A Complexity Dichotomy for Critical Values of the b-Chromatic Number of Graphs

Lars Jaffke
Department of Informatics, University of Bergen, Norway
lars.jaffke@uib.no

Paloma T. Lima
Department of Informatics, University of Bergen, Norway
paloma.lima@uib.no

Abstract

A *b-coloring* of a graph G is a proper coloring of its vertices such that each color class contains a vertex that has at least one neighbor in all the other color classes. The b-Coloring problem asks whether a graph G has a b-coloring with k colors. The *b-chromatic number* of a graph G, denoted by $\chi_b(G)$, is the maximum number k such that G admits a b-coloring with k colors. We consider the complexity of the b-Coloring problem, whenever the value of k is close to one of two upper bounds on $\chi_b(G)$: The maximum degree $\Delta(G)$ plus one, and the m-degree, denoted by $m(G)$, which is defined as the maximum number i such that G has i vertices of degree at least $i-1$. We obtain a dichotomy result for all fixed $k \in \mathbb{N}$ when k is close to one of the two above mentioned upper bounds. Concretely, we show that if $k \in \{\Delta(G)+1-p, m(G)-p\}$, the problem is polynomial-time solvable whenever $p \in \{0,1\}$ and, even when $k=3$, it is NP-complete whenever $p \geq 2$. We furthermore consider parameterizations of the b-Coloring problem that involve the maximum degree $\Delta(G)$ of the input graph G and give two FPT-algorithms. First, we show that deciding whether a graph G has a b-coloring with $m(G)$ colors is FPT parameterized by $\Delta(G)$. Second, we show that b-Coloring is FPT parameterized by $\Delta(G)+\ell_k(G)$, where $\ell_k(G)$ denotes the number of vertices of degree at least k.

2012 ACM Subject Classification Mathematics of computing → Graph coloring

Keywords and phrases b-Coloring, b-Chromatic Number

Digital Object Identifier 10.4230/LIPIcs.MFCS.2019.34

Related Version A full version is available at: `https://arxiv.org/abs/1811.03966` [13].

Funding *Lars Jaffke*: Supported by The Bergen Research Foundation (BFS).
Paloma T. Lima: Supported by Research Council of Norway via the project "CLASSIS".

Acknowledgements We would like to thank Fedor Fomin for useful advice with good timing and Petr Golovach for pointing out the reference [7].

1 Introduction

Given a set of colors, a *proper coloring* of a graph is an assignment of a color to each of its vertices in such a way that no pair of adjacent vertices receive the same color. In the deeply studied Graph Coloring problem, we are given a graph and the question is to determine the smallest set of colors with which we can properly color the input graph. This problem is among Karp's famous list of 21 NP-complete problems [14] and since it often arises in practice, heuristics to solve it are deployed in a wide range of applications. A very natural such heuristic is the following. We greedily find a proper coloring of the graph, and then try to *suppress* any of its colors in the following way: say we want to suppress color c. If there is a vertex v that has received color c, and there is another color $c' \neq c$ that does not appear in the neighborhood of v, then we can safely recolor the vertex v with color c' without making the coloring improper. We terminate this process once we cannot suppress any color anymore.

To predict the worst-case behavior of the above heuristic, Irving and Manlove defined the notions of a *b-coloring* and the *b-chromatic number* of a graph [12]. A b-coloring of a graph G is a proper coloring such that in every color class there is a vertex that has a neighbor in all of the remaining color classes, and the b-chromatic number of G, denoted by $\chi_b(G)$, is the maximum integer k such that G admits a b-coloring with k colors. We observe that in a b-coloring with k colors, there is no color that can be suppressed to obtain a proper coloring with $k-1$ colors, hence $\chi_b(G)$ describes the worst-case behavior of the previously described heuristic on the graph G. We consider the following two computational problems associated with b-colorings of graphs.

b-Coloring

Input:	Graph G, integer k
Question:	Does G admit a b-coloring with k colors?

b-Chromatic Number

Input:	Graph G, integer k
Question:	Is $\chi_b(G) \geq k$?

We would like to point out an important distinction from the "standard" notion of proper colorings of graphs: If a graph G has a b-coloring with k colors, then this implies that $\chi_b(G) \geq k$. However, if $\chi_b(G) \geq k$ then we can in general not conclude that G has a b-coloring with k colors. A graph for which the latter implication holds as well is called *b-continuous*. This notion is mostly of structural interest, since the problem of determining if a graph is b-continous is NP-complete even if an optimal proper coloring and a b-coloring with $\chi_b(G)$ colors are given [2].

Besides observing that $\chi_b(G) \leq \Delta(G)+1$ where $\Delta(G)$ denotes the maximum degree of G, Irving and Manlove [12] defined the *m-degree* of G as the largest integer i such that G has i vertices of degree at least $i-1$. It follows that $\chi_b(G) \leq m(G)$. Since the definition of the b-chromatic number originated in the analysis of the worst-case behavior of graph coloring heuristics, graphs whose b-chromatic numbers take on *critical* values, i.e. values that are close to these upper bounds, are of special interest. In particular, identifying them can be helpful in structural investigations concerning the performance of graph coloring heuristics.

In terms of computational complexity, Irving and Manlove showed that both b-Coloring and b-Chromatic Number are NP-complete [12] and Sampaio observed that b-Coloring is NP-complete even for every fixed integer $k \geq 3$ [17]. Panolan et al. [16] gave an exact exponential algorithm for b-Chromatic Number running in time $\mathcal{O}(3^n n^4 \log n)$ and an algorithm that solves b-Coloring in time $\mathcal{O}(\binom{n}{k} 2^{n-k} n^4 \log n)$. From the perspective of parameterized complexity [6, 8], it has been shown that b-Chromatic Number is W[1]-hard parameterized by k [16] and that the dual problem of deciding whether $\chi_b(G) \geq n-k$, where n denotes the number of vertices in G, is FPT parameterized by k [11].

Since the above mentioned upper bounds $\Delta(G)+1$ and $m(G)$ on the b-chromatic number are trivial to compute, it is natural to ask whether there exist efficient algorithms that decide whether $\chi_b(G) = \Delta(G)+1$ or $\chi_b(G) = m(G)$. It turns out both these problems are NP-complete as well [10, 12, 15]. However, it is known that the problem of deciding whether a graph G admits a b-coloring with $k = \Delta(G)+1$ colors is FPT parameterized by k [16, 17].

The Dichotomy Result. One of the main contributions of this paper is a complexity dichotomy of the b-Coloring problem for fixed k, whenever k is close to either $\Delta(G)+1$ or $m(G)$. In particular, for fixed $k \in \{\Delta(G)+1-p, m(G)-p\}$, we show that the problem is polynomial-time solvable when $p \in \{0,1\}$ and, even in the case $k=3$, NP-complete for all fixed $p \geq 2$. More specifically, we give XP time algorithms for the cases $k=m(G)$, $k=\Delta(G)$, and $k=m(G)-1$ which together with the FPT algorithm for the case $k=\Delta(G)+1$ [16, 17] and the aforementioned NP-hardness result for $k=3$ complete the picture. We now formally state this result.

▶ **Theorem 1.** *Let G be a graph, $p \in \mathbb{N}$ and $k \in \{\Delta(G)+1-p, m(G)-p\}$. The problem of deciding whether G has a b-coloring with k colors is*
(i) NP*-complete if k is part of the input and $p \in \{0,1\}$,*
(ii) NP*-complete if $k=3$ and $p \geq 2$, and*
(iii) *polynomial-time solvable for any fixed positive k and $p \in \{0,1\}$.*

Maximum Degree Parameterizations. The positive results in our dichotomy theorem provide XP-algorithms to decide whether a graph has a b-coloring with a number of colors that either *precisely meets* or is *one below* one of two upper bounds on the b-chromatic number, with the parameter being the number of colors in each of the cases. Towards more "flexible" tractability results, we consider parameterized versions of b-Coloring that involve the maximum degree $\Delta(G)$ of the input graph G, but ask for the existence of b-colorings with a number of colors that in general is different from $\Delta(G)+1$ or $\Delta(G)$.

▶ **Theorem 2.** *Let G be a graph. The problem of deciding whether G has a b-coloring with $m(G)$ colors is* FPT *parameterized by $\Delta(G)$.*

One of the crucially used facts in the algorithm of the previous theorem is that if we ask whether a graph G has a b-coloring with $k=m(G)$ colors, then the number of vertices of degree at least k is at most k. We generalize this setting and parameterize b-Coloring by the maximum degree plus the number of vertices of degree at least k.

▶ **Theorem 3.** *Let G be a graph. The problem of deciding whether G has a b-coloring with k colors is* FPT *parameterized by $\Delta(G)+\ell_k(G)$, where $\ell_k(G)$ denotes the number of vertices of degree at least k in G.*

We now argue that parameterizing by only one of the two invariants used in Theorem 3 is not sufficient to obtain efficient parameterized algorithms. From the result of Kratochvíl et al. [15], stating that b-Coloring is NP-complete for $k=\Delta(G)+1$, it follows that b-Coloring is NP-complete when $\Delta(G)$ is unbounded and $\ell_k(G)=0$. On the other hand, Theorem 1(ii) implies that b-Coloring is already NP-complete when $k=3$ and $\Delta(G)=4$. Together, this rules out the possibility of FPT- and even of XP-algorithms for parameterizations by one of the two parameters alone, unless P = NP.

Parameterizations of graph coloring problems by the number of high degree vertices have previously been considered for vertex coloring [1] and edge coloring [9]. Throughout the text, proofs of statements marked with "♣" are deferred to the full version [13].

2 Preliminaries

We use the following notation: For $k \in \mathbb{N}$, $[k] := \{1,\ldots,k\}$. For a function $f\colon X \to Y$ and $X' \subseteq X$, we denote by $f|_{X'}$ the restriction of f to X' and by $f(X')$ the set $\{f(x) \mid x \in X'\}$. For a set X and an integer n, we denote by $\binom{X}{n}$ the set of all size-n subsets of X.

Graphs. Throughout the paper a graph G with vertex set $V(G)$ and edge set $E(G) \subseteq \binom{V(G)}{2}$ is finite and simple. We often denote an edge $\{u, v\} \in E(G)$ by the shorthand uv. For graphs G and H we denote by $H \subseteq G$ that H is a subgraph of G, i.e. $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. We often use the notation $n \coloneqq |V(G)|$. For a vertex $v \in V(G)$, we denote by $N_G(v)$ the *open neighborhood* of v in G, i.e. $N_G(v) = \{w \in V(G) \mid vw \in E(G)\}$, and by $N_G[v]$ the *closed neighborhood* of v in G, i.e. $N_G[v] \coloneqq \{v\} \cup N_G(v)$. For a set of vertices $X \subseteq V(G)$, we let $N_G(X) \coloneqq \bigcup_{v \in X} N_G(v) \setminus X$ and $N_G[X] \coloneqq X \cup N_G(X)$. When G is clear from the context, we abbreviate "N_G" to "N". The *degree* of a vertex $v \in V(G)$ is the size of its open neighborhood, and we denote it by $\deg_G(v) \coloneqq |N_G(v)|$ or simply by $\deg(v)$ if G is clear from the context. For an integer k, we denote by $\ell_k(G)$ the number of vertices of degree at least k in G.

For a vertex set $X \subseteq V(G)$, we denote by $G[X]$ the subgraph *induced* by X, i.e. $G[X] \coloneqq (X, E(G) \cap \binom{X}{2})$. We furthermore let $G - X \coloneqq G[V(G) \setminus X]$ be the subgraph of G obtained from removing the vertices in X and for a single vertex $x \in V(G)$, we use the shorthand "$G - x$" for "$G - \{x\}$".

A graph G is said to be *connected* if for any 2-partition (X, Y) of $V(G)$, there is an edge $xy \in E(G)$ such that $x \in X$ and $y \in Y$, and *disconnected* otherwise. A *connected component* of a graph G is a maximal connected subgraph of G. A *path* is a connected graph of maximum degree two, having precisely two vertices of degree one, called its *endpoints*. The *length* of a path is its number of edges. Given a graph G and two vertices u and v, the *distance* between u and v, denoted by $dist_G(u, v)$ (or simply $dist(u, v)$ if G is clear from the context), is the length of the shortest path in G that has u and v as endpoints.

A graph G is a *complete graph* if every pair of vertices of G is adjacent. A set $C \subseteq V(G)$ is a *clique* if $G[C]$ is a complete graph. A set $S \subseteq V(G)$ is an *independent set* if $G[S]$ has no edges. A graph G is a *bipartite graph* if its vertex set can be partitioned into two independent sets. A bipartite graph with bipartition (A, B) is a *complete bipartite graph* if all pairs consisting of one vertex from A and one vertex from B are adjacent, and with $a = |A|$ and $b = |B|$, we denote it by $K_{a,b}$. A *star* is the graph $K_{1,b}$, with $b \geq 2$, and we call *center* the unique vertex of degree b and *leaves* the vertices of degree one.

Colorings. Given a graph G, a map $\gamma\colon V(G) \to [k]$ is called a *coloring of G with k colors*. If for every pair of adjacent vertices, $uv \in E(G)$, we have that $\gamma(u) \neq \gamma(v)$, then the coloring γ is called *proper*. For $i \in [k]$, we call the set of vertices $u \in V(G)$ such that $\gamma(u) = i$ the *color class i*. If for all $i \in [k]$, there exists a vertex $x_i \in V(G)$ such that

(i) $\gamma(x_i) = i$, and

(ii) for each $j \in [k] \setminus \{i\}$, there is a neighbor $y \in N_G(x_i)$ of x_i such that $\gamma(y) = j$,

then γ is called a *b-coloring* of G. For $i \in [k]$, we call a vertex x_i satisfying the above two conditions a *b-vertex for color i*.

Parameterized Complexity. Let Σ be an alphabet. A *parameterized problem* is a set $\Pi \subseteq \Sigma^* \times \mathbb{N}$. A parameterized problem Π is said to be *fixed-parameter tractable*, or contained in the complexity class FPT, if there exists an algorithm that for each $(x, k) \in \Sigma^* \times \mathbb{N}$ decides whether $(x, k) \in \Pi$ in time $f(k) \cdot |x|^c$ for some computable function f and fixed integer $c \in \mathbb{N}$. A parameterized problem Π is said to be contained in the complexity class XP if there is an algorithm that for all $(x, k) \in \Sigma^* \times \mathbb{N}$ decides whether $(x, k) \in \Pi$ in time $f(k) \cdot n^{g(k)}$ for some computable functions f and g.

A *kernelization algorithm* for a parameterized problem $\Pi \subseteq \Sigma^* \times \mathbb{N}$ is a polynomial-time algorithm that takes as input an instance $(x, k) \in \Sigma^* \times \mathbb{N}$ and either correctly decides whether

$(x,k) \in \Pi$ or outputs an instance $(x',k') \in \Sigma^* \times \mathbb{N}$ with $|x'| + k' \leq f(k)$ for some computable function f for which $(x,k) \in \Pi$ if and only if $(x',k') \in \Pi$. We say that Π *admits a kernel* if there is a kernelization algorithm for Π.

3 Hardness Results

In this section we prove the hardness results of our complexity dichotomy. First, we show that b-Chromatic Number and b-Coloring are NP-complete for $k = m(G) - 1 = \Delta(G)$, based on a reduction for the case $k = m(G)$ due to Havet et al. [10].

▶ **Theorem 4 (♣).** *b-Chromatic Number and b-Coloring are NP-complete, even when $k = m(G) - 1 = \Delta(G)$.*

The previous theorem, together with the result that b-Coloring is NP-complete when $k = \Delta(G) + 1$ [15] and when $k = m(G)$ [10], proves Theorem 1(i). We now turn to the proof of Theorem 1(ii), that is, we show that b-Coloring remains NP-complete for $k = 3$ if $k = \Delta(G) + 1 - p$ or $k = m(G) - p$ for any $p \geq 2$, based on a reduction due to Sampaio [17]. Note that the following proposition indeed proves Theorem 1(ii) as for fixed $p \geq 2$, we have that $3 \in \{\Delta(G) + 1 - p, m(G) - p\}$ if and only if $\Delta(G) = p + 2$ or $m(G) = p + 3$.

▶ **Proposition 5 (♣).** *For every fixed integer $p \geq 2$, the problem of deciding whether a graph G has a b-coloring with 3 colors is NP-complete when $\Delta(G) = p + 2$ or $m(G) = p + 3$.*

Since b-Chromatic Number and b-Coloring are known to be NP-complete when $k = \Delta(G) + 1$ [15], we make the following observation which is of relevance to us since in Section 5.2, we show that b-Coloring is FPT parameterized by $\Delta(G) + \ell_k(G)$.

▶ **Observation 6.** *b-Chromatic Number and b-Coloring are NP-complete on graphs with $\ell_k(G) = 0$, where k is the integer associated with the respective problem.*

4 Dichotomy Algorithms

In this section we give the algorithms in our dichotomy result, proving Theorem 1(iii). We show that for fixed $k \in \mathbb{N}$, the problem of deciding whether a graph G admits a b-coloring with k colors is polynomial-time solvable when $k = m(G)$ (Sect. 4.2), when $k = \Delta(G)$ (Sect. 4.3), and when $k = m(G) - 1$ (Sect. 4.4), by providing XP-algorithms for each case.

A natural way of solving the b-Coloring problem is to first try to identify a set of k b-vertices, color them bijectively with colors from $[k]$, and for each vertex in the set a set of $k - 1$ neighbors that can be colored in such a way that the vertex becomes a b-vertex for its color. Then try to extend the resulting coloring to the remainder of the graph. We enumerate all such sets and colorings, and show that the extension problem is solvable in polynomial time in each of the above cases.

The strategy of identifying the set of b-vertices and subsets of their neighbors that make them b-vertices was (for instance) also used to give polynomial-time algorithms to compute the b-chromatic number of trees [12] and graphs with large girth [4]. We capture it by defining the notion of a *b-precoloring* in the next subsection.

4.1 b-Precolorings

All algorithms in this section are based on guessing a proper coloring of several vertices in the graph, for which we now introduce the necessary terminology and establish some preliminary results.

▶ **Definition 7** (Precoloring). *Let G be a graph and $k \in \mathbb{N}$. A* precoloring *with k colors of a graph G is an assignment of colors to a subset of its vertices, i.e. for $X \subseteq V(G)$, it is a map $\gamma_X : X \to [k]$. We call γ_X* proper*, if it is a proper coloring of $G[X]$. We say that a coloring $\gamma : V(G) \to [k]$* extends *γ_X, if $\gamma|_X = \gamma_X$.*

We use the following notation. For two precolorings γ_X and γ_Y with $X \cap Y = \emptyset$, we denote by $\gamma_X \cup \gamma_Y$ the precoloring that colors the vertices in X according to γ_X and the vertices in Y according to Y, i.e. the precoloring $\gamma_{X\cup Y} \coloneqq \gamma_X \cup \gamma_Y$ defined as follows: for all $v \in X \cup Y$, if $v \in X$, then $\gamma_{X\cup Y}(v) = \gamma_X$, and if $v \in Y$ then $\gamma_{X\cup Y}(v) = \gamma_Y(v)$.

Next, we define a special type of precoloring with the property that any proper coloring that extends it is a b-coloring of the graph.

▶ **Definition 8** (b-Precoloring). *Let G be a graph, $k \in \mathbb{N}$, $X \subseteq V(G)$ and γ_X a precoloring. We call γ_X a* b-precoloring with k colors *if γ_X is a b-coloring of $G[X]$. A b-precoloring γ_X is called* minimal *if for any $Y \subset X$, $\gamma_X|_Y$ is not a b-precoloring.*

It is immediate that any b-coloring can be obtained by extending a minimal b-precoloring, a fact that we capture in the following observation.

▶ **Observation 9.** *Let G be a graph, $k \in \mathbb{N}$, and γ a b-coloring of G with k colors. Then, there is a set $X \subseteq V(G)$ such that $\gamma|_X$ is a minimal b-precoloring.*

The next observation captures the structure of minimal b-precolorings with k colors. Roughly speaking, each such precoloring only colors a set of k b-vertices and for each b-vertex a set of $k-1$ of its neighbors that make that vertex the b-vertex of its color. We will use this property in the enumeration algorithm in this section to guarantee that we indeed enumerate all minimal b-precolorings with a given number of colors.

▶ **Observation 10.** *Let γ_X be a minimal b-precoloring with k colors. Then, $X = B \cup Z$, where*
(i) *$B = \{x_1, \ldots, x_k\}$ and for $i \in [k]$, $\gamma_X(x_i) = i$, and*
(ii) *$Z = \bigcup_{i\in[k]} Z_i$, where $Z_i \in \binom{N(x_i)}{k-1}$ and $\gamma_X(Z_i) = [k] \setminus \{i\}$.*

We are now ready to give the enumeration algorithm for minimal b-precolorings.

▶ **Lemma 11** (♣). *Let G be a graph on n vertices and $k \in \mathbb{N}$. The number of minimal b-precolorings with k colors of G is at most*

$$\beta(k) \coloneqq n^k \cdot \Delta^{k(k-1)} \cdot (k-1)!^k, \tag{1}$$

where $\Delta \coloneqq \Delta(G)$ and they can be enumerated in time $\beta(k) \cdot k^{\mathcal{O}(1)}$.

4.2 Algorithm for $k = m(G)$

Our first application of Lemma 11 is to solve the b-Coloring problem in the case when $k = m(G)$ in time XP parameterized by k. It turns out that in this case, we are dealing with a Yes-instance as soon as we found a b-precoloring in the input graph that also colors all high-degree vertices (see Claim 12.1).

▶ **Theorem 12.** *Let G be a graph. There is an algorithm that decides whether G has a b-coloring with $k = m(G)$ colors in time $n^{k^2} \cdot 2^{\mathcal{O}(k^2 \log k)}$.*

Proof. Let $D \subseteq V(G)$ denote the set of vertices in G that have degree at least k. Note that by the definition of $m(G)$, we have that $|D| \le k$.

▶ **Claim 12.1** (♣). *G has a b-coloring with k colors if and only if G has a b-precoloring γ_X such that $D \subseteq X$ and there exists $S \subseteq D$ such that $\gamma_X|_{(X \setminus S)}$ is a minimal b-precoloring.*

The algorithm enumerates all minimal b-precolorings with k colors and for each such precoloring, it enumerates all colorings of the vertices D. If combining one such pair of precolorings gives a b-precoloring, it returns a greedy extension of it; otherwise it reports that there is no b-coloring with k colors, see Algorithm 1.

Algorithm 1 Algorithm for b-Coloring with $k = m(G)$.

Input : A graph G
Output : A b-coloring with $m(G)$ colors if it exists, No otherwise.

foreach *minimal b-precoloring* $\gamma_X : X \to [k]$ **do**
  foreach *precoloring* $\gamma_{D\setminus X} : (D \setminus X) \to [k]$ **do**
    if $\gamma_{X\cup D} := \gamma_X \cup \gamma_{D\setminus X}$ *is proper* **then return** a greedy extension of $\gamma_{X\cup D}$;
return No;

The correctness of the algorithm follows from the fact that it enumerates all precolorings that can satisfy Claim 12.1. We discuss its runtime. By Lemma 11, we can enumerate all minimal b-precolorings with k colors in time $\beta(k) \cdot k^{\mathcal{O}(1)}$. For each such minimal b-precoloring, we also enumerate all colorings of D. Since $|D| \leq k$, this gives an additional factor of k^k to the runtime which (with $\Delta \leq n$) then amounts to $\beta(k) \cdot k^k \cdot k^{\mathcal{O}(1)} \leq n^{k^2} \cdot 2^{\mathcal{O}(k^2 \log k)}$. ◀

4.3 Algorithm for $k = \Delta(G)$

Next, we turn to the case when $k = \Delta(G)$. Here the strategy is to again enumerate all minimal b-precolorings, and then for each such precoloring we check whether it can be extended to the remainder of the graph. Formally, we use an algorithm for the following problem as a subroutine.

Precoloring Extension (PrExt)

Input: A graph G, an integer k, and a precoloring $\gamma_X : X \to [k]$ of a set $X \subseteq V(G)$
Question: Does G have a proper coloring with k colors extending γ_X?

Naturally, Precoloring Extension is a hard problem, since it includes Graph Coloring as the special case when $X = \emptyset$. However, when $\Delta(G) \leq k - 1$, then the problem is trivially solvable: we simply check if the precoloring at the input is proper and if so, we compute an extension of it greedily. Since each vertex has degree at most $k - 1$, there is always at least one color available. The case when $k = \Delta(G)$ has also been shown to be solvable in polynomial time.

▶ **Theorem 13** (Thm. 3 in [5], see also [7]). *There is an algorithm that solves* Precoloring Extension *in polynomial time whenever $\Delta(G) \leq k$.*

▶ **Theorem 14.** *There is an algorithm that decides whether a graph G has a b-coloring with $\Delta(G)$ colors in time $n^{k+\mathcal{O}(1)} \cdot 2^{\mathcal{O}(k^2 \log k)}$.*

Proof (sketch). For each minimal b-precoloring γ_X, we apply the algorithm for PrExt of Theorem 13. If it finds a proper coloring extending γ_X, we return it, and if there is no successful run of the algorithm for PrExt, we return No. The details are given in the full version [13]. ◀

4.4 Algorithm for $k = m(G) - 1$

Before we proceed to describe the algorithm for b-Coloring when $k = m(G) - 1$, we show that the algorithm of Theorem 13 can be used for a slightly more general case of Precoloring Extension, namely the case when all high-degree vertices in the input instance are precolored.

▶ **Lemma 15** (♣)**.** *There is an algorithm that solves an instance (G, k, γ_X) of* Precoloring Extension *in polynomial time whenever $\max_{v \in V(G) \setminus X} \deg(v) \le k$.*

▶ **Theorem 16.** *There is an algorithm that decides whether a graph G has a b-coloring with $k = m(G) - 1$ colors in time $n^{k^2+\mathcal{O}(1)} \cdot 2^{k^2 \log k}$.*

Proof (sketch). Let D denote the set of vertices of degree at least $k + 1$ in G. By the definition of $m(G)$, we have that $|D| \le k + 1$. We first enumerate all minimal b-precolorings of G, and for each such precoloring, we enumerate all precolorings of D. Since given a b-precoloring γ_X with $D \subseteq X$, we have that every vertex in $V(G) \setminus X$ has degree at most k, we can apply the algorithm of Lemma 15 to verify whether there is a proper coloring of G that extends γ_X. If so, we output that extension. If no such precoloring can be found, then we conclude that we are dealing with a No-instance. We give the details of the algorithm and its correctness proof in the full version [13].

It remains to argue the runtime. We enumerate $\beta(k)$ (see (1)) minimal b-precolorings in time $\beta(k) \cdot k^{\mathcal{O}(1)}$ using Lemma 11. For each such precoloring, we enumerate all precolorings of $D \setminus X$. Since $|D| \le k + 1$, there are at most k^{k+1} such colorings. Finally, we run the algorithm for PrExt due to Lemma 15 which takes time $n^{\mathcal{O}(1)}$. The total runtime becomes $\beta(k) \cdot k^{\mathcal{O}(1)} \cdot k^{k+1} \cdot n^{\mathcal{O}(1)} \le n^{k^2+\mathcal{O}(1)} \cdot 2^{k^2 \log k}$. ◀

5 Maximum Degree Parameterizations

In this section we consider parameterizations of b-Coloring that involve the maximum degree $\Delta(G)$ of the input graph G. In Section 5.1 we show that we can solve b-Coloring when $k = m(G)$ in time FPT parameterized by $\Delta(G)$ and in Section 5.2 we show that b-Coloring is FPT parameterized by $\Delta(G) + \ell_k(G)$.

Both algorithms presented in this section make use of the following reduction rule, which has already been applied in [16, 17] to obtain the FPT algorithm for the problem of deciding whether a graph G has a b-coloring with $k = \Delta(G) + 1$ colors, parameterized by k.

▶ **Reduction Rule 17** ([16, 17])**.** *Let (G, k) be an instance of b-*Coloring*. If there is a vertex $v \in V(G)$ such that every vertex in $N[v]$ has degree at most $k - 2$, then reduce (G, k) to $(G - v, k)$.*

5.1 FPT Algorithm for $k = m(G)$ parameterized by $\Delta(G)$

Sampaio [17] and Panolan et al. [16] independently showed that parameterized by $\Delta(G)$, it can be decided in FPT time whether a graph G has a b-coloring with $\Delta(G) + 1$ colors. In this section we show that in the same parameterization, it can be decided in FPT time whether a graph has a b-coloring with $m(G)$ colors.

▶ **Theorem** (Thm. 2, restated)**.** *There is an algorithm that given a graph G on n vertices decides whether G has a b-coloring with $k = m(G)$ colors in time $2^{\mathcal{O}(k^4 \cdot \Delta)} + n^{\mathcal{O}(1)} \le 2^{\mathcal{O}(\Delta^5)} + n^{\mathcal{O}(1)}$, where $\Delta := \Delta(G)$.*

Proof. We apply Reduction Rule 17 exhaustively to G and consider the following 3-partition (D, T, R) of $V(G)$, where D contains the vertices of degree at least k, T the vertices of degree precisely $k-1$ and R the remaining vertices, i.e. $R := V(G) \setminus (D \cup T)$. Since we applied Reduction Rule 17 exhaustively, we make

▶ **Observation 2.1.** *Every vertex in R has at least one neighbor in $D \cup T$.*

We pick an inclusion-wise maximal set $B \subseteq D \cup T$ such that for each pair of distinct vertices $b_1, b_2 \in B$, we have that $dist(b_1, b_2) \geq 4$.

Case 1 ($|B \cap T| < k$).[1] We show that for any vertex in $u \in V(G) \setminus B$, there is a vertex $v \in B$ such that $dist(u, v) \leq 4$. Suppose $u \in D \cup T$. Since we did not include u in B, it immediately follows that there is some $v \in B$ such that $dist(u, v) < 4$. Now suppose $u \in R$. By Observation 2.1, u has a neighbor w in $D \cup T$ and by the previous argument, there is a vertex $v \in B$ such that $dist(w, v) < 4$. We conclude that $dist(u, v) \leq 4$. Using this observation, we now show that in this case, the number of vertices in G is polynomial in k and Δ.

▶ **Claim 2.2.** *If $|B \cap T| < k$, then $|V(G)| \leq \mathcal{O}(k^4 \cdot \Delta)$.*

Proof. Note that $(B \cup D, S_1, \ldots, S_4)$ constitutes a partition of $V(G)$, where S_i is the set of vertices of $V(G) \setminus (B \cup D)$ that are at distance exactly i from B. Since $|B \cap T| < k$ and $|D| \leq k$, we have that $|B \cup D| < 2k$, and therefore $|S_1| < 2k \cdot \Delta$. By the definition of $m(G)$, all the vertices in $S_1 \cup \ldots \cup S_4$ have degree at most $k-1$. This implies that $|S_i| < (k-1)^{i-1} \cdot 2k \cdot \Delta$. We conclude that the number of vertices in G is at most $2k + 2k \cdot \Delta \cdot \sum_{i=1}^{4} (k-1)^{i-1} = \mathcal{O}(k^4 \cdot \Delta)$. ◁

By Claim 2.2, we can solve the instance in Case 1 in time $2^{\mathcal{O}(k^4 \cdot \Delta)}$ using the algorithm of Panolan et al. [16].

Case 2 ($|B \cap T| \geq k$). Let $B' \subseteq B \cap T$ with $|B'| = k$ and denote this set by $B' = \{x_1, x_2, \ldots, x_k\}$. We show that we can construct a b-coloring $\gamma \colon V(G) \to [k]$ of G such that for $i \in [k]$, x_i is the b-vertex of color i. For $i \in [k]$, we let $\gamma(x_i) := i$. Next, we color the vertices in D. Recall that $|D| \leq k$, so we can color the vertices in D injectively with colors from $[k]$, ensuring that this will not create a conflict on any edge in $G[D]$. Furthermore, consider $i, j \in [k]$ with $i \neq j$. Since $dist(x_i, x_j) \geq 4$, we have that $N(x_i) \cap N(x_j) = \emptyset$. In particular, there is no vertex in D that has two or more neighbors in B'. To summarize, we can conclude that we can let γ color the vertices of D in such a way that:

(i) γ is injective on D, and
(ii) γ is a proper coloring of $G[B' \cup D]$.

These two items imply that for each x_i ($i \in [k]$), its neighbors $N(x_i) \cap D$ receive distinct colors which are also different from i. Let $\ell := |N(x_i) \cap D|$. It follows that we can let γ color the remaning $(k-1) - \ell$ neighbors of x_i in an arbitrary bijective manner with the $(k-1) - \ell$ colors that do not yet appear in the neighborhood of x_i.

After this process, x_i is a b-vertex for color i. We proceed in this way for all $i \in [k]$. Since for $i, j \in [k]$ with $i \neq j$ we have that $dist(x_i, x_j) \geq 4$, it follows that there are no edges between $N[x_i]$ and $N[x_j]$ in G. Hence, we did not introduce any coloring conflict in the previous step. Now, all vertices in G that have not yet received a color by γ have degree at most $k-1$, so we can extend γ to a proper coloring of G in a greedy fashion.

We summarize the whole procedure in Algorithm 2. We now analyze its runtime. Clearly,

[1] This case is almost identical to [16, Case II in the proof of Theorem 2].

Algorithm 2 An algorithm that either constructs a b-coloring of a graph G with $m(G)$ colors, or reports that there is none, and runs in FPT time parameterized by $\Delta(G)$.

Input : A graph G with $k = m(G)$ `// More generally, graph` G `with` $\ell_k(G) \leq k$
Output : A b-coloring with k colors of G if it exists, and NO otherwise.

Apply Reduction Rule 17 exhaustively;
Let (D, T, R) be a partition of $V(G)$ such that for all $x \in D$, $\deg_G(x) \geq k$, for all $x \in T$, $\deg_G(x) = k - 1$, and $R = V(G) \setminus (D \cup T)$;
Let $B \subseteq D \cup T$ be a maximal set such that for distinct $b_1, b_2 \in B$, $dist(b_1, b_2) \geq 4$;
if $|B \cap T| < k$ **then** `// Case 1`
  Solve the instance in time $2^{\mathcal{O}(k^4 \cdot \Delta)}$ using the b-COLORING algorithm [16];
  if *the algorithm of [16] returned a b-coloring γ* **then return** γ;
  else return NO;
else `// Case 2, i.e.` $|B \cap T| \geq k$
  Pick a size-k subset of $B \cap T$, say $B' := \{x_1, \ldots, x_k\}$;
  Initialize a k-coloring $\gamma \colon V(G) \to [k]$;
  For $i \in [k]$, let $\gamma(x_i) := i$;
  Let γ color the vertices of D injectively such that γ remains proper on $G[B' \cup D]$;
  For $i \in [k]$, let γ color $N(x_i) \cap D$ such that x_i is the b-vertex of color i;
  Extend the coloring γ greedily to the remainder of G;
  return γ;

exhaustively applying Reduction Rule 17 can be done in time $n^{\mathcal{O}(1)}$. As mentioned above, Case 1 can be solved in time $2^{\mathcal{O}(k^4 \cdot \Delta)}$. In Case 2, the coloring of $G[B' \cup D]$ can be found in time $\mathcal{O}(k^2)$, and extending the coloring to the remainder of G can be done in time $n^{\mathcal{O}(1)}$. The claimed bound follows. ◀

▶ Remark 18 (♣). Algorithm 2 solves the problem of deciding whether G admits a b-coloring with k colors in time $2^{\mathcal{O}(k^4 \cdot \Delta)} + n^{\mathcal{O}(1)}$ whenever $\ell_k(G) \leq k$.

Furthermore, in the proof of Theorem 2, we provide a polynomial kernel for the problem: In Case 1, we have a kernelized instance on $\mathcal{O}(k^4 \cdot \Delta)$ vertices (see Claim 2.2) and in Case 2, we always have a YES-instance.

▶ **Corollary 19.** *The problem of deciding whether a graph G has a b-coloring with $k = m(G)$ colors admits a kernel on $\mathcal{O}(k^4 \cdot \Delta) = \mathcal{O}(\Delta^5)$ vertices.*

5.2 FPT Algorithm Parameterized by $\Delta(G) + \ell_k(G)$

The next parameterization of b-COLORING involving the maximum degree that we consider is by $\Delta(G) + \ell_k(G)$. We show that in this case, the problem is FPT. By Observation 6 we know that b-COLORING is NP-complete on graphs with $\ell_k(G) = 0$, and by Theorem 1, it is NP-complete even when $k = 3$ and $\Delta(G) = 4$. Hence, there is no FPT- nor XP-algorithm for a parameterization using only one of the two above mentioned parameters unless P = NP. Note that the algorithm we provide in this section can be used to solve the case of $k = m(G)$ for which we gave a separate algorithm in Section 5.1, see Algorithm 2. However, Algorithm 2 is much simpler than the algorithm presented in this section, and simply applying the following algorithm for the case $k = m(G)$ results in a runtime of $2^{\mathcal{O}(k^{k+3} \cdot \Delta)} + n^{\mathcal{O}(1)}$ which is far worse than the runtime of $2^{\mathcal{O}(k^4 \cdot \Delta)} + n^{\mathcal{O}(1)}$ of Theorem 2.

▶ **Theorem** (Thm. 3, restated). *There is an algorithm that given a graph G on n vertices decides whether G has a b-coloring with k colors in time $2^{\mathcal{O}(\ell\cdot\Delta\cdot\min\{\ell,\Delta\}^{\ell+2})} + n^{\mathcal{O}(1)}$, where $\Delta := \Delta(G)$ and $\ell := \ell_k(G)$.*

Proof. The overall strategy of the algorithm is similar to Algorithm 2. We can make the following assumptions. First, if $\ell \le k$, then we can apply Algorithm 2 directly to solve the instance at hand, see Remark 18. Hence we can assume that $k < \ell$. Furthermore, $k \le \Delta + 1$, otherwise we are dealing with a trivial No-instance; we have that $k \le \min\{\ell - 1, \Delta + 1\}$. Furthermore, we can assume that $k > 2$, otherwise the problem is trivially solvable in time polynomial in n.

We consider a partition (D, T, R) of $V(G)$, where the vertices in D have degree at least k, the vertices in T have degree $k - 1$ and the vertices in R have degree less than $k - 1$. We assume that Reduction Rule 17 has been applied exhaustively, so Observation 2.1 holds, i.e. every vertex in R has at least one neighbor in $D \cup T$.

Now, we pick an inclusion-wise maximal set $B \subseteq D \cup T$ such that for each pair of distinct vertices $b_1, b_2 \in B$, $dist_G(b_1, b_2) \ge \ell + 3$.

Case 1 ($|B \cap T| < k$). By the same argument given in Case 1 of the proof of Theorem 2, we have that any vertex in $T \cup R$ is at distance at most $\ell + 3$ from a vertex in B. We now give a bound on the number of vertices in G in terms of ℓ and Δ.

▶ **Claim 3.1** (♣). *If $|B \cap T| < k$, then $|V(G)| = \mathcal{O}(\ell \cdot \Delta \cdot \min\{\ell, \Delta\}^{\ell+2})$.*

By the previous claim, we can solve the instance in time $2^{\mathcal{O}(\ell\cdot\Delta\cdot\min\{\ell,\Delta\}^{\ell+2})}$ in this case, using the exact exponential time algorithm for b-Coloring due to Panolan et al. [16].

Case 2 ($|B \cap T| \ge k$). Let $B' \subseteq B \cap T$ be of size k and denote it by $B' := \{x_1, \ldots, x_k\}$. The strategy in this case is as follows: We compute a proper coloring of $G[D]$, and then modify it so that can be extended to a b-coloring of G. In this process we will be able to guarantee for each $i \in [k]$, that either x_i can be the b-vertex for color i, or we will have found another vertex in D that can serve as the b-vertex of color i. The difficulty here arises from the following situation: Suppose that in the coloring we computed for $G[D]$, a vertex x_i has two neighbors in D that received the same color. Then, x_i cannot be the b-vertex of color i in any extension of that coloring, since $\deg(x_i) = k - 1$, and $k - 1$ colors need to appear the neighborhood of x_i for it to be a b-vertex. However, recoloring a vertex in $N(x_i) \cap D$ might create a conflict in the coloring of $G[D]$. These potential conflicts can only appear in the connected component of $G[D \cup B']$ that contains x_i. We now show that each component of $G[D \cup B']$ can contain at most one such vertex, by our choice of the set B.

▶ **Claim 3.2** (♣). *Let C be a connected component of $G[D \cup B']$. Then, C contains at most one vertex from B'.*

Throughout the following, for $i \in [k]$, we denote by C_i the connected component of $G[D \cup B']$ that contains x_i, and by ℓ_i the number of vertices of C_i, i.e. $\ell_i := |V(C_i)|$. By Claim 3.2, $C_i \neq C_j$, for all $i, j \in [k], i \neq j$. We now show that each neighbor of x_i has no neighbor in $D \cap N[B']$ outside of $V(C_i) \cup N[x_i]$.

▶ **Claim 3.3** (♣). *Let $i \in [k]$, and $y \in N(x_i) \setminus D$. Then, $N_G[y] \cap (D \cup N[B']) \subseteq V(C_i) \cup N[x_i]$.*

Let $\mathcal{C}_\emptyset$ be the set of connected components of $G[D \cup B']$ that do not contain any vertex from B'. We observe that any proper coloring of $G[D \cup B']$ can be obtained from independently coloring the vertices in $C_1, \ldots, C_k$, and $\mathcal{C}_\emptyset$. If for some $i \in [k]$, C_i is a trivial[2] component,

[2] We call a connected component of a graph *trivial* if it contains only one vertex.

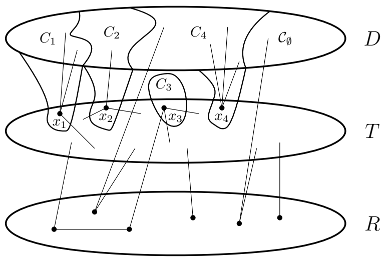

Figure 1 Illustration of the structure of a graph G in the proof of Theorem 3 where $k = 4$. Here, $B' = \{x_1, \ldots, x_4\}$ and $C_1, \ldots, C_4$ are the components of $G[D \cup B']$ containing $x_1, \ldots, x_4$, respectively. Note that all vertices in T are of degree 3, all vertices in R of degree at most 2 and all vertices in R have a neighbor in $D \cup T$.

then $N(x_i) \cap D = \emptyset$. Hence, we can assign x_i any color without creating any conflict with the remaining vertices in $G[D \cup B']$. On top of that, Claim 3.3 ensures that assigning a color to a neighbor of any x_i (that is not contained in D) cannot create a coloring conflict with any vertex in $D \cup N[B']$ that is not contained in $V(C_i) \cup N[x_i]$. We illustrate the structure of G in Figure 1.

▶ **Claim 3.4 (♣).** *Let $i \in [k]$ and let $\gamma\colon V(C_i) \to [k]$ be a proper coloring of C_i. Then, one can find in time $\mathcal{O}(k^2 \cdot \ell_i^2)$ a set $Y_i \subseteq N_G(x_i) \setminus D$ and a proper coloring $\delta\colon V(C_i) \cup Y_i \to [k]$ of $G[V(C_i) \cup Y_i]$ that has a b-vertex for color i.*

We now wrap up the treatment of this case. We compute a proper k-coloring γ of $G[D \cup B']$ using standard methods [3]. We derive from γ another k-coloring δ of some induced subgraph of $G[D \cup N_G[B']]$ containing $D \cup B'$. For each $i \in [k]$, we do the following. With input $\gamma|_{V(C_i)}$ we compute a proper k-coloring δ_i of $G[V(C_i) \cup Y_i]$ using Claim 3.4, where Y_i is the set returned by its algorithm, and we let $\delta|_{V(C_i) \cup Y_i} := \delta_i$. Finally, we let $\delta|_{V(\mathcal{C}_\emptyset)} := \gamma|_{V(\mathcal{C}_\emptyset)}$. As for $i \neq j$, C_i and C_j are distinct connected components of $G[D \cup B']$ and by Claim 3.3, this construction is well-defined and there is no color conflict between any pair of vertices z_i, z_j where $z_i \in V(C_i) \cup Y_i$ and $z_j \in V(C_j) \cup Y_j$ for $i \neq j$. Since for each $i \in [k]$ we applied Claim 3.4, δ is a b-precoloring of G. All vertices in G that have not received a color so far (recall that δ colors all vertices in D) have degree at most $k - 1$, so we can extend the coloring δ greedily to the remainder of G and eventually obtain a b-coloring of G. The runtime of $2^{\mathcal{O}(\ell \cdot \Delta \cdot \min\{\ell, \Delta\}^{\ell+2})} + n^{\mathcal{O}(1)}$ is argued for in the full version [13]. ◀

Similar to above, we obtained a kernel for the problem. While this result does not provide a polynomial kernel for the parameterization $\Delta + \ell$, it does give a polynomial kernel if we consider the problem for *fixed* values of ℓ and parameter Δ.

▶ **Corollary 20.** *The problem of deciding whether a graph G admits a b-coloring with k colors admits a kernel on $\mathcal{O}(\ell \cdot \Delta \cdot \min\{\ell, \Delta\}^{\ell+2})$ vertices, where $\Delta := \Delta(G)$ and $\ell := \ell_k(G)$.*

6 Conclusion

We have presented a complexity dichotomy for b-Coloring with respect to two upper bounds on the b-chromatic number, in the following sense: We have shown that given a graph G and for fixed $k \in \{\Delta(G) + 1 - p, m(G) - p\}$, it can be decided in polynomial time whether G has a b-coloring with k colors whenever $p \in \{0, 1\}$ and the problem remains NP-complete whenever $p \geq 2$, already for $k = 3$.

The most immediate question left open in this work is the parameterized complexity of the b-Coloring problem when $k \in \{m(G), \Delta(G), m(G) - 1\}$. In all of these cases, we have provided XP-algorithms, and it would be interesting to see whether these problems are FPT or W[1]-hard. We showed that b-Coloring is FPT parameterized by $\Delta(G) + \ell_k(G)$, where $\ell_k(G)$ denotes the number of vertices of degree at least k in G, and this is optimal in the sense that there is no FPT nor XP algorithm for the problem parameterized by only one of the two invariants. It would be interesting to see if one could devise an FPT-algorithm for the parameterization that replaces the maximum degree by the number of colors.

References

1 Pierre Aboulker, Nick Brettell, Frédéric Havet, Dániel Marx, and Nicolas Trotignon. Coloring Graphs with Constraints on Connectivity. *J. Graph Theory*, 85(4):814–838, 2017.

2 Dominique Barth, Johanne Cohen, and Taoufik Faik. On the b-continuity property of graphs. *Discrete Appl. Math.*, 155:1761–1768, 2007.

3 Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set Partitioning via Inclusion-Exclusion. *SIAM J. Comput.*, 39(2):546–563, 2009.

4 Victor Campos, Carlos Vinicius G. C. Lima, and Ana Silva. Graphs of girth at least 7 have high b-chromatic number. *Eur. J. Combin.*, 48:154–164, 2015.

5 Miroslav Chelbík and Janka Chlebíkova. Hard Coloring Problems in Low Degree Planar Bipartite Graphs. *Discrete Appl. Math.*, 154:1960–1965, 2006.

6 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 1st edition, 2015.

7 Konrad K. Dabrowski, François Dross, Matthew Johnson, and Daniël Paulusma. Filling the complexity gaps for colouring planar and bounded degree graphs. In *Proc. IWOCA '15*, volume 9538 of *Lecture Notes in Computer Science*, pages 100–111. Springer, 2015.

8 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

9 Esther Galby, Paloma T. Lima, Daniël Paulusma, and Bernard Ries. On the Parameterized Complexity of k-Edge Colouring, 2019. arXiv:1901.01861.

10 Frédéric Havet, Cláudia Linhares Sales, and Leonardo Sampaio. b-Coloring of Tight Graphs. *Discrete Appl. Math.*, 160:2709–2715, 2012.

11 Frédéric Havet and Leonardo Sampaio. On the Grundy and b-chromatic numbers of a graph. *Algorithmica*, 65(4):885–899, 2013.

12 Robert W. Irving and David F. Manlove. The b-Chromatic Number of a Graph. *Discrete Appl. Math.*, 91(1-3):127–141, 1999.

13 Lars Jaffke and Paloma T. Lima. A Complexity Dichotomy for Critical Values of the b-Chromatic Number of Graphs. *CoRR*, 2018. arXiv:1811.03966.

14 Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

15 Jan Kratochvíl, Zsolt Tuza, and Margit Voigt. On the b-Chromatic Number of Graphs. In *Proc. WG '02*, volume 2573 of *LNCS*, pages 310–320, 2002.

16 Fahad Panolan, Geevarghese Philip, and Saket Saurabh. On the parameterized complexity of b-Chromatic Number. *J. Comput. Syst. Sci.*, 84:120–131, 2017.

17 Leonardo Sampaio. *Algorithmic Aspects of Graph Coloring Heuristics*. PhD thesis, Université Nice Sophia Antipolis, France, 2012.