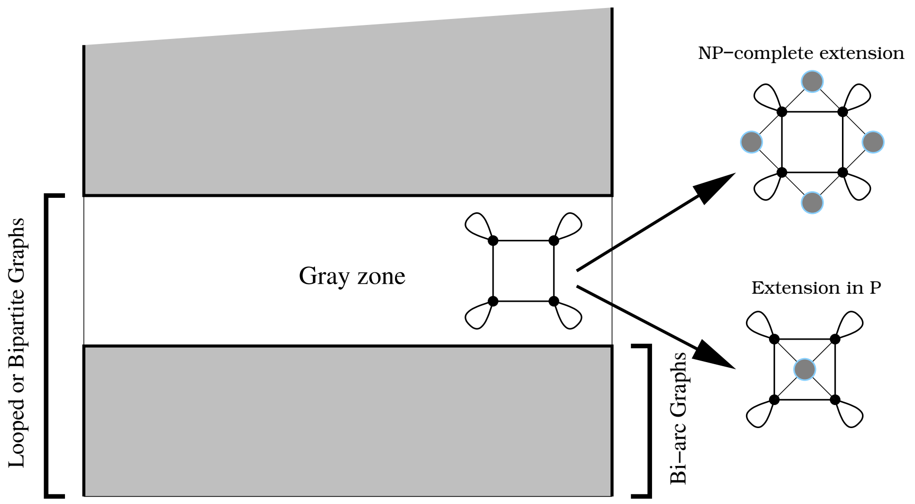

Figure 7: The landscape of the complexity of (H, C, K)-coloring

Certainly, Theorem 3.1(4) does not provide a dichotomy on the complexity of the (H, C, K)-coloring problem. The "grey zone" of this theorem deals with graphs that are not bi-arc but that either are bipartite or contain a loop (see Figure 7). If a graph is loopless and non-bipartite we say that it is "above" the grey zone. If it is bi-arc then it is below. Theorem 3.1(4) says that if F is a graph outside from this grey zone and (H, C, K) is any weighted extension of it, then the complexity of the (H, C, K)-coloring problem depends only on whether F is above (NP-complete cases) or below (cases in P) the grey zone. It is interesting to see that things are quite different inside the grey zone: using Theorems 3.3, 3.4, and 3.5, we can construct graphs which for different weighted extensions produce parameterized coloring problems with different complexities (i.e. a non reflexive cycle of length 6 and a reflexive cycle of length 4 (see Figure 5). This prompts us to conjecture the following analogous of Theorem 3.5,

Conjecture 1. For any graph F that is not a bi-arc graph there is a weighted extension (H, C, K) of F such that the (H, C, K)-coloring is NP-complete.

The above observations indicate that "closing" the gap of Theorem 3.1(4) is not an easy problem as it appears that the "frontier", if it exists, depends not only on $H - C$ but also on the structure imposed by the weighted vertices. However, we observe that in all our complexity results, positive or negative, this dichotomy does not depend on the choice of the numbers in K when K is positive.

References

[1] R. Bačík. *Structure of graph homomorphism.* PhD thesis, School of Computer Science, Simon Fraser University, 1997.

[2] J. Díaz, M. Serna, and D. Thilikos. The restrictive H-coloring problem. *Discrete Applied Mathematics*, 2004. To appear.

[3] J. Díaz, M. Serna, and D. M. Thilikos. Recent results on Parameterized H-coloring. In *Graph, morphisms and statistical physics*, J. Nešetřil and P. Winkler, editors, *DIMACS*

Series in Discrete Mathematics and Theoretical Computer Science, Vol. 63, pp. 65–85, 2004.

[4] J. Díaz, M. Serna, and D. M. Thilikos. Fixed parameter algorithms for counting and deciding bounded restrictive H-colorings. In *Algorithms ESA 2004*, S. Albers and T. Radzik, editors, *Lecture Notes in Computer Science*, Vol. 3221, pp. 275–286, 2004.

[5] M. Dyer and C. Greenhill. The complexity of counting graph homomorphisms (extended abstract). *Random Structures and Algorithms*, 17:260–289, 2000.

[6] T. Feder and P. Hell. List homomorphisms to reflexive graphs. *J. Combin. Theory Ser. B*, 72(2):236–250, 1998.

[7] T. Feder, P. Hell, and J. Huang. List homomorphisms and circular arc graphs. *Combinatorica*, 19(4):487–505, 1999.

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

[9] P. Hell and J. Nešetřil. Counting list homomorphisms and graphs with bounded degrees. In In *Graph, morphisms and statistical physics*, J. Nešetřil and P. Winkler, editors, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 63, pp. 105–112, 2004.

[10] P. Hell and J. Nešetřil. *Graphs and homomorphisms.* Oxford University Press, 2004.