Breaking the 2^n Barrier for 5-Coloring and 6-Coloring

Or Zamir
Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract

The coloring problem (i.e., computing the chromatic number of a graph) can be solved in $O^*(2^n)$ time, as shown by Björklund, Husfeldt and Koivisto in 2009. For $k = 3, 4$, better algorithms are known for the k-coloring problem. 3-coloring can be solved in $O(1.33^n)$ time (Beigel and Eppstein, 2005) and 4-coloring can be solved in $O(1.73^n)$ time (Fomin, Gaspers and Saurabh, 2007). Surprisingly, for $k > 4$ no improvements over the general $O^*(2^n)$ are known. We show that both 5-coloring and 6-coloring can also be solved in $O((2-\varepsilon)^n)$ time for some $\varepsilon > 0$. As a crucial step, we obtain an exponential improvement for computing the chromatic number of a very large family of graphs. In particular, for any constants $\Delta, \alpha > 0$, the chromatic number of graphs with at least $\alpha \cdot n$ vertices of degree at most Δ can be computed in $O((2-\varepsilon)^n)$ time, for some $\varepsilon = \varepsilon_{\Delta,\alpha} > 0$. This statement generalizes previous results for bounded-degree graphs (Björklund, Husfeldt, Kaski, and Koivisto, 2010) and graphs with bounded average degree (Golovnev, Kulikov and Mihajlin, 2016). We generalize the aforementioned statement to List Coloring, for which no previous improvements are known even for the case of bounded-degree graphs.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms; Theory of computation → Graph algorithms analysis

Keywords and phrases Algorithms, Graph Algorithms, Graph Coloring

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.113

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2007.10790

Acknowledgements The author would like to deeply thank Noga Alon for important discussions and insights regarding the subset removal lemma, and Haim Kaplan and Uri Zwick for many helpful discussions and comments on the paper. The author would also like to thank anonymous reviewers for helpful comments.

1 Introduction

The problem of k-coloring a graph, or determining the *chromatic number* of a graph (i.e., finding the smallest k for which the graph is k-colorable) is one of the most classic and well studied NP-Complete problems. Computing the chromatic number is listed as one of the first NP-Complete problems in Karp's paper from 1972 [17]. In a similar fashion to k-SAT, the problem of 2-coloring is polynomial, yet k-coloring is NP-complete for every $k \geqslant 3$ (proven independently by Lovász [22] and Stockmeyer [30]). An algorithm solving 3-coloring in sub-exponential time would imply, via the mentioned reductions, that 3-SAT can also be solved in sub-exponential time. It is strongly believed that this is not possible (as stated in a widely believed conjecture called *The Exponential Time Hypothesis* [15]), and thus it is believed that exact algorithms solving k-coloring must be exponential.

There is a substantial and ever-growing body of work exploring exponential-time worst-case algorithms for NP-Complete problems. A 2003 survey of Woeginger [31] covers and refers to dozens of papers exploring such algorithms for many problems including satisfiability, graph coloring, knapsack, TSP, maximum independent sets and more. Subsequent review article of Fomin and Kaski [10] and book of Fomin and Kratsch [11] further cover the topic of exact exponential-time algorithms.

For satisfiability (commonly abbreviated as SAT), the running time of the trivial algorithm enumerating over all possible assignments is $O^*(2^n)$. No algorithms solving SAT in time $O^*\left((2-\varepsilon)^n\right)$ for any $\varepsilon > 0$ are known, and a popular conjecture called The *Strong* Exponential Time Hypothesis [5] states that no such algorithm exists. On the other hand, it is known that for every fixed k there exists a constant $\varepsilon_k > 0$ such that k-SAT can be solved in $O^*\left((2-\varepsilon_k)^n\right)$ time. A result of this type was first published by Monien and Speckenmeyer in 1985 [23]. A long list of improvements for the values of ε_k were published since, including the celebrated 1998 PPSZ algorithm of Paturi, Pudlák, Saks and Zane [25] and the recent improvement over it by Hansen, Kaplan, Zamir and Zwick [13].

For coloring, on the other hand, the situation is less understood. The trivial algorithm solving k-coloring by enumerating over all possible colorings takes $O^*(k^n)$ time. Thus, it is not even immediately clear that computing the chromatic number of a graph can be done in $O^*(c^n)$ time for a constant c independent of k. In 1976, Lawler [21] introduced the idea of using dynamic-programming to find the minimal number of independent sets covering the graph. The trivial implementation of this idea results in an $O^*(3^n)$ algorithm. More sophisticated bounds on the number of maximal independent sets in a graph and fast algorithms to enumerate over them (Moon and Moser [24], Paull and Unger [26]) resulted in an $O^*(2.4422^n)$ algorithm. This was improved several times (including Eppstein [8] and Byskov [4]), until finally an algorithm computing the chromatic number in $O^*(2^n)$ time was devised by Björklund, Husfeldt and Koivisto in 2009 [3]. This settled an open problem of Woeginger [31]. A relatively recent survey of Husfeldt [14] covers the progress on graph coloring algorithms.

For $k = 3, 4$, better algorithms are known for the k-coloring problem. Schiermeyer [28] showed that 3-coloring can be solved in $O^*(1.415^n)$ time. Beigel and Eppstein [1] gave algorithms solving 3-coloring in $O^*(1.3289^n)$ time and 4-coloring in $O^*(1.8072^n)$ time in 2005. Fomin, Gaspers and Saurabh [9] have improved the running time of 4-coloring to $O^*(1.7272^n)$ in 2007. Unlike the situation in k-SAT, for every $k > 4$ the best known running time for k-coloring is $O^*(2^n)$, the same as computing the chromatic number. Thus, a very fundamental question was left wide open.

▶ **Open Problem 1.** *Can 5-coloring be solved in $O^*\left((2-\varepsilon)^n\right)$ time, for some $\varepsilon > 0$?*

More generally,

▶ **Open Problem 2.** *Can k-coloring be solved in $O^*\left((2-\varepsilon_k)^n\right)$ time, for some $\varepsilon_k > 0$, for every k?*

In our work, we answer Problem 1 affirmatively, the answer extends to 6-coloring as well. We also make steps towards settling Problem 2.

The main technical theorem of our paper follows.

▶ **Definition 3.** *For $0 \leqslant \alpha \leqslant 1$ and $\Delta > 0$ we say that a graph $G = (V(G), E(G))$ is* (α, Δ)-bounded *if it contains at least $\alpha \cdot |V(G)|$ vertices of degree at most Δ.*

▶ **Theorem 4.** *For every $\Delta, \alpha > 0$ there exists $\varepsilon_{\Delta,\alpha} > 0$ such that we can compute the chromatic number of (α, Δ)-bounded graphs in $O\left((2-\varepsilon_{\Delta,\alpha})^n\right)$ time.*

In other words, we can answer Problem 2 affirmatively unless the graph has almost only vertices of super-constant degrees. This theorem generalizes a few previous results. A similar statement for the restricted case of bounded degree graphs was obtained by Björklund et al. in [2]. Golovnev, Kulikov and Mihajlin [12] used a variant of FFT to get an algorithmic improvement for computing the chromatic number of graphs with bounded average degree. Prior to that, Cygan and Pilipczuk [7] obtained an improvement for the running time required for the Traveling Salesman Problem for graphs with bounded average degree.

It is important to stress that Theorem 4 is much more general than the mentioned results. In particular, (α, Δ)-bounded graphs may have $\Omega(n^2)$ edges (and in turn average degree $\Omega(n)$). The techniques used for graphs with bounded average degree do not extend to this case. Another important difference is that our algorithms extend to finding a coloring of the graph while the mentioned ones solve the decision problem, as later discussed in Section 2.2. Moreover, the generality of Theorem 4 is crucial for our derivation of the reductions resulting in the improvements for 5 and 6 coloring.

In the List Coloring problem (defined formally in Section 2), we are given a graph and lists C_v of colors for each vertex v, and are asked to find a valid coloring of the graph such that each vertex v is colored by some color appearing in its list C_v. In the k-list-coloring problem each list C_v is of size at most k.

List Coloring can also be solved in $O^*(2^n)$ time [3], yet no improvements were known even for the bounded-degree case. We extend Theorem 4 to k-list-coloring, for any constant k, as follows.

▶ **Theorem 5.** *For every $k, \Delta, \alpha > 0$ there exists $\varepsilon_{k,\Delta,\alpha} > 0$ such that we can solve k-list-coloring for (α, Δ)-bounded graphs in $O\left((2-\varepsilon_{k,\Delta,\alpha})^n\right)$ time, regardless of the size of the universe of colors.*

Using Theorem 4 as a crucial component, we devise the following reductions and corollaries.

▶ **Theorem 6.** *Given an algorithm solving $(k-1)$-list-coloring in time $O\left((2-\varepsilon)^n\right)$ for some constant $\varepsilon > 0$, we can construct an algorithm solving k-coloring in time $O\left((2-\varepsilon')^n\right)$ for some (other) constant $\varepsilon' > 0$. Furthermore, the reduction is deterministic.*

▶ **Theorem 7.** *Given an algorithm solving $(k-2)$-list-coloring in time $O\left((2-\varepsilon)^n\right)$ for some constant $\varepsilon > 0$, we can construct an algorithm solving k-coloring with high probability in time $O\left((2-\varepsilon')^n\right)$ for some (other) constant $\varepsilon' > 0$.*

From which we finally conclude the following, answering Problem 1 affirmatively.

▶ **Theorem 8.** *5-coloring can be solved in time $O\left((2-\varepsilon)^n\right)$ for some constant $\varepsilon > 0$.*

▶ **Theorem 9.** *6-coloring can be solved with high probability in time $O\left((2-\varepsilon)^n\right)$ for some constant $\varepsilon > 0$.*

We note that our 5-coloring algorithm is deterministic, while our 6-coloring algorithm is randomized with an exponentially small one-sided error probability.

As part of our work, we develop a new removal lemma for small subsets. This could be of independent interest. Very roughly, it states that every collection of small sets must have a large sub-collection that can be made pairwise-disjoint by the removal of a small subset of the universe. The exact statement follows.

▶ **Theorem 10.** *Let $\mathcal{F}$ be a collection of subsets of a universe U such that every set $F \in \mathcal{F}$ is of size $|F| \leqslant \Delta$. Let $C > 0$ be any constant. Then, there exist subsets $\mathcal{F}' \subseteq \mathcal{F}$ and $U' \subseteq U$, such that*

- *$|\mathcal{F}'| > \rho(\Delta, C) \cdot |\mathcal{F}| + C \cdot |U'|$, where $\rho(\Delta, C) > 0$ depends only on Δ, C.*
- *The sets in $\mathcal{F}'$ are disjoint when restricted to $U \backslash U'$, i.e., for every $F_1, F_2 \in \mathcal{F}'$ we have $F_1 \cap F_2 \subseteq U'$.*

In the full version of the paper we present an upper bound for the function ρ appearing in Theorem 10. This upper bound implies that the constant ε we can obtain using our technique must be very small.

2 Preliminaries

The terminology used throughout the paper is standard. For a graph G we denote by $V(G)$ and $E(G)$ its vertex-set and edge-set, respectively. Throughout the paper, n is used to denote $|V(G)|$. For a subset $V' \subseteq V(G)$ we denote by $G[V']$ the sub-graph of G induced by V'. For $v \in V$ we denote by $\deg(v)$ the degree of v in G, by $N(v)$ the set of neighbours of v, and by $N[v] := N(V) \cup \{v\}$.

For $0 \leqslant \alpha \leqslant 1$ and $\Delta > 0$ we say that a graph $G = (V(G), E(G))$ is (α, Δ)-*bounded* if it contains at least $\alpha \cdot |V(G)|$ vertices of degree at most Δ. Note that if $\alpha = 1$ this definition coincides with the standard definition of a bounded degree graph.

In the k-*coloring problem*, we are given a graph G and need to decide whether there exists a k-coloring $c : V(G) \to [k]$ of G, such that for every $(u, v) \in E(G)$ we have $c(u) \neq c(v)$. If a graph has a k-coloring, we say that it is k-colorable. In the *chromatic number problem*, we are given a graph G and need to compute $\chi(G)$, the minimal integer k for which G is k-colorable.

In the k-*list-coloring problem*, we are given a graph G and a set $C_v \subseteq U$ of size $|C_v| \leqslant k$ for every $v \in V(G)$, where U is some arbitrary universe. We need to decide whether there exists a coloring $c : V(G) \to U$ such that for every $v \in V(G)$ we have $c(v) \in C_v$ and for every $(u, v) \in E(G)$ we have $c(u) \neq c(v)$.

In a general (a, b)-CSP (Constraint Satisfaction Problem, see [20] or [29] for a complete definition and discussions) we are given a list of *constraints*[1] on the values of subsets of size b of n a-ary variables, and need to decide whether there exists an assignment of values to the variables for which all constraints are satisfied. k-coloring and k-list-coloring are examples of $(k, 2)$-CSP problems. k-SAT is an example of a $(2, k)$-CSP problem.

2.1 Inverse Möbius Transform

Let U be an n-element set. The *Inverse Möbius* transform (sometimes also called the *Zeta* transform) [27] maps a function $f : P(U) \to \mathbb{R}$ from the power-set of U into another function $\hat{f} : P(U) \to \mathbb{R}$ defined as

$$\hat{f}(X) = \sum_{Y \subseteq X} f(Y).$$

Naively, $\hat{f}(X)$ is computed using $2^{|X|}$ additions. Thus, we can compute all values of $\hat{f}$ in a straightforward manner with $O(3^n)$ operations. Yates' method from 1937 ([19, 32]) improves on the above and computes all values of $\hat{f}$ using just $O(n2^n)$ operations. The resulting algorithm is usually called *the fast möbius transform* or *the fast zeta transform* ([3, 18]). The authors of [2] and [3] use the fast Inverse Möbius Transform to devise algorithms for combinatorial optimization problems such as computing the chromatic and the domatic numbers of a graph. The algorithm of [3] is summarized in Section 3.

A description of Yates' method follows.

[1] A general constraint on a set $x_1, \ldots, x_b$ of a-ary variables is a subset T of the a^b possible assignments in $\{x_1, \ldots x_b\} \to [a]$. The constraint is satisfied by an assignment c, possibly on more variables, if $c|_{\{x_1, \ldots x_b\}} \in T$.

▶ **Lemma 11.** *The Inverse Möbius Tranform $\hat{f}$ for some function $f : P(U) \to \mathbb{R}$ can be computed in $O(n2^n)$ time, where $n := |U|$.*

Proof. Denote by $U = \{u_1, \ldots, u_n\}$ some enumeration of U's elements. Denote by $f_0 := f$. We preform n iterations for $i = 1, \ldots, n$, in which we compute all values of the function $f_i : P(U) \to \mathbb{R}$ defined using f_{i-1} as follows.

$$f_i(X) = \begin{cases} f_{i-1}(X) + f_{i-1}(X \backslash \{u_i\}) & \text{if } u_i \in X \\ f_{i-1}(X) & \text{otherwise} \end{cases}$$

Namely, in the i-th iteration we add the values the function gets in the sub-cube defined by $u_i = 0$ to the corresponding values in the sub-cube defined by $u_i = 1$.

A simple induction on i shows that $f_i(X) = \sum_{Y \in S_i(X)} f(Y)$ where $S_i(X)$ is the set of all subsets $Y \subseteq X$ such that

$$\{u_j \in Y \mid j > i\} = \{u_j \in X \mid j > i\}$$

In particular, by the end of the algorithm $f_n = \hat{f}$. ◀

2.2 Decision versus Search

The k-coloring problem can be stated in two natural ways. In the first, given a graph G decide whether it can be colored using k colors. The second, given a graph G return a k-coloring for it if one exists, or say that no such coloring exists. A few folklore reductions show that the two problems have the same running time up to polynomial factors. We state one for completeness. Others appear in the survey of [14].

▶ **Lemma 12.** *Let $\mathcal{A}$ be an algorithm deciding whether a graph is k-colorable in $O(T(n))$ time. Then, there exists an algorithm $\mathcal{A}'$ that finds a k-coloring for G, if one exists, in $O^*(T(n))$ time.*

Proof. We describe $\mathcal{A}'$. First, use $\mathcal{A}(G)$ to decide whether G is k-colorable, if it returns *False* we return that no k-coloring exists. Otherwise, repeat the following iterative process. For every pair of distinct vertices $(u, v) \notin E(G)$ that is not an edge of G, use $\mathcal{A}(G' := (V(G), E(G) \cup \{(u, v)\}))$ to check whether G stays k-colorable after adding (u, v) as an edge. If it does, add (u, v) to $E(G)$. We stop when no such pair (u, v) exists.

The reader can verify that the resulting graph must be a complement of k disjoint cliques, and thus we can easily construct a k-coloring. ◀

A problem comes up while trying to use this type of reductions in the settings of this paper. The aforementioned reduction adds edges to the graph, and therefore increases the degrees of vertices. In particular, we cannot use it (or other similar reductions) in a black-box manner for statements like Theorem 4. The algorithm of [2] solves the decision version of k-coloring for bounded degree graphs, and cannot be trivially converted into an algorithm that finds a coloring. The algorithms presented in this paper, on the other hand, can be easily converted into algorithms that find a k-coloring. This is briefly discussed later in Section 4.4.

3 Overview of the $O^*(2^n)$ algorithm

In this section we present a summary of Björklund, Husfeldt and Koivisto's algorithm from [3]. We present a concise variant of their work that applies specifically to the coloring problem. The original paper covers a larger variety of set partitioning problems and thus the description in this section is simpler.

We begin by making the following very simple observation, yielding an equivalent phrasing of the coloring problem.

▶ **Observation 13.** *A graph G is k-colorable if and only if its vertex set $V(G)$ can be* covered *by k independent sets.*

A short outline of the algorithm follows, complete details appear below. We need to decide whether $V(G)$ can be covered by k independent sets. In order to do so, we compute the number of independent sets in every induced sub-graph and then use a simple inclusion-exclusion argument in order to compute the number of (ordered) covers of $V(G)$ by k independent sets. We are interested in whether this number is positive.

▶ **Definition 14.** *For a subset $V' \subseteq V(G)$ of vertices, let $i(G[V'])$ denote the number of independent sets in the induced sub-graph $G[V']$.*

We next show that using dynamic programming, we can quickly compute these values.

▶ **Lemma 15.** *We can compute the values of $i(G[V'])$ for all $V' \subseteq V$ in $O^*(2^n)$ time.*

Proof. Let $v \in V'$ be an arbitrary vertex contained in V'. The number of independent sets in V' that do not contain v is exactly $i(G[V' \backslash \{v\}])$. On the other hand, the number of independent sets in V' that do contain v is exactly $i(G[V' \backslash N[v]])$. Thus, we have

$$i(G[V']) = i(G[V' \backslash \{v\}]) + i(G[V' \backslash N[v]]).$$

We note that both $V' \backslash \{v\}$ and $V' \backslash N[v]$ are of size strictly less than $|V'|$. Thus, we can compute all 2^n values of $i(G[\cdot])$ using dynamic programming processing the sets in non-decreasing order of size. ◀

Consider the expression

$$F(G) = \sum_{V' \subseteq V(G)} (-1)^{|V(G)|-|V'|} \cdot i(G[V'])^k.$$

Using the values of $i(G[\cdot])$ computed in Lemma 15, we can easily compute the value of $F(G)$ by directly evaluating the above expression in $O^*(2^n)$ time.

▶ **Lemma 16.** *Let $S_1 \subseteq S_2$ be sets. It holds that*

$$\sum_{S_1 \subseteq S \subseteq S_2} (-1)^{|S|} = \begin{cases} 0 & \text{if } S_1 \neq S_2 \\ (-1)^{|S_2|} & \text{if } S_1 = S_2 \end{cases}$$

Proof. If $S_1 \subsetneq S_2$ then there exists a vertex $v \in S_2 \backslash S_1$. We can pair each set $S_1 \subseteq S \subseteq S_2$ with $S \triangle \{v\}$, its symmetric difference with $\{v\}$. Clearly, in each pair of sets one is of odd size and one is of even size, and thus their signs cancel each other. Therefore, the sum is zero. In the second case, the claim is straightforward. ◀

▶ **Lemma 17.** *$F(G)$ equals the number of k-tuples $(I_0, \ldots, I_{k-1})$ of independent sets in G such that $V(G) = I_0 \cup \ldots \cup I_{k-1}$.*

Proof. As $i(G[V'])$ counts the number of independent sets in $G[V']$, raising it to the k-th power (namely, $i(G[V'])^k$) counts the number of k-tuples of independent sets in $G[V']$.

Let $(I_0, \ldots, I_{k-1})$ be a k-tuple of independent sets in G. It appears exactly in terms of the sum corresponding to sets V' such that $I_0 \cup \ldots \cup I_{k-1} \subseteq V' \subseteq V(G)$. Each time this k-tuple is counted, it is counted with a sign determined by the parity of V'. By Lemma 16, the sum of the signs corresponding to sets $I_0 \cup \ldots \cup I_{k-1} \subseteq V' \subseteq V(G)$ is zero if $I_0 \cup \ldots \cup I_{k-1} \neq V(G)$ and one if $I_0 \cup \ldots \cup I_{k-1} = V(G)$. ◀

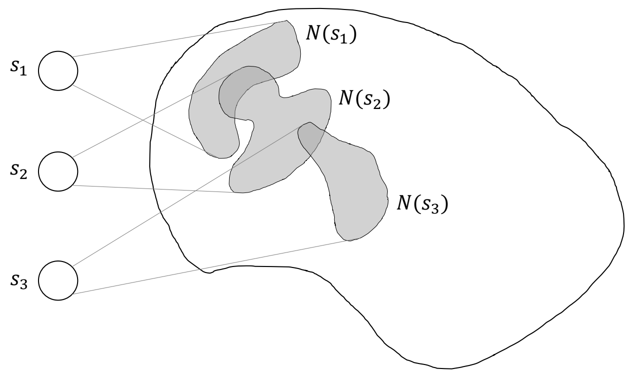
We conclude with

▶ **Corollary 18.** *$F(G)$ can be computed in time $O^*(2^n)$, and G is k-colorable if and only if $F(G) > 0$.*

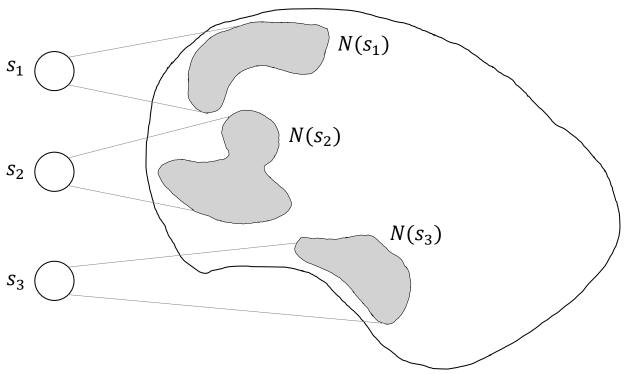
4 Faster Coloring Algorithms for (α, Δ)-bounded Graphs

The main purpose of this section is proving Theorem 4.

We first outline our approach. Let G be a graph with a constant chromatic number $\chi(G) \leqslant k$. It is well known that G must contain a large independent set. Let S be an independent set in G. We think of $|S|$ as a constant fraction of $|V(G)|$, when we consider k as a constant. Let $c : (V(G)\backslash S) \to [k]$ be a k-coloring of the induced sub-graph $G[V(G)\backslash S]$. We say that c can be *extended* to a k-coloring of G if there exists a proper k-coloring $c' : V(G) \to [k]$ such that $c'|_{V(G)\backslash S} = c$. For a subset $V' \subseteq V(G)\backslash S$ of vertices, we say that *c does not use the full palette on* V' if $|c(V')| < k$, namely, if c does not use all k colors on the vertices of V'. Clearly, a proper k-coloring c of $V(G)\backslash S$ can be extended to a proper k-coloring of G if and only if $|c(N(s))| < k$ for every $s \in S$. Our approach, on a high-level, is to construct an algorithm that finds an extendable k-coloring of $V(G)\backslash S$. We aim to do so in $O\left(2^{|V(G)\backslash S|}(2-\varepsilon)^{|S|}\right)$ time.



In Section 4.1 we consider a restricted version of the problem in which the independent set S has the following two additional properties. First, we assume that every vertex $s \in S$ is of degree $\deg(s) \leqslant \Delta$, where Δ is some constant. Second, we assume that no pair of vertices $s_1, s_2 \in S$ share a neighbor in G. Equivalently, the neighborhoods $N(s)$ for every $s \in S$ are all disjoint. Under these conditions, we present an algorithm that runs in $O\left(2^{|V(G)\setminus S|}\left(2-\varepsilon\right)^{|S|}\right)$ time, where ε depends only on Δ. As ε does not depend on k, we can in fact compute the chromatic number of G exponentially faster than $O^*(2^n)$ if G contains an independent set S with these properties. We also observe that if G is of maximum degree Δ then it contains a large such independent set S. Our algorithm is based on methods that generalize Section 3, and on a simple approach to implicitly compute values of the Inverse Möbius Transform.



In Section 4.2 we modify the algorithm of Section 4.1 and remove the second assumption on S. Namely, we now only assume that S is an independent set and that for every $s \in S$ we have $\deg(s) \leqslant \Delta$. Our algorithm still runs in $O\left(2^{|V(G)\setminus S|}\left(2-\varepsilon\right)^{|S|}\right)$ time. A main ingredient in the modification is a new removal lemma for small subsets. The proof of this combinatorial lemma is given in the full version of the paper and its statement is used in a black-box manner in this section.

In Section 4.3 we extend the result to List Coloring.

4.1 k-coloring bounded-degree graphs

In this subsection we begin illustrating the ideas leading towards proving Theorem 4. We also prove the following (much) weaker statement.

▶ **Theorem 19.** *For every k, Δ there exists $\varepsilon_{k,\Delta} > 0$ such that we can solve k-coloring for graphs with maximum degree Δ in $O\left(\left(2-\varepsilon_{k,\Delta}\right)^n\right)$ time.*

In fact, as a graph G with maximum degree Δ has chromatic number $\chi(G) \leqslant \Delta + 1$, we can compute the chromatic number of a graph with degrees bounded by Δ in time $O\left(\left(2-\varepsilon_{\Delta+1,\Delta}\right)^n\right)$.

As outlined in the beginning of this section, our approach begins by finding a large independent set with some additional properties. We show that a graph with bounded degrees must contain a very large independent set S such that the distance between each pair of vertices in S is at least three. In other words, S is an independent set, and no pair of vertices in S share a neighbor. In particular, the neighborhoods $N(s)$ for $s \in S$ are all disjoint. The core theorem of this subsection is

▶ **Theorem 20.** *Let G be a graph and $S \subseteq V(G)$ a set of vertices such that the distance between each two vertices in S is at least three and the degree of each vertex in S is at most Δ. For any k, we can solve k-coloring for G in $O^*\left(2^{|V(G)|-|S|} \cdot (2-2^{-\Delta})^{|S|}\right)$ time.*

It is important to note that the existence of such a set S is our sole use of the bound on the graph degrees. Note that the bound of Theorem 20 does not depend on k. Thus, we get an exponential improvement for computing the chromatic number of a graph G that contains a large enough set S with the stated properties.

Before proving Theorem 20, we describe a simple algorithm for finding a set S with the required properties in bounded-degree graphs.

▶ **Lemma 21.** *Let G be a graph with maximum degree at most Δ. There exists a set $S \subseteq V(G)$ of at least $\frac{1}{1+\Delta^2} \cdot |V(G)|$ vertices such that the distance between every distinct pair $s_1, s_2 \in S$ is at least three. Furthermore, we can find such S efficiently.*

Proof. We construct S in a greedy manner. We begin with $S = \varnothing$ and $V' = V(G)$. As long as V' is not empty we pick an arbitrary vertex $v \in V'$ and add it to S. We then remove from V' the vertex v and every vertex of distance at most two from it.

By construction, the minimum distance between a pair of vertices in S is at least three. The size of the 2-neighborhood of a vertex is bounded by $1 + \Delta + \Delta \cdot (\Delta - 1) = 1 + \Delta^2$ and thus we get the desired lower bound on the size of S. ◀

Theorem 19 now follows from Lemma 21 and Theorem 20.

Proof of Theorem 19. Let G be a graph of maximum degree at most Δ and let k be an integer. By Lemma 21, we can construct a set S of size $|S| \geqslant \frac{1}{1+\Delta^2} \cdot |V(G)|$ satisfying the conditions of Theorem 20. Thus, by Theorem 20, we can solve k-coloring for G in time

$$O^*\left(2^{n-\frac{1}{1+\Delta^2}n} \cdot (2-2^{-\Delta})^{\frac{1}{1+\Delta^2}n}\right) = O^*\left(\left(2 \cdot \left(1 - 2^{-(\Delta+1)}\right)^{\frac{1}{1+\Delta^2}}\right)^n\right). \quad ◀$$

In the rest of the subsection we prove Theorem 20.

▶ **Definition 22.** *For subsets $V' \subseteq V(G) \backslash S$ and $S' \subseteq S$ denote by $\beta(V', S')$ the number of independent sets I in $G[V']$ that intersect every neighborhood $N(s)$ of $s \in S'$, that is, $I \cap N(s) \neq \varnothing$ for every $s \in S'$.*

Consider, for a subset $S' \subseteq S$, the following sum

$$h(G, S') := \sum_{V' \subseteq V(G) \backslash S} (-1)^{|V(G)|-|V'|} \beta\left(V', S'\right)^k .$$

The following proof is almost identical to the proof of Lemma 17 in Section 3.

▶ **Lemma 23.** *$h(G, S')$ is the number of covers of $V(G) \backslash S$ by k-tuples $(I_0, \ldots, I_{k-1})$ of independent sets in $G[V(G) \backslash S]$ such that $I_i \cap N(s) \neq \varnothing$ for every $s \in S'$ and every $0 \leqslant i \leqslant k-1$.*

Proof. Each value of $\beta(V', S')$ counts independent sets in $G[V']$ that intersect every neighborhood $N(s)$ for $s \in S'$.

Each k-tuple $(I_0, \ldots, I_{k-1})$ of that type is counted in terms corresponding to sets V' such that

$$I_0 \cup \ldots \cup I_{k-1} \subseteq V' \subseteq V(G) \backslash S.$$

By Lemma 16 the multiplicity with which such k-tuple is counted is one if

$$I_0 \cup \ldots \cup I_{k-1} = V(G) \backslash S.$$

and zero otherwise. ◀

Consider the following expression.

$$H(G,S) := \sum_{S' \subseteq S} (-1)^{|S'|}\, h(G,S')$$

$H(G,S)$ is the number of covers of $V(G)\backslash S$ by k-tuples of independent sets that *do not use the full palette* on any neighborhood $N(s)$ for $s \in S$. The precise claim follows.

▶ **Lemma 24.** *$H(G,S)$ is the number of covers of $V(G)\backslash S$ by k-tuples $(I_0,\ldots,I_{k-1})$ of independent sets in $G[V(G)\backslash S]$ such that for every $s \in S$ there exists $0 \leqslant i \leqslant k-1$ such that $I_i \cap N(s) = \varnothing$.*

Proof. In Lemma 23 we showed that $h(G,S')$ counts the number of covers of $V(G)\backslash S$ by k-tuples $(I_0,\ldots,I_{k-1})$ of independent sets in $G[V(G)\backslash S]$ such that for every $s \in S'$ and for every $0 \leqslant i \leqslant k-1$ we have $I_i \cap N(s) \neq \varnothing$.

A covering k-tuple of independent sets $(I_0,\ldots,I_{k-1})$ is counted exactly in terms corresponding to subsets S' such that for every $0 \leqslant i \leqslant k-1$ and every $s \in S'$, the independent set I_i intersects the neighborhood $N(s)$. These are exactly the subsets S' such that

$$S' \subseteq \{s \in S \mid \forall 0 \leqslant i \leqslant k-1,\ I_i \cap N(s) \neq \varnothing\}.$$

Using Lemma 16 with $S_1 = \varnothing$ and $S_2 = \{s \in S \mid \forall 0 \leqslant i \leqslant k-1,\ I_i \cap N(s) \neq \varnothing\}$ we deduce that the multiplicity with which the k-tuple is counted is one if

$$\{s \in S \mid \forall 0 \leqslant i \leqslant k-1,\ I_i \cap N(s) \neq \varnothing\} = \varnothing$$

and zero otherwise. ◀

As outlined at the beginning of the section, we now claim that $H(G,S)$ is positive if and only if G is k-colorable. Note that for the correctness of this lemma we still did not use the disjointness of the neighborhoods $N(s)$. We will need this property to improve the computation time.

▶ **Lemma 25.** *Let G be a graph and S an independent set in it. Then, $H(G,S) > 0$ if and only if G is k-colorable.*

Proof. Assume that there exists a k-coloring $c : V(G) \to [k]$ of G. For $0 \leqslant i \leqslant k-1$ denote by

$$I_i := \{v \in V(G)\backslash S \mid c(v) = i\}$$

the subset of $V(G)\backslash S$ colored by i. Each I_i is an independent set as c is a proper coloring of G. Furthermore, for each $s \in S$, the neighborhood $N(s)$ does not intersect $I_{c(s)}$. Thus, $(I_0,\ldots,I_{k-1})$ is a cover of $V(G)\backslash S$ by k independent sets that do not all intersect any neighborhood $N(s)$ of $s \in S$. By Lemma 24, $H(G,S) \geqslant 1$.

On the other hand, if $H(G,S) > 0$ then by Lemma 24 there exists a cover by independent sets and in particular a k-coloring $c : V(G)\backslash S \to [k]$ of $G[V(G)\backslash S]$ such that the full palette is not used on any neighborhood $N(s)$ for $s \in S$. Thus, we may extend c to a k-coloring $c' : V(G) \to [k]$ of the entire graph by coloring each $s \in S$ with a color that does not appear in $c(N(s))$. As S is an independent set, this coloring is proper. ◀

Up to this point, we have formalized the outline from the beginning of this section, reducing k-coloring to a problem of k-coloring with some restrictions the smaller graph $G[V(G)\backslash S]$ and then to the computation of $H(G,S)$.

Unfortunately, $H(G,S)$ is a sum of $2^{|S|}$ terms, each of the form $h(G,S')$ which is a sum of $2^{|V(G)|-|S|}$ terms by itself. Evidently, there are 2^n different terms of the form $\beta(V',S')$ that are used in the definition of $H(G,S)$. Thus, we cannot hope to compute $H(G,S)$ in less than 2^n steps if we need to explicitly examine 2^n terms of the form $\beta(\cdot,\cdot)$. Moreover, it is also not clear how quickly we can compute the values of $\beta(\cdot,\cdot)$.

We begin by explaining how values of $\beta(\cdot)$ can be computed efficiently. The term $h(G,S')$ is a weighted sum of the values $\beta(V',S')$ for all $V' \subseteq V(G)\backslash S$. Denote by $\beta_\mu(V',S')$ the indicator function that gets the value 1 if V' is an independent set in $G[V(G)\backslash S]$ and for every $s \in S'$ we have $V' \cap N(s) \neq \emptyset$, and 0 otherwise. We can efficiently compute the value of β_μ for a specific input in a straightforward manner (i.e., checking whether it is an independent set that intersects the relevant sets). We observe that

$$\beta\left(V',S'\right) = \sum_{V''\subseteq V'} \beta_\mu\left(V'',S'\right),$$

thus, $\beta = \hat{\beta_\mu}$ as functions of V', and we can compute the values of $\beta(V',S')$ for all $V' \subseteq V(G)\backslash S$ in $O^*(2^{|V(G)|-|S|})$ time using the Inverse Möbius Transform presented in Section 2.1.

An improvement to the running time comes from noticing that for many inputs (V',S') the value of $\beta(V',S')$ is zero. In particular, if $V' \cap N(s) = \emptyset$, for some $s \in S'$, then $\beta(V',S') = 0$ as no subset (and in particular no independent set) in V' intersects $N(s)$. In the computation of $h(G,S')$ we only need to consider terms corresponding to subsets $V' \subseteq V(G)\backslash S$ in which for every $s \in S'$ the intersection $V' \cap N(s)$ is non-empty, as the values of other terms are all zero. We present a variant of the Inverse Möbius Transform that computes only the non-zero values by implicitly setting the others to zero. We then show that for most subsets $S' \subseteq S$ the number of non-zero entries is exponentially smaller than $2^{|V(G)|-|S|}$.

▶ **Definition 26.** *For any $S' \subseteq S$ denote by $B(S') := \{V' \subseteq V(G)\backslash S \mid \forall s \in S'.\, V' \cap N(s) \neq \emptyset\}$ the set of all subsets of $V(G)\backslash S$ intersecting all neighborhoods of S'.*

As we observed above, for every $V' \notin B(S')$ we have $\beta(V',S') = 0$. We conclude that

▶ **Observation 27.** *For every S' we have*

$$h(G,S') = \sum_{V'\in B(S')} (-1)^{|V(G)|-|V'|}\, \beta\left(V',S'\right)^k.$$

▶ **Lemma 28.** *If the neighborhoods $N(s)$ are disjoint for all $s \in S'$, then we can compute $h(G,S')$ in $O^*(|B(S')|)$ time.*

Proof. It suffices to compute $\beta(V',S')$ for every $V' \in B(S')$ and then use Observation 27. We do so by introducing a variant of the Inverse Möbius Transform that implicitly sets the value of $\beta(V',S')$ to zero for every $V' \notin B(S')$.

We first note that

$$B(S') \cong P\left(V(G)\backslash\left(S \cup \bigcup_{s\in S'} N(s)\right)\right) \times \bigtimes_{s\in S'} \left(P(N(s))\backslash\{\emptyset\}\right).$$

Thus, we can efficiently construct a simple bijection between $[|B(S')|]$ and $B(S')$ as a Cartesian product. We can also efficiently check if a set V' belongs to $B(S')$. Let *index* : $B(S') \to [|B(S')|]$ be a map from $B(S')$ to indices of $[|B(S')|]$. If $V' \notin B(S')$ we define

$index(V') = -1$. By the observation above, we can define $index$ in way for which $index$ and $index^{-1}$ are efficiently computable. We also arbitrarily order the vertices of $V(G)\backslash S$ as $v_1, v_2, \ldots, v_{|V(G)\backslash S|}$.

We describe the algorithm in pseudo-code.

Algorithm 1 Algorithm for the proof of Lemma 28.

```
Initialize an array f of size |B(S')|;
for ℓ in [|B(S')|] do
    if index^{-1}(ℓ) is an independent set in G[V(G)\S] then
        f(ℓ) ← 1 ;
    else
        f(ℓ) ← 0 ;
for i in [|V(G)\S|] do
    for ℓ in [|B(S')|] do
        V' ← index^{-1}(ℓ) ;
        if v_i ∈ V' and index(V'\{v_i}) ≠ −1 then
            f(ℓ) ← f(ℓ) + f(index(V'\{v_i}));
```

We view f throughout the algorithm as function $f : B(S') \to \mathbb{N}$. Denote the function represented by f at the end of the first *for* loop by f_0. By definition, $f_0(V') = \beta_\mu(V', S')$ for every $V' \in B(S')$. Denote by f_i the function represented by f at the end of the i-th iteration of the second (outer) *for* loop.

We observe that f_i is defined using f_{i-1} as

$$f_i(V') = \begin{cases} f_{i-1}(V') + f_{i-1}(V'\backslash\{v_i\}) & \text{if } v_i \in V' \\ f_{i-1}(V') & \text{otherwise} \end{cases}$$

where $f_{i-1}(V'\backslash\{v_i\})$ is implicitly defined to be zero if $V'\backslash\{v_i\} \notin B(S')$.

By induction on i, similar to this of Section 2.1, we can show that

$$f_i(V') = \sum_{\substack{V'' \subseteq V' \\ V''\backslash\{v_1,\ldots,v_i\} = V'\backslash\{v_1,\ldots,v_i\}}} f(V'').$$

In particular, by the end of the algorithm $f = \hat{f}_0 = \hat{\beta}_\mu = \beta$ for the entire domain $B(S')$. ◀

After computing $h(G, S')$ for every $S' \subseteq S$ we can compute $H(G, S)$ in $O^*(2^{|S|})$ time. We thus finish the proof of Theorem 20 with the following counting lemma.

▶ **Lemma 29.** *Assume that the neighborhoods $N(s)$ are disjoint for all $s \in S$ and that each neighborhood is of size $|N(s)| \leqslant \Delta$. Then, $\sum_{S' \subseteq S} |B(S')| = O^*\left(2^{|V(G)\backslash S|} \cdot (2 - 2^{-\Delta})^{|S|}\right)$.*

Proof. Denote $n(s) := |N(s)|$. Also denote by $N = \bigcup_{s \in S} N(s)$ all neighbors of vertices of S and by $N^c = (V(G)\backslash S)\backslash N$ their complement in $V(G)\backslash S$. We have

$$\begin{aligned}|B(S')| &= 2^{|N^c|} \cdot \prod_{s \in S'} \left(2^{n(s)} - 1\right) \cdot \prod_{s \in S\backslash S'} 2^{n(s)} \\ &= 2^{|N^c|} \cdot \prod_{s \in S'} \left(1 - 2^{-n(s)}\right) \cdot \prod_{s \in S} 2^{n(s)} \\ &= 2^{|N^c|} \cdot \prod_{s \in S'} \left(1 - 2^{-n(s)}\right) \cdot 2^{|N|} \\ &= 2^{|V(G)\backslash S|} \cdot \prod_{s \in S'} \left(1 - 2^{-n(s)}\right).\end{aligned}$$

For every $s \in S$ we have $n(s) \leqslant \Delta$ and thus $\left(1-2^{-n(s)}\right) \leqslant \left(1-2^{-\Delta}\right)$. Hence,

$$\begin{aligned} |B(S')| &\leqslant 2^{|V(G)\setminus S|} \cdot \prod_{s \in S'} \left(1-2^{-\Delta}\right) \\ &= 2^{|V(G)\setminus S|} \cdot \left(1-2^{-\Delta}\right)^{|S'|}. \end{aligned}$$

Therefore we have

$$\begin{aligned} \sum_{S' \subseteq S} |B(S')| &\leqslant \sum_{S' \subseteq S} 2^{|V(G)\setminus S|} \cdot \left(1-2^{-\Delta}\right)^{|S'|} \\ &= 2^{|V(G)\setminus S|} \cdot \sum_{i=0}^{|S|} \binom{|S|}{i} \left(1-2^{-\Delta}\right)^{i} \\ &= 2^{|V(G)\setminus S|} \cdot (2-2^{-\Delta})^{|S|}. \end{aligned}$$

◀

4.2 From bounded-degree graphs to (α, Δ)-bounded graphs

In this section we prove the main technical theorem of the paper.

▶ **Theorem 4.** *For every $\Delta, \alpha > 0$ there exists $\varepsilon_{\Delta,\alpha} > 0$ such that we can compute the chromatic number of (α, Δ)-bounded graphs in $O\left((2-\varepsilon_{\Delta,\alpha})^n\right)$ time.*

We prove the following seemingly weaker statement.

▶ **Theorem 30.** *For every $k, \Delta, \alpha > 0$ there exists $\varepsilon_{k,\Delta,\alpha} > 0$ such that we can solve k-coloring for (α, Δ)-bounded graphs in $O\left((2-\varepsilon_{k,\Delta,\alpha})^n\right)$ time.*

We then note that Theorem 30 in fact implies Theorem 4. Let G be a (α, Δ)-bounded graph. We use Theorem 30 for every $1 \leqslant k \leqslant \Delta$. If we did not find a valid coloring of G, then $\chi(G) \geqslant \Delta + 1$ and we may use a standard argument (fully presented in the full version of the paper) to show that removing all vertices of degree at most Δ does not change $\chi(G)$. By definition of (α, Δ)-bounded graphs, removing these vertices leaves a graph with at most $(1-\alpha)n$ vertices and thus the standard chromatic number algorithm runs in $O^*(2^{(1-\alpha)n})$ time.

As in Section 4.1, we deduce Theorem 30 from the following theorem.

▶ **Theorem 31.** *Let G be a graph and $S \subseteq V(G)$ an independent set in G. Assume that the degree of each vertex in S is at most Δ. Then, we can solve k-coloring for G in $O^*\left(2^{|V(G)|} \cdot (1-\varepsilon_{k,\Delta})^{|S|}\right)$ time, for some constant $\varepsilon_{k,\Delta} > 0$.*

Let G be a graph with a subset $U \subseteq V(G)$ of vertices such that for every $v \in U$ we have $\deg(v) \leqslant \Delta$. In a similar fashion to Lemma 21 of the previous subsection (and even slightly simpler), we can greedily construct a subset $S \subseteq U$ of size $|S| \geqslant \frac{1}{1+\Delta} \cdot |U|$ which is an independent set. Thus, Theorem 31 immediately implies Theorem 4. Unlike the case of Section 4.1, this time the neighborhoods $N(s)$ for $s \in S$ are not necessarily disjoint. Thus, statements comparable to Lemma 29 are not true. Our solution for this problem is surprisingly general. In the full version of the paper we prove Theorem 10. Plugging $\mathcal{F} = \{N(s)\}_{s \in S}$, we get a small set $U' \subseteq V(G)\setminus S$ of graph vertices, and a large subset $S' \subseteq S$ of the independent set, such that the neighborhoods $N(s)$ of $s \in S'$ become pairwise disjoint if we remove the vertices of U' from G. As we want to preserve the correctness of the algorithm, we do not actually remove U' from G, but *enumerate* over the colors they receive in a proper k-coloring, if one exists. The main technical gap is adjusting the algorithm and proofs of Section 4.1 to the case in which some of the graph vertices have fixed colors.

▶ **Theorem 32.** *Let G be a graph, $V_0 \subseteq V(G)$ a subset of its vertices and $c : V_0 \to [k]$ a proper k-coloring of $G[V_0]$. Denote by $V := V(G)\backslash V_0$. Let $S \subseteq V$ be an independent set in G such that the distance in $G[V]$ between each two vertices of S is at least three and the degree in $G[V]$ of each vertex in S is at most Δ. For any k, we can decide whether c can be extended to a k-coloring of the entire graph G in $O^*\left(2^{|V|-|S|} \cdot (2-2^{-\Delta})^{|S|}\right)$ time.*

Throughout the rest of the section, it is important to carefully distinguish $V(G)$ from V. Note that V does not include the vertices of V_0, as their colors are already fixed. For $j \in [k]$, denote by $V_0^j := c^{-1}(j)$ the subset of V_0 colored by j color. Note that $V_0 = \bigcup_{j=1}^{k} V_0^j$. We begin adapting the algorithm by redefining the $\beta(\cdot,\cdot)$ function.

▶ **Definition 33.** *For subsets $V' \subseteq V\backslash S$, $S' \subseteq S$, and a color $j \in [k]$, we denote by $\beta_j(V', S')$ the number of sets $I \subseteq V'$ such that $I \cup V_0^j$ is an independent set in G and that $I \cup V_0^j$ intersects $N(s)$ for every $s \in S'$, that is, for every $s \in S'$ we have $\left(I \cup V_0^j\right) \cap N(s) \neq \varnothing$.*

We also revise the definition of

$$h(G, S') := \sum_{V' \subseteq V\backslash S} (-1)^{|V|-|V'|} \prod_{j=0}^{k-1} \beta_j\left(V', S'\right).$$

The proof of Lemma 23 can be easily revised to show the following.

▶ **Lemma 34.** *$h(G, S')$ is the number of covers of $V\backslash S$ by k-tuples of sets $I_0, \ldots, I_{k-1}$ such that for every $j \in [k]$, $I_j \cup V_0^j$ is an independent set in G and that for every $s \in S'$ and every $j \in [k]$ the set $I_j \cup V_0^j$ intersects the neighborhood $N(s)$.*

Without revising the definition of $H(G, S)$, the proof of Lemma 24 now shows that

▶ **Lemma 35.** *$H(G, S)$ is the number of covers of $V\backslash S$ by k-tuples of sets $I_0, \ldots, I_{k-1}$ such that for every $j \in [k]$, $I_j \cup V_0^j$ is an independent set in G and that for every $s \in S$ the neighborhood $N(s)$ is not intersected by at least one of the k independent sets $\left(I_j \cup V_0^j\right)$ for $j \in [k]$.*

Therefore, we have

▶ **Lemma 36.** *Let G be a graph, $V_0 \subseteq V(G)$ a subset of its vertices and $c : V_0 \to [k]$ a proper k-coloring of $G[V_0]$. Denote by $V := V(G)\backslash V_0$. Let $S \subseteq V$ be an independent set in G. Then, $H(G, S) > 0$ if and only if c can be extended to a k-coloring of G.*

The non-trivial part of the revision and the heart of this subsection, is adjusting the algorithm for computing the values of $h(G, S')$ without increasing the running time.

For $j \in [k]$, denote by

$$S_j := \{s \in S \mid N(s) \cap V_0^j \neq \varnothing\}$$

the set of vertices in S whose neighborhood intersects V_0^j. The key observation of this subsection follows.

▶ **Lemma 37.** *For any $j \in [k]$, $S' \subseteq S$, $V' \subseteq V$, we have*

$$\beta_j\left(V', S'\right) = \beta_j\left(V', S' \cup S_j\right)$$

Proof. For any set $I \subseteq V'$ the set $I \cup V_0^j$ intersects every set in $\{N(s)\}_{s \in S_j}$. In particular, an independent set $I \subseteq V'$ intersects all of $\{N(s)\}_{s \in S'}$ if and only if it intersects all of $\{N(s)\}_{s \in (S' \cup S_j)}$. ◀

Lemma 37 implies that it is enough to compute $\beta_j(V', S')$ only for sets $S' \subseteq S \backslash S_j$, as its other values can be deduced from these as $\beta_j(V', S') = \beta_j(V', S' \backslash S_j)$.

For any $S' \subseteq S$ we again denote by $B(S') := \{V' \subseteq V \backslash S \mid \forall s \in S'.\ V' \cap N(s) \neq \emptyset\}$ the set of all subsets of $V \backslash S$ intersecting all neighborhoods of S'. Note the slight difference from Section 4.1 of considering subsets of $V \backslash S$ and not of $V(G) \backslash S$.

As for every $s \in S \backslash S_j$, $N(s) \cap V_0^j = \emptyset$, we still have that for every $V' \notin B(S')$ the value of $\beta_j(V', S')$ is zero. In particular, we can still use the implicit Inverse Möbius Transform of Lemma 28 and get

▶ **Lemma 38.** *Assume $S' \subseteq S \backslash S_j$. We can compute $\beta_j(V', S')$ for every $V' \in B(S')$ in $O^*(|B(S')|)$ time.*

By Lemma 29 we get

$$\sum_{S' \subseteq S \backslash S_j} |B(S')| = O^* \left(2^{|V \backslash S|} \cdot (2 - 2^{-\Delta})^{|S \backslash S_j|} \right).$$

We can thus compute $\beta_j(V', S')$ for every $S' \subseteq S \backslash S_j$ and every $V' \in B(S')$ in $O^*\left(2^{|V \backslash S|} \cdot (2 - 2^{-\Delta})^{|S \backslash S_j|}\right)$ time. This is the time to emphasise a crucial point. Note that if we would consider every $S' \subseteq S$ instead of $S' \subseteq S \backslash S_j$, then the running time would be $O^*\left(2^{|V \backslash S|} \cdot (2 - 2^{-\Delta})^{|S \backslash S_j|} \cdot 2^{|S_j|}\right)$, as the neighborhoods corresponding to S_j are intersected by V_0^j. This is why we compute every β_j separately, and do so for *all* relevant sets S' before computing even a single value $h(G, S')$. As it always holds that $|S \backslash S_j| \leqslant |S|$, we conclude that

▶ **Corollary 39.** *We can compute $\beta_j(V', S')$ for all $j \in [k]$, $S' \subseteq S \backslash S_j$ and $V' \in B(S')$ in $O^*\left(2^{|V \backslash S|} \cdot (2 - 2^{-\Delta})^{|S|}\right)$ time.*

Note that $k = O^*(1)$.

We are now ready to compute the values of $h(G, S')$. We start by making the following observation.

▶ **Observation 40.** *If $\bigcap_{j=0}^{k-1} S_j \neq \emptyset$ then c cannot be extended to a coloring of G.*

This holds as if some $s \in S$ has neighbors colored in each of the k colors then it cannot be properly colored. We are thus dealing with the case where $\bigcap_{j=0}^{k-1} S_j = \emptyset$.

▶ **Lemma 41.** *For any $S' \subseteq S$ and $V' \subseteq V \backslash S$ such that $V' \notin B(S')$ we have*

$$\prod_{j=0}^{k-1} \beta_j(V', S') = 0.$$

Proof. As $V' \notin B(S')$ there exists some $s \in S$ such that $V' \cap N(s) = \emptyset$. As $\bigcap_{j=0}^{k-1} S_j = \emptyset$, there exists a $j \in [k]$ for which $s \notin S_j$. Thus, $V_0^j \cap N(s) = \emptyset$ as well. We conclude that $\beta_j(V', S') = 0$. ◀

From Lemma 37 and Lemma 41 we conclude that

$$h(G, S') := \sum_{V' \in B(S')} (-1)^{|V| - |V'|} \prod_{j=0}^{k-1} \beta_j\left(V', S' \backslash S_j\right).$$

Thus, we can compute $h(G, S')$ in $O^*(|B(S')|)$ time using the values computed in Corollary 39. Using Lemma 29 once again, we get that

$$\sum_{S' \subseteq S} |B(S')| = O^* \left(2^{|V \setminus S|} \cdot (2 - 2^{-\Delta})^{|S|} \right)$$

which completes the proof of Theorem 32.

We can now prove Theorem 31.

Proof. We apply the removal lemma of Theorem 10 to $\mathcal{F} = \{N(s)\}_{s \in S}$ with C to be chosen later. We thus get a sub-collection $S' \subseteq S$ and a subset of vertices $V_0 \subseteq V(G) \setminus S$ such that $|S'| > \rho(\Delta, C) \cdot |S| + C \cdot |V_0|$ and that for every $s_1, s_2 \in S'$ it holds that $N(s_1) \cap N(s_2) \subseteq V_0$. Denote by $V = V(G) \setminus (S' \cup V_0)$. We enumerate over all k-colorings $c : V_0 \to [k]$. For each coloring c, we check if it is a proper k-coloring of $G[V_0]$ and if so we apply Theorem 32 on G with V_0, c, S'. If any of the applications of Theorem 32 returned that there exists a valid extension of c to a coloring of G, we return that G is k-colorable, and otherwise that it is not.

The running time of the entire algorithm, up to polynomial factors, is

$$\begin{aligned} k^{|V_0|} \cdot \left(2^{|V \setminus S'|} \cdot (2 - 2^{-\Delta})^{|S'|} \right) &= 2^{|V|} \cdot k^{|V_0|} \cdot (1 - 2^{-(\Delta+1)})^{|S'|} \\ &\leqslant 2^{|V|} \cdot k^{|V_0|} \cdot (1 - 2^{-(\Delta+1)})^{\rho(\Delta, C) \cdot |S| + C \cdot |V_0|}. \end{aligned}$$

By picking $C = \frac{\log k}{-\log\left(1 - 2^{-(\Delta+1)}\right)} > 0$ we have

$$k^{|V_0|} \cdot (1 - 2^{-(\Delta+1)})^{C \cdot |V_0|} = 1$$

and thus the running time is bound by

$$2^{|V|} \cdot (1 - 2^{-(\Delta+1)})^{\rho(\Delta, C) \cdot |S|}.$$ ◀

4.3 Generalization to List Coloring

In this Section we deduce Theorem 5.

Proof. Let $G = (V, E)$ be a (α, Δ)-bounded graph with color lists C_v of size at most k for each $v \in V$. Denote by $U = \bigcup_{v \in V} C_v$ the color universe. Note that $|U|$ might be as large as kn, where $n = |V|$. We construct a new graph G' on the set of vertices $V \cup U$ by adding $|U|$ isolated vertices to the graph G and then connecting each node $v \in V$ to every node $u \in U$ such that $u \notin C_v$. If we color each $u \in U$ by the color u, then there is an extension of this coloring to a (regular) $|U|$-coloring for all of G' if and only if G is list-colorable.

We now follow the proof of Theorem 31. We can again find a subset $S \subset V$ of size $|S| \geqslant \frac{\alpha}{1+\Delta} n$ which is an independent set in G that contains only vertices of degree at most Δ. We then apply the removal lemma of Theorem 10 to $\mathcal{F} = \{N(s)\}_{s \in S}$ where the neighbourhoods are within G and C is to be chosen later. Define S' and V_0 as in the proof of Theorem 31. Since every color-list C_v is of size at most k, there are only $k^{|V_0|}$ possible colorings $c : V_0 \to U$. We enumerate over these colorings and for each one which is a proper coloring of $G[V_0]$ we apply Theorem 32 on G' where $U \cup V_0$ are already colored (by their corresponding colors and by c, respectively). The crucial point here is that in Theorem 32 the running time depends on $|V|$, the number of uncolored vertices, and is independent of the number of colored vertices. In particular, the total runtime is thus

$$\begin{aligned} k^{|V_0|} \cdot \left(2^{|V \setminus (S' \cup V_0)|} \cdot (2 - 2^{-\Delta})^{|S'|} \right) &\leqslant 2^{|V|} \cdot k^{|V_0|} \cdot (1 - 2^{-(\Delta+1)})^{|S'|} \\ &\leqslant 2^{|V|} \cdot k^{|V_0|} \cdot (1 - 2^{-(\Delta+1)})^{\rho(\Delta, C) \cdot |S| + C \cdot |V_0|}. \end{aligned}$$

We can thus again pick $C = \frac{\log k}{-\log\left(1-2^{-(\Delta+1)}\right)} > 0$ to have

$$k^{|V_0|} \cdot (1 - 2^{-(\Delta+1)})^{C\cdot|V_0|} = 1$$

which results in a running time bounded by

$$2^{|V|} \cdot (1 - 2^{-(\Delta+1)})^{\rho(\Delta,C)\cdot|S|}.$$ ◀

4.4 On finding a coloring

In both previous subsections, we used the bounds on the degrees only in order to construct a *good* independent set S. After doing so, we may apply the self-reduction of Section 2.2 to the graph $G[V(G)\backslash S]$, in which we no longer care about the number of edges nor the degrees. This would result in finding a k-coloring of $G[V(G)\backslash S]$. Such coloring can be extended to a k-coloring of G by the constructive proof of Lemma 25. The exact claim follows.

▶ **Lemma 42.** *In the conditions of Theorem 20 or Theorem 31 we can also* find *a k-coloring of G.*

Proof. Consider the reduction between the decision and search versions of k-coloring of Lemma 12. Since adding edges to vertices whose both endpoints are in $V(G)\backslash S$ does not violate the conditions of the theorems, we may apply the reduction of Lemma 12 to $G[V(G)\backslash S]$. By the end of the reduction, we have a k-coloring of $G[V(G)\backslash S]$ that is a restriction of some k-coloring of G. We can extend this k-coloring to a k-coloring of G using the algorithm of Lemma 25. ◀

As a corollary, in the conditions of Theorem 4 we can also *find* a k-coloring of G.

5 Summary of the full version of this paper

Due to length limitation, several main components of the paper appear only in the full version of the paper. The full version is available on-line.

5.1 Removal Lemma For Small Sets

We show that any collection of small sets must contain a *large* sub-collection of *almost* pairwise-disjoint sets. The precise statement follows.

▶ **Theorem 10.** *Let $\mathcal{F}$ be a collection of subsets of a universe U such that every set $F \in \mathcal{F}$ is of size $|F| \leqslant \Delta$. Let $C > 0$ be any constant. Then, there exist subsets $\mathcal{F}' \subseteq \mathcal{F}$ and $U' \subseteq U$, such that*

- $|\mathcal{F}'| > \rho(\Delta, C) \cdot |\mathcal{F}| + C \cdot |U'|$, *where $\rho(\Delta, C) > 0$ depends only on Δ, C.*
- *The sets in $\mathcal{F}'$ are disjoint when restricted to $U\backslash U'$, i.e., for every $F_1, F_2 \in \mathcal{F}'$ we have $F_1 \cap F_2 \subseteq U'$.*

We should think of the statement of Theorem 10 in the following manner. We interpret an *almost* pairwise-disjoint sub-collection as a sub-collection that would become pairwise-disjoint after the removal of a *small* number of elements of the universe. If Δ is a constant, then the precise meaning of *small* and *large* is that on the one hand, the size of the sub-collection is at least a constant fraction of the size of the entire collection, and on the other hand, its size is arbitrarily larger than the number of removed universe elements. The constant C represents the exact meaning of *arbitrarily larger*.

We also discuss the optimality of Theorem 10. In particular, we show that in the settings of Theorem 10 we must have $\rho(\Delta, C) \leqslant (C + 1)^{-\Delta}$.

5.2 Reducing k-coloring to $(k-1)$-list-coloring

We prove the following reduction.

▶ **Theorem 6.** *Given an algorithm solving $(k-1)$-list-coloring in time $O\left((2-\varepsilon)^n\right)$ for some constant $\varepsilon > 0$, we can construct an algorithm solving k-coloring in time $O\left((2-\varepsilon')^n\right)$ for some (other) constant $\varepsilon' > 0$. Furthermore, the reduction is deterministic.*

Beigel and Eppstein [1] show that 4-list-coloring (as a special case of a $(4,2)$-CSP) can be solved in time $O(1.81^n)$. Therefore we conclude the proof of Theorem 8 regarding 5-coloring.

We include a short intuitive description of the reduction. By Theorem 4, it suffices to solve k-coloring for graphs in which most vertices have high degrees. We show that in this case, the graph has a small *dominating set*, this is a subset R of vertices such that every vertex not in R is adjacent to at least one vertex of R. Given a k-coloring of the dominating set, the problem of extending the coloring to a k-coloring of the entire graph becomes a problem of $(k-1)$-list-coloring the rest of the graph. This is because each vertex not in the dominating set has a neighbor in it, and thus has at least one of the k colors which it cannot use. Assuming the dominating set is small enough, we can enumerate over the k-colorings of vertices in it, and then solve the remaining $(k-1)$-list-coloring problem. The complete details appear in the full version.

5.3 Reducing k-coloring to $(k-2)$-list-coloring

We then prove the following much more complicated reduction.

▶ **Theorem 7.** *Given an algorithm solving $(k-2)$-list-coloring in time $O\left((2-\varepsilon)^n\right)$ for some constant $\varepsilon > 0$, we can construct an algorithm solving k-coloring with high probability in time $O\left((2-\varepsilon')^n\right)$ for some (other) constant $\varepsilon' > 0$.*

Once again, we use the 4-list-coloring algorithm of Beigel and Eppstein [1] to conclude the proof of Theorem 9 regarding 6-coloring.

The problem with generalizing the idea used in the proof of Theorem 6 is that even if R contains several neighbors of every graph vertex, it could be that in the correct k-coloring all of these neighbors are colored by the same color. In that case, the size of the list of possible colors would not get smaller than $(k-1)$. Thus, more involved algorithmic ideas are necessary for proving Theorem 7.

6 Conclusions and Open Problems

The main algorithmic contribution of the paper is Theorem 4. We use it in order to answer a few fundamental questions regarding the running time of k-coloring algorithms. In particular, we present the first $O\left((2-\varepsilon)^n\right)$ algorithms solving 5-coloring and 6-coloring, for some $\varepsilon > 0$. While the ε we can get using our tools is very small, this serves as the first proof that 5-coloring can be solved faster than we can currently compute the chromatic number in general. The upper bound on ρ in the full version shows that the magnitude of ε is a necessary consequence of using the removal lemma.

The main open problem that we leave unsettled is

▶ **Open Problem 43.** *Can we solve k-coloring in $O^*\left((2-\varepsilon_k)^n\right)$ time for some $\varepsilon_k > 0$, for every k?*

Theorem 4 makes some progress towards answering it, by giving some additional conditions on the input graph under which the answer is affirmative. In particular, we show that it holds for every graph that does not contain almost only vertices of super-constant degrees. In [12] very different techniques (using modifications of the FFT algorithm) were used to get a statement similar to Theorem 4 for graphs with bounded average degree. It seems like their methods do not extend to the case of (α, Δ)-bounded graphs, nevertheless, it is intriguing to find out whether a combination of their techniques with these presented in this paper can lead to further improvements.

While it is believed that $O^*(2^n)$ is the right bound for computing the chromatic number, we have no strong evidence to support this. There are reductions from popular problems and conjectures (like SETH) to other partitioning problems [6] or other parameterizations of the coloring problem [16]. It is interesting whether it can be showed that an $O^*\left((2-\varepsilon)^n\right)$ algorithm for computing the chromatic number would refute any other popular conjecture. This question was raised several times, including in the book of Fomin and Kratsch [11].

Another technical contribution of the paper is Theorem 10. We believe that the presented removal lemma could serve as a tool in the design of other exponential time algorithms. It would be interesting to find more problems for which it can be used.

References

1 Richard Beigel and David Eppstein. 3-coloring in time $O(1.3289^n)$. *Journal of Algorithms*, 54(2):168–204, 2005.

2 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Trimmed moebius inversion and graphs of bounded degree. *Theory of Computing Systems*, 47(3):637–654, 2010.

3 Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusion-exclusion. *SIAM Journal on Computing*, 39(2):546–563, 2009.

4 Jesper Makholm Byskov. Enumerating maximal independent sets with applications to graph colouring. *Operations Research Letters*, 32(6):547–556, 2004.

5 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In *International Workshop on Parameterized and Exact Computation*, pages 75–85. Springer, 2009.

6 Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as CNF-SAT. *ACM Transactions on Algorithms (TALG)*, 12(3):1–24, 2016.

7 Marek Cygan and Marcin Pilipczuk. Faster exponential-time algorithms in graphs of bounded average degree. *Information and Computation*, 243:75–85, 2015.

8 David Eppstein. Small maximal independent sets and faster exact graph coloring. In *Workshop on Algorithms and Data Structures*, pages 462–470. Springer, 2001.

9 Fedor V Fomin, Serge Gaspers, and Saket Saurabh. Improved exact algorithms for counting 3-and 4-colorings. In *International Computing and Combinatorics Conference*, pages 65–74. Springer, 2007.

10 Fedor V Fomin and Petteri Kaski. Exact exponential algorithms. *Communications of the ACM*, 56(3):80–88, 2013.

11 F.V. Fomin and D. Kratsch. *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2010.

12 Alexander Golovnev, Alexander S Kulikov, and Ivan Mihajlin. Families with infants: speeding up algorithms for np-hard problems using fft. *ACM Transactions on Algorithms (TALG)*, 12(3):1–17, 2016.

13 Thomas Dueholm Hansen, Haim Kaplan, Or Zamir, and Uri Zwick. Faster k-SAT algorithms using biased-PPSZ. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 578–589, 2019.

14 Thore Husfeldt. *Graph colouring algorithms*, page 277–303. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2015. doi:10.1017/CBO9781139519793.016.

15 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

16 Lars Jaffke and Bart MP Jansen. Fine-grained parameterized complexity analysis of graph coloring problems. In *International Conference on Algorithms and Complexity*, pages 345–356. Springer, 2017.

17 Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

18 Robert Kennes. Computational aspects of the mobius transformation of graphs. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):201–223, 1992.

19 Donald Ervin Knuth. Seminumerical algorithms. *The art of computer programming*, 2, 1997.

20 Vipin Kumar. Algorithms for constraint-satisfaction problems: A survey. *AI magazine*, 13(1):32–32, 1992.

21 Eugene L Lawler. A note on the complexity of the chromatic number problem, 1976.

22 László Lovász. Coverings and colorings of hypergraphs. In *Proc. 4th Southeastern Conference of Combinatorics, Graph Theory, and Computing*, pages 3–12. Utilitas Mathematica Publishing, 1973.

23 Burkhard Monien and Ewald Speckenmeyer. Solving satisfiability in less than 2n steps. *Discrete Applied Mathematics*, 10(3):287–295, 1985.

24 John W Moon and Leo Moser. On cliques in graphs. *Israel journal of Mathematics*, 3(1):23–28, 1965.

25 Ramamohan Paturi, Pavel Pudlák, Michael E Saks, and Francis Zane. An improved exponential-time algorithm for k-SAT. *Journal of the ACM (JACM)*, 52(3):337–364, 2005.

26 Marvin C Paull and Stephen H Unger. Minimizing the number of states in incompletely specified sequential switching functions. *IRE Transactions on Electronic Computers*, pages 356–367, 1959.

27 Gian-Carlo Rota. On the foundations of combinatorial theory i. theory of möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 2(4):340–368, 1964.

28 Ingo Schiermeyer. Deciding 3-colourability in less than $O(1.415^n)$ steps. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 177–188. Springer, 1993.

29 T Schoning. A probabilistic algorithm for k-SAT and constraint satisfaction problems. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 410–414. IEEE, 1999.

30 Larry Stockmeyer. Planar 3-colorability is polynomial complete. *ACM Sigact News*, 5(3):19–25, 1973.

31 Gerhard J Woeginger. Exact algorithms for NP-hard problems: A survey. In *Combinatorial optimization – eureka, you shrink!*, pages 185–207. Springer, 2003.

32 Frank Yates. *The design and analysis of factorial experiments*. Imperial Bureau of Soil Science Harpenden, UK, 1937.