Counting List H-Colorings and Variants*

Josep Díaz† Maria Serna† Dimitrios M. Thilikos†

Abstract

We study the counting versions of several variants of the H-coloring problem. The complexity of the $\#H$-coloring problem is described by a dichotomy theorem that classifies the problem according to the the topology of H as #P-complete or polynomial time solvable [6]. We first prove that the same dichotomy criterion holds also for its "list" extension, the list $\#H$-coloring problem. We further investigate these problems by examining a parameterization imposing restrictions on the number of preimages of certain vertices in H. We prove a nearly dichotomy theorem on the complexity of counting such restricted homomorphisms, as well as a dichotomy theorem on the complexity of its "list" version. Finally, we give a condition that makes the parameterizedd problems solvable in linear time.

1 Introduction

Suppose that we have a graph whose vertices represent jobs to be processed and whose edges represent communication demands between two jobs. Let also H be a computer network where its 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 a homomorphism from G to H, the *H-coloring* problem [7]. Hell and Nešetřil proved a "dichotomy" theorem on the complexity of the H-coloring problem asserting that it is in P when H contains a loop or is bipartie, otherwise it is NP-complete [10]. Its counting version, the *$\#H$-coloring* problem presents a similar behaviour. Dyer and Greenhill proved another dichotomy theorem able to discriminate the cases where the $\#H$-coloring problem is #P-complete and the cases when it is polynomially solvable [6]. These results appear to have interesting connections with statistical physics as it implies the #P-completeness of the partition function of certain models such as the Widom-Rowlinson model or the Beach model [6]. Moreover the same dichotomies is preserved even if G has bounded degree [9, 6].

*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 TIC-2000-1970-CE. 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

If the processes have some restrictions, in the sense that can be executed only in some specified processors, this problem variation can be modeled by the *list H-coloring problem* [3, 11]. There are two dichotomy theorems on the complexity of the list $\#H$-coloring problem. The first states that if H is a *reflexive* graph (every vertex is looped) and the theorem states then the list H-coloring problem can be solved in polynomial time if H is an interval graph, and it is NP-complete otherwise [7]. The second theorem states that if H is *irreflexive* (no vertex is looped) then the list H-coloring is in P if the complement of H is a circular arc graph of clique covering number two, otherwise the problem is NP-complete [8]. The counting version of the list H-coloring problem is defined in the obvious way, we call it *list $\#H$-coloring* problem. The first result of this paper is that the dichotomy criterion for the $\#H$-coloring problem applies also for the list $\#H$-coloring problem.

Suppose now that, in the network embedding problem, we further restrict the number of jobs that can be assigned to a processor. This restriction is quite natural as in many practical cases the computing resources of some of the processors are restricted. This new problem constitutes a way to define variants of the H-coloring problem amenable to parameterization [2]. Fix a subset C of the vertices of H and select a weigth assignment K on C. A graph G has a *(H, C, K)-coloring* if it admits an H-coloring where the number of preimages of any vertex in C equals its weight. The *(H, C, K)-coloring* problem asks if an input graph G has a (H, C, K)-coloring. A parameterization of the (H, C, K)-coloring problem is obtained considering the numbers in K as small fixed constant. Also, the corresponding "list" version, the *list (H, C, K)-coloring* problem, is defined analogusly. In [2] several dichotomy theorems are proved concerning the fixed parameter tractability of the (H, C, K)-coloring and the list (H, C, K)-coloring problems. In this paper we study their counting version, the *$\#(H, C, K)$-coloring* and the *list $\#(H, C, K)$-coloring* problems.

To study the parameterized complexity of counting (H, C, K)-colorings, with the numbers in K as parameters, we set up two main questions: The first asks which partial weighted assignments (H, C, K) lead to #P-complete problems and which (H, C, K) lead to a polynomial time counting algorithm. Notice that in the cases where such a polynomial time algorithm exists, it may have time complexity $O(n^{f(K,H)})$, which makes the polynomial time algorithm rather impractical. On this point lies the genesis of our second question, characterize the set of partial weighted assignments, for which it is possible to design an algorithm with time complexity $O(f(K, H)n^{\alpha})$, for α a positive integer that does not depent on the parameter K. We mention that such complexity bounds have already beed achieved for the decisional versions [2] and other parameterizations of the H-coloring problems [1]. At first sight, it looks like these two questions fit in the parameterized complexity framework developed by Downey and Fellows [4, 5]. However, this is not formally correct as the parameterized complexity theory concerns only decisional problems.

In this paper we provide partial answers to the two questions for both the $\#(H, C, K)$-coloring and the list $\#(H, C, K)$-coloring problems. For the first question we give some conditions for (H, C, K) for which, the parameterization of K does not break the #P-completeness barrier of the list $\#H$-coloring problem and we give looser conditions where this is possible. Especially for the list $\#(H, C, K)$-coloring problem, our conditions merge to a dichotomy theorem. Moreover, we prove that the same dichotomy theorem holds for two other versions of list $\#(H, C, K)$-coloring defined in terms of their list distribution.

For the second question we prove that if the graph obtained by removing from H its parameterized vertices is edgeless, then the list $\#(H, C, K)$-coloring problem, as well as the $\#(H, C, K)$-coloring problem, can be solved in $O(kn+f(k,h))$ steps (k is the sum of the integers in K and $h = |V(H)|$). This algorithm is quite good as its non-polynomial part contributes to the overall complexity with an additive factor.

We suspect that this is sharp, because the problem of counting independent sets of size k can be cast as a $\#(H, C, K)$-coloring in which removing C lefts only one edge (see Figure 1). As the parameterized version of independent set is W [1]-hard, and thus unlikely to be in FPT, we cannot expect much improvement, unless the W hierarchy collapses [4].

The paper is organized as follows. In Section 2 we give the basic definitions and preliminary results. In Sections 3 and 4 we prove two dichotomy theorems: one characterizing the complexity of the list $\#H$-coloring problem and the other the complexity of the list $\#(H, C, K)$-coloring problem. In Section 5 we give a condition for the $\#\mathsf{P}$-completeness of the $\#(H, C, K)$-coloring problem to be $\#\mathsf{P}$-complete and in Section 6 we give a condition implying an $O(kn + f(k, h))$ time algorithm for the (list) $\#(H, C, K)$-coloring problem. Finally, in Section 7 we describe the consecuence of our results to other parametrization and we conclude in Section 8 with some open problems.

2 Definitions and preliminary results

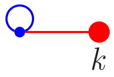
All the graphs in this paper are undirected, have no multiedges, but may have loops. Let G be a graph, for a set $S \subseteq V(G)$ we denote by $G[S]$ the subgraph induced by S, and by $G - S$ the graph $G[V(G) - S]$. For a vertex $v \in V$ let $N_G(v)$ denote the neighborhood v. We denote by $\gamma(G)$ the number of connected components of G.

For a function $\varphi : A \to B$ and any subset C of A, $\varphi|_C$ denotes the *the restriction of* φ *to* C. For a fixed graph H, we say that a mapping $\chi : V(G) \to V(H)$ is an *H-coloring* of G if for all $\{u, v\} \in E(G)$, $\{\chi(u), \chi(v)\} \in E(H)$.

The *H-coloring* problem asks whether, given a graph G, there is an H-coloring of G. Given a graph G and, for every $v \in V(G)$, a set $L(v) \subseteq V(H)$, the *list H-coloring* problem asks whether there is an H-coloring χ of G such that for every $u \in V(G)$ we have $\chi(u) \in L(u)$. Notice, that making every list $L(u) = V(H)$, the list H-coloring is the H-coloring problem.

The complexity of the list H-coloring has been analyzed for reflexive graphs in [7] and irreflexive graphs in [8]. We will also consider two variants of the list H-coloring, the *C-list H-coloring*: For every vertex $v \in V(G)$ the graph $H[L(v)]$ is connected and the *OA-list H-coloring*: the list are either $V(H)$ or a single vertex, both variants were introduced in [7].

We are interested in counting problems and, as usual, we will prefix the H in the problem name by $\#$. A basic result on the complexity of the $\#H$-coloring problem is the following.



IS

Figure 1: The partial weighted assignments (H, C, K) for the independent set

Theorem 2.1 ([6]). The $\#H$-coloring problem is $\#\mathsf{P}$-complete if H has a connected component which is not a complete reflexive graph or a complete irreflexive bipartite graph. Otherwise the counting problem is in P. Moreover, the same theorem holds even if G is a graph of bounded degree.

Following [2] the H-coloring problems can be *parameterized* as follows: Let $C = \{a_1, \dots, a_r\}$ be a set of r vertices of H and $K = (k_1, \dots, k_r)$ an r-tuple of non negative integers. We call the triple (H, C, K) a *partial weighted assignment* on H. We say that a mapping $\chi : V(G) \to V(H)$ is an (H, C, K)*-coloring* of G if χ is an H-coloring of G such that, for all i, $1 \leq i \leq r$, $|\chi^{-1}(v_i)| = k_i$. We call the vertices of C *weighted vertices*. For each parameter assignment (H, C, K) we will use the notation $h = |V(H)|$ and $k = \sum_{i=1,\dots,r} k_i$.

The same type of parameterization generates the *list* (H, C, K)*-coloring* and, the counting versions of both problems.

3 The list $\#H$-coloring problem

The following lemma concerns the cases for which counting the number of list H-colorings of a given graph can be done in polynomial time.

Lemma 3.1. The list $\#H$-coloring problem can be solved in polynomial time whenever H is a reflexive complete graph or an irreflexive complete bipartite graph.

Proof. Given a graph G and lists $L(u) \subseteq V(H)$ for each $u \in V(G)$, assume that G is connected, the proof considers three cases:
Case 1 H is an isolated vertex without a loop. If G has no edges and all the vertices have a non-empty list then there is a unique H-coloring of G, otherwise there is no H-coloring for G.
Case 2 H is a complete graph with h looped vertices. Notice that for such an H every mapping is an H-coloring, so we have only to compute the number of mappings compatible with the lists and, if $l_u = |L(u)|$, this number is

$$\prod_{u \in V(G)} l_u.$$

Case 3 H is a complete bipartite graph with no loops. We may assume without loss of generality that G is connected and bipartite. Assume that H has bipartition (C_1, C_2) and that G has bipartition (V_1, V_2). Now we split the vertex lists; for every $u \in V(G)$ let

$L_i(u) = L(u) \cap C_i$ and $l_i(u) = |L_i(u)|$ for $i = 1, 2$. Taking in to account that bipartitions must be preserved, the total number of list H-colorings of G is

$$\prod_{u \in V_1} l_1(u) \prod_{v \in V_2} l_2(v) + \prod_{u \in V_1} l_2(u) \prod_{v \in V_2} l_1(v).$$

When G is disconnected, the number of list H-colorings is the product of the number of list H-colorings of its connected components. □

We can extend the previous result as follows

Lemma 3.2. The list $\#H$-coloring problem can be solved in polynomial time whenever all the connected componente of H are either a reflexive complete graph or an irreflexive complete bipartite graph.

Proof. Let $(G, \{L(v)\})$ be an input to the list $\#H$-coloring problem. Assume that G has α connected componnets, $G_1, \dots, G_\alpha$, and that H has β connected components $H_1, \dots, H_\beta$.

For every i, j, $1 \leq i \leq \alpha$, $1 \leq j \leq \beta$, let $m_{i,j}$ be the number of list H_j-colorings of $(G_i, \{L(v) \cap V(H_j)\})$. In any list H-coloring of $(G, \{L(v)\})$, the connected components must be mapped to connected components, furthermore different conncted components can be mapped independently. Therefore, we have that the total number of list H-coloring of $(G, \{L(v)\})$ can be computed by

$$\sum_{\sigma \in \mathcal{F}} \prod_{i=1}^{\alpha} \quad m_{ij},$$

where $\mathcal{F}$ is the set of functions from $\{1, \dots, \alpha\}$ to $\{1, \dots, \beta\}$. By the previous lemma the values m_{ij} can be computed in polynomial time. □

Making each list equal to $V(H)$, the number of H-colorings of a given graph G coincides with the number of list H-colorings, therefore as a corollary to Lemma 3.2 and Theorem 2.1, we get

Theorem 3.1. The list $\#H$-coloring is $\#\mathsf{P}$-complete if H has a connected component which is not a complete reflexive graph or a complete irreflexive bipartite graph. Otherwise the counting problem is in P. Moreover, the same theorem holds even if G is a graph of degree bounded by some constant.

The above dichotomy theorem will be one of the ingredients of the proof of its parameterized analogus (Theorem 4.3). Furthermore we can extend the hardness results to the variants of the list H-coloring.

Lemma 3.3. 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 3.4. If the C-list $\#H$-coloring is in P then the $\#H$-coloring problem is in P.

Proof. Let G be a 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 dichotomy theorem.

Corollary 3.1. The C-list $\#H$-coloring and the OA-list $\#H$-coloring problems are $\#\mathsf{P}$-complete if H has a connected component which is not a complete reflexive graph or a complete irreflexive bipartite graph. Otherwise the counting problems are in P. Moreover, the same theorem holds even if G is a graph of degree bounded by some constant.

4 The list $\#(H, C, K)$-coloring problem

In this section we will prove a dichotomy theorem for the list $\#(H, C, K)$-coloring and its variations. We first show that for any parameter assignment (H, C, K) where the problem of counting list-$(H - C)$-colorings is in P, the problem of counting list $\#(H, C, K)$-coloring is also in P.

Theorem 4.1. 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.

Proof. The idea of the algorithm is the construction of a list $(H - C)$-coloring instance for each partial assignment of vertices to the parameterized nodes that constitute a list $(H[C], C, K)$-coloring. As the number of such partial assignments is at most n^k we obtain the result.

In particular, assume that we have a list $(H[C], C, K)$-coloring χ for $(G[D], \{L(d)\}$ for some $D \subseteq V(G)$ with k vertices. We consider the graph $G' = G[V(G) - D]$ and associate to each vertex $v \in V(G')$ the list

$$L'(v) = \begin{cases} L(v) \cap \bigcap_{N_G(v) \cap D} (N_H(\varphi(u)) - C), & \text{if } N_G(v) \cap D \neq \emptyset \\ V(H - C) & \text{otherwise} \end{cases}$$

Notice that χ can be completed to an list (H, C, K)-coloring if and only if $(G', \{L'(v)\})$ has a list $(H - C)$-coloring. Using Lemma 3.2, we get the result. □

We now give sufficient conditions for the $\#\mathsf{P}$-completeness of list $\#(H, C, K)$-coloring problem.

Theorem 4.2. For any parameter assignment (H, C, K), the list $\#(H, C, K)$-coloring problem is $\#\mathsf{P}$-complete whenever the $\#(H - C)$-coloring problem is $\#\mathsf{P}$-complete.

Proof. We set $H' = H - C$ and, in what follows, we will reduce the $\#H'$-coloring problem to the list $\#(H, C, K)$-coloring problem. Let G be an instance of the H'-coloring problem. We will construct an instance G' of the (H, C, K)-coloring problem as follows: Take as G' the disjoint union of G and a graph $F = (\{f_1^1, \dots, f_{k_1}^1 \dots f_1^r, \dots, f_{k_r}^r\}, \emptyset)$. We associated to every vertex $v \in V(G)$ the list $L(v) = V(H')$ and for the additional vertices, we make, for any i, $1 \leq i \leq r$, and any j, $1 \leq j \leq k_i$ $L(f_j^i) = \{a_i\}$.

From this construction we have that the H'-colorings of G are in one to one correspondence with the list H-coloring of (G', L). Therefore, if there exists a polynomial time algorithm counting the list (H, C, K)-colorings of (G', L) then a polynomial time algorithm can be constructed for counting the H'-colorings of G. $\square$

Putting together Theorems 3.1,4.1, and 4.2 we get

Theorem 4.3. The list $\#(H, C, K)$-coloring is $\#\mathsf{P}$-complete if $H - C$ has a connected component which is not a complete reflexive graph or a complete irreflexive bipartite graph. Otherwise the counting problem is in P. Moreover, the same theorem holds even if G is a graph of degree bounded for some constant.

The reduction provided in Theorem 4.2 generates an instance in which the list are either $V(H)$ or a single vertex. Therefore, using Theorems 4.1 and 4.2 we have

Corollary 4.1. The OA-list $\#(H, C, K)$-coloring is $\#\mathsf{P}$-complete if $H - C$ has a connected component which is not a complete reflexive graph or a complete irreflexive bipartite graph. Otherwise the counting problem is in P. Moreover, the same theorem holds even if G is a graph of degree bounded for some constant.

The reduction provided in Theorem 4.2 does not generate a connected list instance. However, we can replace the list $V(H)$ by the set of vertices in the connected component of $H - C$ that is not a complete reflexive graph or a complete irreflexive bipartite graph we get the same $\#\mathsf{P}$-completeness result. This together with Theorem 4.1 gives.

Corollary 4.2. The C-list $\#(H, C, K)$-coloring is $\#\mathsf{P}$-complete if $H - C$ has a connected component which is not a complete reflexive graph or a complete irreflexive bipartite graph. Otherwise the counting problem is in P. Moreover, the same theorem holds even if G is a graph of degree bounded for some constant.

5 The $\#(H, C, K)$-coloring problem

As any H-coloring problem can be seen as a list H-coloring in which the list for each vertex is $V(H)$. Therefore, all the positive results for the list $\#(H, C, K)$-coloring problem translate as well to the $\#(H, C, K)$-coloring problem. Hardness results seems more difficult to obtain, we have been able to obtain them only for a particular type of parameter assignmemnt.

Let (H, C, K) be a partial weighted assignment and let a_i be a vertex in C. We call it *i-reducible* if H is H'-colorable where $H' = H - \{a_i\}$. We say that (H, C, K) is *reducible* if it is i-reducible for some $1 \leq i \leq r$, and we say that it is *irreducible* when it is not reducible.

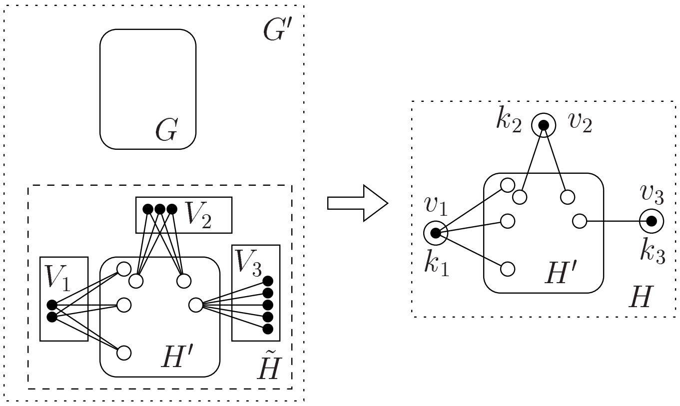


Figure 2: The reduction of Theorem 5.1

Lemma 5.1. For any irreducible parameter assignment (H, C, K), the problem of counting the $\#(H, C, K)$-colorings of a graph G is $\#\mathsf{P}$-complete whenever the $\#(H - C)$-coloring problem is $\#\mathsf{P}$-complete.

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

Define the graph $\tilde{H}$ constructed from H' in such a way that each vertex $a_i \in C$ is replaced by a set of k_i vertices $V_i = \{v_i^1, \dots, v_i^{k_i}\}$. Any edge connecting two vertices in $V(H) - C$ is maintained, while an edge connecting $u \in V(H) - C$ with $a_i \in C$ is replicated as $\{(u, v_i^j) \mid 1 \leq j \leq k_i\}$. An edge connecting a vertex a_i and $a_{i'}$ in C is replicated as $\{(v_i^j, v_{i'}^{j'}) \mid 1 \leq j \leq k_i, 1 \leq j' \leq k_{i'}$.

Notice that $\tilde{H}$ is (H, C, K)-colorable, as we can set $\chi(v) = v$ for $v \in V(\tilde{H}) - C'$ and $\chi(v) = a_i$ for $v \in V_i$. Furthermore as (H, C, K) is irreducible, the vertices in V_i can only be mapped to a_i.

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 a (H, C, K)-coloring.

Notice now that if G is H'-colorable, then G' is also (H, C, K)-colorable, because $\tilde{H}$ is (H, C, K)-colorable.

If G' is (H, C, K)-colorable then, as the vertices in V_i can only be mapped to vertices in C, and its size is k, all the vertices of G must be mapped to vertices of H', therefore G is H'-colorable.

Notice that 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 $\tilde{H}$. As this last number can be computed in consant time, the claim follows. $\square$

The following is a direct consequence of Lemmata 5.1, 4.2, and Theorem 3.2.

Theorem 5.1. For any irreducible partial weight assignment (H, C, K). The problem of counting the $\#(H, C, K)$-coloring problem is $\#\mathsf{P}$-complete if H has a connected component which is not a complete reflexive graph or a complete irreflexive bipartite graph.

Algorithm Count-List-Colorings(G, L, H, C, K).

Input: Two graphs G, H, a function $L : V(G) \to 2^{V(H)}$, and a partial weighted assignment (H, C, K) on H such that $E(H - C) = \emptyset$.
Output: The number of list (H, C, K)-colorings of G.

1: Let R_1 be the set of vertices in G of degree $> k$.
2: If $|R_1| > k$ or $|E(G)| > kn$ then return 0.
3: Set $G' = G[V(G) - R_1]$.
4: Let R_2 be the non isolated vertices in G' and let R_3 be the isolated vertices in G'.
5: If $|R_2| > k^2 + k$ then return 0
6: Set up a partition $\mathcal{R} = (P_1, \ldots, P_m), m \leq q \cdot 2^h$ of R_3 where
$\forall_{v,u \in R_3} \; (\exists_{1 \leq i \leq qh} \; \{v, u\} \in P_i \Leftrightarrow (N_G(v) = N_G(u) \wedge (L(v) = L(u))))$.
(q is the number of different neighborhoods of vertices in R_3)
7: Let $Q = \emptyset$.
8: For $i = 1, \ldots, m$,
if $|P_i| \leq k + 1$, then set $F_i = P_i$,
otherwise let F_i be any subset of P_i where $|F_i| = k + 1$.
Set $Q = Q \cup F_i$ and $P_i = P_i - F_i$.
9: Let $\mathcal{H}$ be the set of all the list (H, C, K)-colorings of $G[R_1 \cup R_2 \cup Q]$.
10: Set $\eta = 0$.
11: For any $\chi \in \mathcal{H}$, do
Set $\beta = 1$.
For any nonempty $P_i \in \mathcal{R}$, do
Let V_i contain any vertex x of $H - C$ where $N_H(x) \supseteq \chi(P_i)$.
Set $\beta := \beta \cdot |V_i|^{|P_i|}$
Set $\eta := \eta + \beta$
12: return η.
13: End.

Figure 3: Algorithm Count-List-Colorings

Otherwise the counting problem is in P. Moreover, the same theorem holds even if G is a graph of bounded degree.

6 Easy cases

In this subsection we will give a sufficient condition for the list $\#(H, C, K)$-coloring problem to be solvable in $O(kn + f(k, h))$ steps. The main idea of our algorithm is to create an equivalent problem whose size does not depend in the size of G (steps **1**–**8** of the algorithm).

Lemma 6.1. There exists an $O(kn + f(k, h))$ time algorithm that returns the number of (H, C, K)-colorings of G when $E(H - C) = \emptyset$

Proof. The algorithm is given in Figure 3. We proceed now to justify the correction of all the steps. We will refer to the notation used in the algorithm.

Notice first, that if G has a (H, C, K)-coloring χ, then $\chi^{-1}(C)$ is a vertex cover of G. This implies that any vertex $v \in \chi^{-1}(V(H) - C)$ is adjacent in G only with vertices in $\chi^{-1}(C)$. As the vertices of $\chi^{-1}(C)$ are exactly $k = \sum_{1 \leq i \leq r} k_i$ we have that $d_G(v) \leq k$ for any $v \in \chi^{-1}(V(H) - C)$. Therefore, if a vertex in G has degree $> k$, then it must be mapped to some vertex in C, i.e. $R_1 \subseteq \chi^{-1}(C)$. Moreover, it is easy to see that $|E(G)| \leq kn$. This justifies step **2** of the algorithm. In order to justify step **5**, we claim that, if there exists a (H, C, K)-coloring of G, then the non isolated vertices of G' are at most $k^2 + k$. To prove this claim, notice that all the vertices of G' have degree at most k and that exactly k of them are mapped via χ in C. The rest, if not isolated, should be adjacent to vertices in $\chi^{-1}(C)$ (otherwise, as in the justification of step **2**, C cannot be a vertex cover of H). This means that $\chi^{-1}(C)$ and its neighborhood are containing all the non isolated vertices of G'. As any vertex in $\chi^{-1}(C)$ has degree bounded by k in G', the claim follows. It now remains to prove that G has an (H, C, K)-coloring iff $\tilde{G} = R_1 \cup R_2 \cup Q$ has.

After computing the set $\mathcal{H}$, for each $\chi \in \mathcal{H}$, the algorithm computes incrementally the ways the vertices of each set P_i can contribute to the number of the (H, C, K)-colorings of $\tilde{G}$. Let β_i be the number of (H, C, K)-colorings of $G[V(\tilde{G}) \cup P_1 \cup \cdots, \cup P_i]$. Suppose that the algorithm computes correctly β_{i-1}. If P_i is empty then, obviously, $\beta_i = \beta_{i-1}$. Notice now that, if P_i is nonempty, the set V_i computed will be also non empty. This is the case as the definition of P_i implies that there will be some vertex $v \in V(\tilde{G})$, where $N_G(v) = N_G(P_i)$ and such that v is mapped, via χ, to some vertex of $H - C$. Notice that V_i is exactly the set of vertices of H where vertices of P_i can be mapped (recall that all the vertices in $\cup_{i=1,\ldots,2^{|R_1|}} P_i$ are pairwise independent). Therefore, β_i is equal to β_{i-1} multiplied by the number of ways the vertices of P_i can be mapped to the vertices of V_i. This implies that $\beta_i = \beta_{i-1}|V_i|^{|P_i|}$ and proves the correctness of the algorithm.

Let us verify the complexity. Step **1** takes $O(kn)$ steps. Step **2** costs $O(kn)$ steps. Step **3** costs $O(kn)$, and Steps **4** and **5** cost both $O(n)$ steps. Step **6** can be accomplished in $O(kn + 2^{k \cdot h})$ steps. As $m \leq 2^k \cdot 2^h$, step **8** needs $O((k+1) \cdot 2^{k \cdot h})$ steps. Notice that $|Q| \leq (k+1) \cdot 2^{k \cdot h}$. Therefore $G[R_1 \cup R_2 \cup Q]$ has at most $k^2 + 2k + (k+1) \cdot 2^{k \cdot h}$ vertices and Step **9** can be implemented in $O(|H|^{|R_1 \cup R_2 \cup Q|}) = O(h^{k^2+k+(k+1)2^k})$ steps. Consequently, The first loop of Step **11** is repeated at most $h^{k^2+k+(k+1)2^k}$ times. The second loop is repeated at most $|\mathcal{R}| = 2^{k \cdot h}$ times and the computation of V_i can be done in $O(hk^2)$ steps. All these together and under the assumption that arithmetical operations can be done in $O(1)$ steps, the overall complexity of the algorithm is $O(kn + 2^{k \cdot h} k^2 h^{k^2+2k+(k+1)2^{k \cdot h}+1})$. □

Notice, the above algorithm can also be used to count (H, K, C)-colorings. Its complexity can be slightly improved for this problem if we remove the condition $(L(v) = L(u))$ from Step **6**. For a fast algorithm for the (H, C, K)-coloring problem, in more general cases than those of Theorem 6.1, see [2].

7 Other parameterizations

We can obtain another parameterization by relaxing the condition of the number of vertices in C. If in the definition of (H, C, K)-coloring we replace "=" with "$\leq$" we say that χ is a $(H, C, \leq K)$-*coloring* of G. That way we can define the "$\leq$"-versions of all the problems mentioned so far.

We claim that all the results of the previous sections hold also if we replace (H, C, K) by $(H, C, \leq K)$.

For the positive results it is enough to observe that the $\#(H, C, \leq K)$-coloring problem can be reduced to the computation of at most k^r instances of the $\#(H, C, K')$-problem where $K' \leq K$.

For the negative results, notice that 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 selecting arbitrarily a non-zero component of K, and decreasing it by one.

Finally we mention that the positive results of Theorem 6.1 can be easily adapted to solve their "$\leq$"-versions. That way, we can maintain the same complexity bounds without using the reduction described above.

8 Open problems

It remains open to prove the negative part of Theorem 5.1 for reducible partial weighted assignments. A natural question is to know whether the $O(f(K, H)n^\alpha)$-time complexity bounds of Theorem 6.1 can be extended to a wider class of graphs. To prove negative results on this direction, we need to show completeness for some counting analogue of the W-hierarchy defined by Downey and Fellows for studying the parameterized complexity of decision problems, see [4]. Unfortunately, for counting problems such a theory is to be developed.

References

[1] J. Díaz, M. Serna, and D. M. Thilikos. Evaluating H-colorings of graphs with bounded decomposability. In *Proceedings of The 7th Annual International Computing and Combinatorics Conference, COCOON 2001*, 2001.

[2] J. Díaz, M. Serna, and D. M. Thilikos. (H, C, K)-colorings: Fast, easy and hard cases. In *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science, MFCS-2001*, 2001.

[3] Q. Donner. On the number of list-colorings. *J. Graph Theory*, 16(3):239–245, 1992.

[4] R. G. Downey and M. R. Fellows. *Parameterized complexity*. Springer-Verlag, New York, 1999.

[5] R. G. Downey and M. R. Fellows. Parameterized complexity after (almost) ten years: review and open questions. In *Combinatorics, computation & logic '99 (Auckland)*, pages 1–33. Springer, Singapore, 1999.

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

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

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

[9] A. Galluccio, P. Hell, and J. Nešetřil. The complexity of H-colouring of bounded degree graphs. *Discrete Mathematics*, 222(1-3):101–109, 2000.

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

[11] J. Kratochvíl and Z. Tuza. Algorithmic complexity of list colorings. *Discrete Applied Mathematics*, 50(3):297–302, 1994.