PARAMETERIZED PRE-COLORING EXTENSION AND LIST COLORING PROBLEMS*

GREGORY GUTIN†, DIPTAPRIYO MAJUMDAR†, SEBASTIAN ORDYNIAK‡, AND MAGNUS WAHLSTRÖM†

Abstract. Golovach, Paulusma and Song (Inf. Comput. 2014) asked to determine the parameterized complexity of the following problems parameterized by k: (1) Given a graph G, a clique modulator D (a *clique modulator* is a set of vertices, whose removal results in a clique) of size k for G, and a list $L(v)$ of colors for every $v \in V(G)$, decide whether G has a proper list coloring; (2) Given a graph G, a clique modulator D of size k for G, and a pre-coloring $\lambda_P : X \to Q$ for $X \subseteq V(G)$, decide whether λ_P can be extended to a proper coloring of G using only colors from Q. For Problem 1 we design an $\mathcal{O}^*(2^k)$-time randomized algorithm and for Problem 2 we obtain a kernel with at most $3k$ vertices. Banik et al. (IWOCA 2019) proved the following problem is fixed-parameter tractable and asked whether it admits a polynomial kernel: Given a graph G, an integer k, and a list $L(v)$ of exactly $n-k$ colors for every $v \in V(G)$, decide whether there is a proper list coloring for G. We obtain a kernel with $\mathcal{O}(k^2)$ vertices and colors and a compression to a variation of the problem with $\mathcal{O}(k)$ vertices and $\mathcal{O}(k^2)$ colors.

1. Introduction.

Graph coloring is a central topic in Computer Science and Graph Theory due to its importance in theory and applications. Every text book in Graph Theory has at least a chapter devoted to the topic and the monograph of Jensen and Toft [25] is completely devoted to graph coloring problems focusing especially on more than 200 unsolved ones. There are many survey papers on the topic including recent ones such as [13, 22, 31, 33].

For a graph G, a *proper coloring* is a function $\lambda : V(G) \to \mathbb{N}_{\geq 1}$ such that for no pair u, v of adjacent vertices of G, $\lambda(u) = \lambda(v)$. In the widely studied COLORING problem, given a graph G and a positive integer p, we are to decide whether there is a proper coloring $\lambda : V(G) \to [p]$, where henceforth $[p] = \{1, \ldots, p\}$. In this paper, we consider two extensions of COLORING: the PRE-COLORING EXTENSION problem and the LIST COLORING problem. In the PRE-COLORING EXTENSION problem, given a graph G, a set Q of colors, and a *pre-coloring* $\lambda_P : X \to Q$, where $X \subseteq V(G)$, we are to decide whether there is a proper coloring $\lambda : V(G) \to Q$ such that $\lambda(x) = \lambda_P(x)$ for every $x \in X$. In the LIST COLORING problem, given a graph G and a list $L(u)$ of possible colors for every vertex u of G, we are to decide whether G has a proper coloring λ such that $\lambda(u) \in L(u)$ for every vertex u of G. Such a coloring λ is called a *proper list coloring*. Clearly, PRE-COLORING EXTENSION is a special case of LIST COLORING, where all lists of vertices $x \in X$ are singletons and the lists of all other vertices are equal to Q.

The p-COLORING problem is a special case of COLORING when p is fixed (i.e., not part of input). When $Q \subseteq [p]$ ($L(u) \subseteq [p]$, respectively), PRE-COLORING EXTENSION (LIST COLORING, respectively) are called p-PRE-COLORING EXTENSION (LIST p-COLORING, respectively). In classical complexity, it is well-known that p-COLORING, p-PRE-COLORING EXTENSION and LIST p-COLORING are polynomial-time solvable for $p \leq 2$, and the three problems become NP-complete for every $p \geq 3$ [28, 31]. In this paper, we solve several open problems about pre-coloring extension and list coloring problems, which lie outside classical complexity, so-called parameterized problems.

*A preliminary and shortened version of this paper has been accepted at STACS 2020.

†Royal Holloway, University of London, UK (gutin@cs.rhul.ac.uk, diptapriyo.majumdar@rhul.ac.uk, Magnus.Wahlstrom@rhul.ac.uk)

‡University of Leeds, UK (sordyniak@gmail.com)

We provide basic notions on parameterized complexity in the next section. For more information on parameterized complexity, see recent books [14, 18, 20].

The first two problems we study are the following ones stated by Golovach et al. [23] (see also [30]) who asked to determine their parameterized complexity. These questions were motivated by a result of Cai [10] who showed that COLORING WITH CLIQUE MODULATOR (the special case of PRE-COLORING EXTENSION WITH CLIQUE MODULATOR when $X = \emptyset$) is fixed-parameter tractable (FPT). Note that a *clique modulator* of a graph G is a set D of vertices such that $G - D$ is a clique. When using the size of a clique modulator as a parameter we will for convenience assume that the modulator is given as part of the input. Note that this assumption is not necessary (however it avoids having to repeat how to compute a clique modulator) as we will show in Section 2 that computing a clique modulator of size k is FPT and can be approximated to within a factor of two.

LIST COLORING WITH CLIQUE MODULATOR parameterized by k

Input: A graph G, a clique modulator D of size k for G, and a list $L(v)$ of colors for every $v \in V(G)$.

Problem: Is there a proper list coloring for G?

PRE-COLORING EXTENSION WITH CLIQUE MODULATOR parameterized by k

Input: A graph G, a clique modulator D of size k for G, and a pre-coloring $\lambda_P : X \to Q$ for $X \subseteq V(G)$ where Q is a set of colors.

Problem: Can λ_P be extended to a proper coloring of G using only colors from Q?

In Section 3 we show that LIST COLORING WITH CLIQUE MODULATOR is FPT. We first show a randomized $\mathcal{O}^*(2^{k \log k})$-time algorithm, then we improve the running time to $\mathcal{O}^*(2^k)$ using more refined tools and approaches. Note that all our randomized algorithms are one-sided error algorithms having a constant probability of being wrong, when the algorithm outputs no.

We note that the time $\mathcal{O}^*(2^k)$ matches the best known running time of $\mathcal{O}^*(2^n)$ for CHROMATIC NUMBER (where $n = |V(G)|$) [6], while applying to a more powerful parameter. It is a long-open problem whether CHROMATIC NUMBER can be solved in time $\mathcal{O}(2^{cn})$ for some $c < 1$ and Cygan et al. [15] ask whether it is possible to show that such algorithms are impossible assuming the Strong Exponential Time Hypothesis (SETH).

We conclude Section 3 by showing that LIST COLORING WITH CLIQUE MODULATOR does not admit a polynomial kernel unless NP $\subseteq$ coNP/poly. The reduction used to prove this result allows us to observe that if LIST COLORING WITH CLIQUE MODULATOR could be solved in time $\mathcal{O}(2^{ck} n^{\mathcal{O}(1)})$ for some $c < 1$, then the well-known SET COVER problem could be solved in time $\mathcal{O}(2^{c|U|} |\mathcal{F}|^{\mathcal{O}(1)})$, where U and $\mathcal{F}$ are universe and family of subsets, respectively. The existence of such an algorithm is open, and it has been conjectured that no such algorithm is possible under SETH; see Cygan et al. [15]. Thus, up to the assumption of this conjecture (called *Set Cover Conjecture* [27]) and SETH, our $\mathcal{O}^*(2^k)$-time algorithm for LIST COLORING WITH CLIQUE MODULATOR is best possible w.r.t. its dependency on k.

In Section 4, we consider PRE-COLORING EXTENSION WITH CLIQUE MODULATOR, which is a subproblem of LIST COLORING WITH CLIQUE MODULATOR and prove that PRE-COLORING EXTENSION WITH CLIQUE MODULATOR, unlike LIST COLORING WITH CLIQUE MODULATOR, admits a polynomial kernel: a linear kernel with at most $3k$ vertices. This kernel builds on a known, but counter-intuitive property of bipartite matchings (see Proposition 2.2), which was previously used in kernelization by Bodlaender et al. [8].

In Section 5, we study an open problem stated by Banik et al. [3]. In a classic result, Chor et al. [12] showed that COLORING has a linear vertex kernel parameterized by $k = n-p$, i.e., if the task is to "save k colors". Arora et al. [2] consider the following as a natural extension to list coloring, and show that it is in XP. Banik et al. [3] show that the problem is FPT, but leave as an open question whether it admits a polynomial kernel.

$(n-k)$-REGULAR LIST COLORING parameterized by k

Input: A graph G on n vertices, an integer k, and a list $L(v)$ of exactly $n-k$ colors for every $v \in V(G)$.

Problem: Is there a proper list coloring for G?

We answer this question in affirmative by giving a kernel with $\mathcal{O}(k^2)$ vertices and colors, as well as a compression to a variation of the problem with $\mathcal{O}(k)$ vertices, encodable in $\mathcal{O}(k^2 \log k)$ bits. We note that this compression is asymptotically almost tight, as even 4-COLORING does not admit a compression into $\mathcal{O}(n^{2-\varepsilon})$ bits for any $\varepsilon > 0$ unless the polynomial hierarchy collapses [24].

This kernel is more intricate than the above. Via known reduction rules from Banik et al. [3], we can compute a clique modulator of at most $2k$ vertices (hence our result for LIST COLORING WITH CLIQUE MODULATOR also solves $(n-k)$-REGULAR LIST COLORING in $2^{O(k)}$ time). However, the usual "crown rules" (as in [12] and in Section 4) are not easily applied here, due to complications with the color lists. Instead, we are able to show a set of $\mathcal{O}(k)$ vertices whose colorability make up the "most interesting" part of the problem, leading to the above-mentioned compression and kernel.

In Section 6, we consider further natural pre-coloring and list coloring variants of the "saving k colors" problem of Chor et al. [12]. We show that the known fixed-parameter tractability and linear kernelizability [12] carries over to a natural pre-coloring generalization but fails for a more general list coloring variant. Since $(n-k)$-REGULAR LIST COLORING was originally introduced in [2] as a list coloring variant of the "saving k colors" problem, it is natural to consider other such variants. We conclude the paper in Section 7, where in particular a number of open questions are discussed.

2. Preliminaries.

2.1. Parameterized Complexity. An instance of a parameterized problem Π is a pair (I, k) where I is the *main part* and k is the *parameter*; the latter is usually a non-negative integer. A parameterized problem is *fixed-parameter tractable* (FPT) if there exists a computable function f such that instances (I, k) can be solved in time $\mathcal{O}(f(k)|I|^c)$ where $|I|$ denotes the size of I and c is an absolute constant. The class of all fixed-parameter tractable decision problems is called FPT and algorithms which run in the time specified above are called FPT algorithms. As in other literature on FPT algorithms, we will often omit the polynomial factor in $\mathcal{O}(f(k)|I|^c)$ and write $\mathcal{O}^*(f(k))$ instead. To establish that a problem under a specific parameterization is not in FPT we prove that it is W[1]-hard as it is widely believed that FPT≠W[1].

A *reduction rule* R for a parameterized problem Π is an algorithm A that given an instance (I,k) of a problem Π returns an instance (I',k') of the *same* problem. The reduction rule is said to be *safe* if it holds that $(I,k) \in \Pi$ if and only if $(I',k') \in \Pi$. If A runs in polynomial time in $|I| + k$ then R is a *polynomial-time reduction rule*. Often we omit the adjectives "safe" and "polynomial-time" in "safe polynomial-time reduction rule" as we consider only such reduction rules.

A *kernelization* (or, a *kernel*) of a parameterized problem Π is a reduction rule such that $|I'|+k' \le f(k)$ for some computable function f. It is not hard to show that a decidable parameterized problem is FPT if and only if it admits a kernel [14, 18, 20]. The function f is called the *size* of the kernel, and we have a *polynomial kernel* if $f(k)$ is polynomially bounded in k.

A kernelization can be generalized by considering a reduction (rule) from a parameterized problem Π to another parameterized problem Π'. Then instead of a kernel we obtain a *generalized kernel* (also called a bikernel [1] in the literature). If the problem Π' is not parameterized, then a reduction from Π to Π' (i.e., (I,k) to I') is called a *compression*, which is *polynomial* if $|I'| \le p(k)$, where p is a fixed polynomial in k. If there is a polynomial compression from Π to Π' and Π' is polynomial-time reducible back to Π, with a reduction $I' \mapsto (I,k)$ such that furthermore $k \le |I'|^{O(1)}$, then combining the compression with the reduction gives a polynomial kernel for Π.

2.2. Graphs, Matchings, and Clique Modulator. We consider finite simple undirected graphs. For basic terminology on graphs, we refer to a standard textbook [16]. For an undirected graph $G = (V,E)$ we denote by $V(G)$ the vertex set of G and by $E(G)$ the edge set of G. For a vertex $v \in V(G)$, we denote by $N_G(v)$ and $N_G[v]$ the *open* respectively *closed neighborhood* of v in G, i.e., $N_G(v) := \{\, u \mid \{u,v\} \in E(G) \,\}$ and $N_G[v] := N_G(v) \cup \{v\}$. We extend this notion in the natural manner to subsets $V' \subseteq V(G)$, by setting $N_G(V') := \bigcup_{v \in V'} N_G(v)$ and $N_G[V'] := \bigcup_{v \in V'} N_G[v]$. Moreover, we omit the subscript G, if the graph G can be inferred from the context. If $V' \subseteq V(G)$, we denote by $G \setminus V'$ the graph obtained from G after deleting all vertices in V' together with their adjacent edges and we denote by $G[V']$ the graph induced by the vertices in V', i.e., $G[V'] = G \setminus (V(G) \setminus V')$. We say that G is *bipartite with bi-partition* (A,B), if $\{A,B\}$ partitions $V(G)$ and $G[A]$ as well as $G[B]$ have no edges.

A *matching* M is a subset of $E(G)$ such that no two edges in M share a common endpoint. We say that M is *maximal* if there is no edge $e \in E(G)$ such that $M \cup \{e\}$ is a matching and we say that M is *maximum* if it is maximal and there is no maximal matching in G containing more edges than M. We denote by $V(M)$ the set of all endpoints of the edges in M, i.e., the set $\bigcup_{e \in M} e$. We say that M *saturates* a subset $V' \subseteq V(G)$ if $V' \subseteq V(M)$. Let $H = (V,E)$ be an undirected bipartite graph with bi-partition (A,B). We say that a set C is a *Hall set* for A or B if $C \subseteq A$ or $C \subseteq B$, respectively, and $|N_H(C)| < |C|$. We will need the following well-known properties for matchings.

PROPOSITION 2.1 (Hall's Theorem [16]). *Let G be an undirected bipartite graph with bi-partition (A,B). Then G has a matching saturating A if and only if there is no Hall set for A, i.e., for every $A' \subseteq A$, it holds that $|N(A')| \ge |A'|$.*

PROPOSITION 2.2 ([8, Theorem 2]). *Let G be a bipartite graph with bi-partition (X,Y) and let X_M be the set of all vertices in X that are endpoints of a maximum matching M of G. Then, for every $Y' \subseteq Y$, it holds that G contains a matching that* *covers Y' if and only if so does $G[X_M \cup Y]$.*

Clique Modulator Let G be an undirected graph. We say that a set $D \subseteq V(G)$ is a *clique modulator* for G if $G - D$ is a clique. Since we will use the size of a smallest clique modulator as a parameter for our coloring problems, it is natural to ask whether the following problem can be solved efficiently.

CLIQUE MODULATOR parameterized by k	
Input:	A graph G and an integer k
Problem:	Does G have a clique modulator of size at most k?

The following proposition shows that this is indeed the case. Namely, CLIQUE MODULATOR is both FPT and can be approximated within a factor of two. The former is important for our FPT algorithms and the later for our kernelization algorithms as it allows us to not depend on a clique modulator given as part of the input.

PROPOSITION 2.3. CLIQUE MODULATOR *is fixed-parameter tractable (in time $\mathcal{O}^*(1.2738^k)$) and can be approximated within a factor of two.*

Proof. It is straightforward to verify that a graph G has a clique modulator of size at most k if and only if the complement $\overline{G}$ of G has a vertex cover of size at most k. The statement now follows from the fact that the vertex cover problem is fixed-parameter tractable [11] (in time $\mathcal{O}^*(1.2738^k)$) and can be approximated within a factor of two [21]. ∎

2.3. Polynomial sieving. Algorithms based on polynomial sieving and similar algebraic techniques have become an important component of the toolbox for parameterized and exact algorithms. One of the early examples within the field is the algorithm for computing CHROMATIC NUMBER in time $\mathcal{O}^*(2^n)$ by Björklund et al. [6]. Further developments include techniques such as *multilinear detection* [26] (see also [7]). We review only what we need for this paper; for more background and further techniques, see [15, 26, 7, 5].

For a positive integer p, $[p]$ denotes the set $\{1, 2, \ldots, p\}$. For a polynomial P, we denote the coefficient of a monomial T of P by $\text{coef}_P T$.

The following lemma is central to the approach.

LEMMA 2.4. *(Schwartz-Zippel [32, 36]). Let $P(x_1, \ldots, x_n)$ be a multivariate polynomial of total degree at most d over a field $\mathbb{F}$, and assume that P is not identically zero. Pick $r_1, \ldots, r_n$ uniformly at random from $\mathbb{F}$. Then $Pr[P(r_1, \ldots, r_n) = 0] \leq d/|\mathbb{F}|$.*

The general approach is to construct a polynomial whose terms enumerate potential solutions, and then use sieving techniques over the polynomial to ensure that undesired solutions cancel and only actual solutions remain. As long as the sieved polynomial can be evaluated in FPT time, this then gives a randomized FPT algorithm using the Schwartz-Zippel lemma, as above. In the case that we are working over a field of characteristic 2, we will implicitly assume that the field is large enough to allow an application of the above lemma with good success probability, e.g., by moving to an extension field or starting with a large enough field $\text{GF}(2^\ell)$.

We will use the following simple inclusion-exclusion based sieving technique, previously used by Wahlström [34]. Let $P(x_1, \ldots, x_n)$ be a polynomial and $I \subseteq [n]$ a set of indices. Define $P_{-I}(x_1, \ldots, x_n) = P(y_1, \ldots, y_n)$, where $y_i = 0$ for $i \in I$ and $y_i = x_i$ otherwise. Then the following holds. (The variant for a field of characteristic 2 was proved by Wahlström [34]. The other variant can be proved similarly.)

LEMMA 2.5. *Let $P(x_1, \ldots, x_n)$ be a polynomial over a field of characteristic two (over reals, respectively), and $J \subseteq [n]$ a set of indices. Define*

$$Q(x_1, \ldots, x_n) = \sum_{I \subseteq J} P_{-I}(x_1, \ldots, x_n)$$

$$(Q(x_1, \ldots, x_n) = \sum_{I \subseteq J} (-1)^{|I|} P_{-I}(x_1, \ldots, x_n), \text{ respectively}).$$

Then for any monomial T divisible by $\Pi_{i \in J} x_i$ we have $\mathrm{coef}_Q T = \mathrm{coef}_P T$, and for every other monomial T we have $\mathrm{coef}_Q T = 0$.

We will also use the connection between permanents and bipartite matchings. Let G be a bipartite graph with balanced bi-partition (U, V), i.e., $|U| = |V|$. The bipartite adjacency matrix of G is a matrix A, with rows are indexed by U and columns indexed by V, such that $A[u, v] = 1$ for $u \in U$, $v \in V$ if $uv \in E(G)$, and $A[u, v] = 0$ otherwise. It is well known that the permanent $\mathrm{per}\, A$ enumerates perfect matchings of G, but that it is hard to evaluate in general. The exception is in fields of characteristic 2, where it coincides with the determinant, but where we furthermore have to worry about cancellations due to the characteristic.

In order to work with determinants instead of the permanent, we define the following. The *Edmonds matrix* A of G is defined as the bipartite adjacency matrix, except every non-zero entry $A[u, v] = 1$ is replaced by a distinct variable $A[u, v] = y_{uv}$. Letting $Y = \{y_{uv} \mid uv \in E(G)\}$, we see that $\det A$ is a polynomial in Y of degree $n = |U|$. We extend this to the case when G is a bipartite multigraph. Let $Y = \{y_e \mid e \in E(G)\}$ as above, and, if G contains d edges $e_1, \ldots, e_d$ between u and v for $u \in U$, $v \in V$, then we let $A[u, v] = \sum_{i=1}^{d} y_{e_i}$. In both cases, if we view $\det A$ as a polynomial in Y, then the monomials of $\det A$ are in bijection with the perfect matchings of G. Now the Schwartz-Zippel lemma allows us to test for perfect matchings via a randomized evaluation of $\det A$. Furthermore, given a set of edge weights $w(e)$ for edges of G, we define the *weighted Edmonds matrix* in the same way as the Edmonds matrix, except every occurrence of a variable y_e for an edge $e \in E(G)$ is replaced by $w(e) y_e$. In the case where the weights $w(e)$ are themselves polynomials, in a set of further variables X, this allows us to use Lemma 2.5 with $P(X, Y) = \det A$ to sieve in FPT time for particular kinds of matchings in G. See Theorem 3.1 for an example.

3. List Coloring with Clique Modulator.

We are ready to prove the first result of this section.

THEOREM 3.1. LIST COLORING WITH CLIQUE MODULATOR *can be solved by a randomized algorithm in time $\mathcal{O}^*(2^{k \log k})$.*

Proof. Let $L = \bigcup_{v \in V(G)} L(v)$ and $C = G - D$. We say that a proper list coloring λ for G is compatible with $(\mathcal{D}, \mathcal{D}')$ if:

- $\mathcal{D} = \{D_1, \ldots, D_p\}$ is the partition of all vertices in D that do not reuse colors used by λ in C into color classes given by λ and
- $\mathcal{D}' = \{D'_1, \ldots, D'_t\}$ is the partition of all vertices in D that do reuse colors used by λ in C into color classes given by λ.

Note that $\{D_1, \ldots, D_p, D'_1, \ldots, D'_t\}$ is the partition of D into color classes given by λ.

For a given pair $(\mathcal{D}, \mathcal{D}')$, where each set D_i and D'_i is independent in G, we will now construct a bipartite multigraph B (with weights on its edges) such that B has a perfect matching satisfying certain additional properties if and only if G has a proper list coloring that is compatible with $(\mathcal{D}, \mathcal{D}')$. B has bi-partition $(C \cup \{D_1, \ldots, D_p\}, L)$ and edges as follows. Let $c \in C$ and $\ell \in L$ be such that $\ell \in L(c)$. Then B contains an edge $e_{c\ell}$ between c and ℓ. Furthermore, for every $j \in [t]$ there is a further edge $e_{c\ell,j}$ between c and ℓ if and only if $\ell \in (\bigcap_{d \in D'_j} L(d)) \cap L(c)$ and c is not adjacent to any vertex in D'_j. Moreover, B has an edge between a vertex D_i and a vertex $\ell \in L$ if and only if $\ell \in \bigcap_{d \in D_i} L(d)$. Finally, if $|C| + p > |L|$ then λ cannot exist and we have a no-instance. Otherwise, we add $|L| - |C| - p$ dummy vertices to the partite set $C \cup \{D_1, \ldots, D_p\}$ and make the dummy vertices adjacent to all vertices in L.

For weights, we introduce a new set of variables $X = \{x_1, \ldots, x_t\}$, and for every edge $e_{c\ell,j}$ created above we set $w(e_{c\ell,j}) = x_j$. Every other edge e of B has weight $w(e) = 1$. For an illustration of B, see Figure 1.

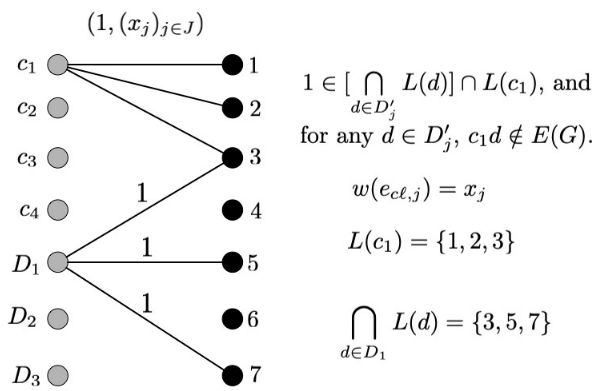

FIG. 1. *Illustration of the construction of B. $(1, (x_j)_{j \in J})$ means that there are $1 + |J|$ parallel edges between c_1 and 1 with weights $1, x_{j_1}, x_{j_2}, \ldots, x_{j_{|J|}}$, where $J = \{j_1, j_2, \ldots, j_{|J|}\}$.*

Note that G has a proper list coloring that is compatible with $(\mathcal{D}, \mathcal{D}')$ if and only if B has a perfect matching F such that there is a bijection α between $[t]$ and t edges in F such that for every $i \in [t]$, the weight of the edge $\alpha(i)$ is x_i. Indeed, we have $w(\alpha(i)) = x_i$ if and only if $\alpha(i) = e_{c\ell,i}$ for some vertices c and ℓ, which in turn implies that $D'_i \cup \{c\}$ is an independent set in G and $\ell \in L(u)$ for every $u \in D'_i \cup \{c\}$. Along with the further edges of F of weight 1, this defines a proper coloring λ for G which is compatible with $(\mathcal{D}, \mathcal{D}')$.

Let M be the weighted Edmonds matrix of B with weights w (see Section 2.3), for simplicity constructed over a field of characteristic 2. Let $Y = \{y_e \mid e \in E(B)\}$ be the set of further variables introduced in the construction of M. Then $\det M$ is a polynomial in variables $X \cup Y$, and as discussed in Section 2.3, the monomials of $\det M$ are in bijection with perfect matchings of B; in particular, the latter holds since every weight $w(e)$ defined above is a single monomial. Furthermore, for every perfect matching F of B, the monomial of $\det M$ corresponding to F equals $\prod_{e \in F} w(e) y_e$.

Now it is not hard to see that $\det M$ has a monomial containing $\prod_{j=1}^{t} x_j$ if and only if B has a perfect matching F such that there is a bijection α between $[t]$ and t edges in F such that for every $i \in [t]$, the weight of the edge $\alpha(i)$ is x_i, which in turn is equivalent to G having a proper list coloring that is compatible with $(\mathcal{D}, \mathcal{D}')$. Note that the other $|C| - t$ edges of the form $c\ell$ contribute a factor 1 to the monomial, as do the edges of the form $D_i \ell$.

Hence, deciding whether G has a proper list coloring that is compatible with $(\mathcal{D}, \mathcal{D}')$ boils down to deciding whether $\det M$ has a monomial containing $\prod_{j=1}^{t} x_j$. For any evaluation of variables X and Y, we can compute $\det M$ in polynomial-time [9].

Now write $y = (y_1, \ldots, y_m)$, and let $P(x_1, \ldots, x_t, y) = \det M$. Define

$$Q(x_1, \ldots, x_t) = \sum_{I \subseteq [t]} P_{-I}(x_1, \ldots, x_t, y).$$

Note that each of P and Q is of degree at most $2n$.

By Lemma 2.5, $Q(x_1, \ldots, x_t) \neq 0$ if and only if $\det M$ has a monomial containing $\prod_{j=1}^{t} x_j$. Moreover, using Lemmas 2.4 and 2.5 (with P and Q just defined), we can verify with a single evaluation of Q whether $Q(x_1, \ldots, x_t) = 0$ (i.e. whether $\det M$ contains a monomial containing $\prod_{j=1}^{t} x_j$) with probability at least $1 - \frac{2n}{|\mathbb{F}|} \geq 2/3$ for a field $\mathbb{F}$ of characteristic 2 such that $|\mathbb{F}| \geq 6n$. Furthermore, Q can be evaluated in time $\mathcal{O}^*(2^t)$.

Our algorithm sets $t = k$ and for every pair $(\mathcal{D}, \mathcal{D}')$, where $\mathcal{D} \cup \mathcal{D}'$ is a partition of D into independent sets, constructs the graph B and matrix M. It then verifies in time $\mathcal{O}^*(2^t)$ whether $Q(x_1, \ldots, x_t, y_1, \ldots, y_m) = 0$, and if $Q(x_1, \ldots, x_t, y_1, \ldots, y_m) \neq 0$ it returns 'Yes' and terminates. If the algorithm runs to the end, it returns 'No'.

Note that the time complexity of the algorithm is dominated by the number of choices for $(\mathcal{D}, \mathcal{D}')$, which is in turn dominated by $\mathcal{O}^*(\mathcal{B}_k)$, where $\mathcal{B}_k$ is the k-th Bell number. By Berend and Tassa [4], $\mathcal{B}_k < (\frac{0.792k}{\ln(k+1)})^k$, and thus the total running time is $\mathcal{O}^*(\mathcal{B}_k 2^k) = \mathcal{O}^*(2^{k \log k})$. ◻

3.1. A faster FPT algorithm. We now show a faster FPT algorithm, running in time $\mathcal{O}^*(2^k)$. It is a variation on the same algebraic sieving technique as above, but instead of guessing a partition of the modulator it works over a more complex matrix. We begin by defining the matrix, then we show how to perform the sieving step in $\mathcal{O}^*(2^k)$ time.

3.1.1. Matrix definition. As before, let $L = \bigcup_{v \in V(G)} L(v)$ be the set of all colors, and let $C = G - D$. Define an auxiliary bipartite graph $H = (U_H \cup V_H, E_H)$ where initially $U_H = V(G)$ and $V_H = L$, and where $v\ell \in E_H$ for $v \in V(G)$, $\ell \in L$ if and only if $\ell \in L(v)$. Additionally, introduce a set $L' = \{\ell'_d \mid d \in D\}$ of k artificial colors, add L' to V_H, and for each $d \in D$ connect ℓ'_d to d but to no other vertex. Finally, pad U_H with $|V_H| - |U_H|$ artificial vertices connected to all of V_H; note that this is a non-negative number, since otherwise $|L| < |V(C)|$ and we may reject the instance.

Next, we associate with every edge $v\ell \in E_H$ a set $S(v\ell) \subseteq 2^D$ as follows.

- If $v \in V(C)$, then $S(v\ell)$ contains all sets $S \subseteq D$ such that the following hold:
  1. S is an independent set in G
  2. $N(v) \cap S = \emptyset$
  3. $\ell \in \bigcap_{s \in S} L(s)$.
- If $v \in D$ and $\ell \in L$, then $S(v\ell)$ contains all sets $S \subseteq D$ such that the following hold:
  1. $v \in S$
  2. S is an independent set in G
  3. $\ell \in \bigcap_{s \in S} L(s)$.
- If v or ℓ is an artificial vertex – in particular, if $\ell = \ell'_d$ for some $d \in D$ – then $S(v\ell) = \{\emptyset\}$.

Finally, define a matrix A of dimensions $|U_H| \times |V_H|$, with rows labelled by U_H and columns labelled by V_H, whose entries are polynomials as follows. Define a set of variables $X = \{x_d \mid d \in D\}$ corresponding to vertices of D, and additionally a set $Y = \{y_e \mid e \in E_H\}$. Then for every edge $v\ell$ in H, $v \in U_H$, $\ell \in V_H$ we define

$$P(v\ell) = \sum_{S \in S(v\ell)} \prod_{s \in S} x_s,$$

where as usual an empty product equals 1. Then for each edge $v\ell \in E_H$ we let $A[v,\ell] = y_{v\ell}P(v\ell)$, and the remaining entries of A are 0. We argue the following. (Expert readers may note although the argument can be sharpened to show the existence of a multilinear term, we do not wish to argue that there exists such a term with odd coefficient. Therefore we use the simpler sieving of Lemma 2.5 instead of full multilinear detection, cf. [14].)

LEMMA 3.2. *Let A be defined as above. Then* $\det A$ *(as a polynomial) contains a monomial divisible by* $\prod_{x \in X} x$ *if and only if* G *is properly list colorable.*

Proof. We first note that no cancellation happens in $\det A$. Note that monomials of $\det A$ correspond (many-to-one) to perfect matchings of H, and thanks to the formal variables Y, two monomials corresponding to distinct perfect matchings never interact. On the other hand, if we fix a perfect matching M in H, then the contributions of M to $\det A$ equal $\sigma_M \prod_{e \in M} y_e P(e)$, where $\sigma_M \in \{1, -1\}$ is a sign term depending only on M. Since the polynomials $P(e)$ contain only positive coefficients, no cancellation occur, and every selection of a perfect matching M of H and a factor from every polynomial $P(e)$, $e \in M$ results (many-to-one) to a monomial with non-zero coefficient in $\det A$.

We now proceed with the proof. On the one hand, let c be a proper list coloring of G. Define an ordering $\prec$ on $V(G)$ such that $V(C)$ precedes D, and define a matching M as follows. For every vertex $v \in V(C)$, add $vc(v)$ to M. For every vertex $v \in D$, add $vc(v)$ to M if v is the first vertex according to $\prec$ that uses color $c(v)$, otherwise add $v\ell'_v$ to M. Note that M is a matching in H of $|V(G)|$ edges. Pad M to a perfect matching in H by adding arbitrary edges connected to the artificial vertices in U_H; note that this is always possible. Finally, for every edge $v\ell \in M$ with $\ell \in L$ we let $D_{v\ell} = D \cap c^{-1}(\ell)$. Observe that for every edge $v\ell$ in M, $D_{v\ell} \in S(v\ell)$; indeed, this holds by construction of $S(v\ell)$ and since c is a proper list coloring. Further let $p_{v\ell} = \prod_{v \in D_{v\ell}} x_v$; thus $p_{v\ell}$ is a term of $P(v\ell)$. It follows, by the discussion in the first paragraph of the proof, that

$$\alpha\sigma_M \prod_{v\ell \in M} y_{v\ell}p_{v\ell}$$

is a monomial of $\det A$ for some constant $\alpha > 0$, where $\sigma_M \in \{1, -1\}$ is the sign term for M. It remains to verify that every variable $x_d \in X$ occurs in some term $p_{v\ell}$. Let $\ell = c(d)$ and let v be the earliest vertex according to $\prec$ such that $c(v) = \ell$. Then $v\ell \in M$ and x_d occurs in $p_{v\ell}$. This finishes the first direction of the proof.

On the other hand, assume that $\det A$ contains a monomial T divisible by $\prod_{x \in X} x$, and let M be the corresponding perfect matching of H. Let $T = \alpha \prod_{e \in M} y_e p_e$ for some constant factor α, where p_e is a term of $P(e)$ for every $e \in M$. Clearly such a selection is possible; if it is ambiguous, make the selection arbitrarily. Now define a mapping $c\colon V(G) \to L$ as follows. For $v \in V(C)$, let $v\ell \in M$ be the unique edge connected to v, and set $c(v) = \ell$. For $v \in D$, let v' be the earliest vertex according to $\prec$ such that x_v occurs in $p_{v'\ell}$, where $v'\ell \in M$. Set $c(v) = \ell$. We verify that c is a proper list coloring of G. First of all, note that $c(v)$ is defined for every $v \in V(G)$ and that $c(v) \in L(v)$. Indeed, if $v \in V(C)$ then $c(v) \in L(v)$ since $vc(v) \in E_H$; and if $v \in D$ then $c(v) \in L(v)$ is verified in the creation of the term $p_{vc(v)}$ in $P(vc(v))$. Next, consider two vertices $u, v \in V(G)$ with $c(u) = c(v)$. If $u, v \in D$, then u and v are represented in the same term $p_{v'c(v)}$ for some v', hence u and v form an independent set; otherwise assume $u \in V(C)$. Note that $u, v \in V(C)$ is impossible since otherwise the matching M would contain two edges $uc(u)$ and $vc(u)$ which intersect. Thus $v \in D$, and v is represented in the term $p_{uc(u)}$. Therefore $uv \notin E(G)$, by construction of $P(uc(u))$. We conclude that c is a proper coloring respecting the lists $L(v)$, i.e., a proper list coloring. ∎

3.1.2. Fast evaluation. By the above description, we can test for the existence of a list coloring of G using 2^k evaluations of $\det A$, as in Theorem 3.1; and each evaluation can be performed in $\mathcal{O}^*(2^k)$ time, including the time to evaluate the polynomials $P(v\ell)$, making for a running time of $\mathcal{O}^*(4^k)$ in total (or $\mathcal{O}^*(3^k)$ with more careful analysis). We show how to perform the entire sieving in time $\mathcal{O}^*(2^k)$ using fast subset convolution.

For $I \subseteq D$, let us define A_{-I} as A with all occurrences of variables x_i, $i \in I$ replaced by 0, and for every edge $v\ell$ of H, let $P(v\ell)_{-I}$ denote the polynomial $P(v\ell)$ with x_i, $i \in I$ replaced by 0. Then a generic entry (v, ℓ) of A_{-I} equals

$$A_{-I}[v, \ell] = y_{v\ell} P_{-I}(v\ell),$$

and in order to construct A_{-I} it suffices to pre-compute the value of $P_{-I}(v\ell)$ for every edge $v\ell \in E_H$, $I \subseteq D$. For this, we need the *fast zeta transform* of Yates [35], which was introduced to exact algorithms by Björklund et al. [6].

Lemma 3.3 ([35, 6]). *Given a function $f : 2^N \to R$ for some ground set N and ring R, we may compute all values of $\hat{f} : 2^N \to R$ defined as $\hat{f}(S) = \sum_{A \subseteq S} f(A)$ using $\mathcal{O}^*(2^{|N|})$ ring operations.*

We show the following lemma, which is likely to have analogues in the literature, but we provide a short proof for the sake of completeness.

Lemma 3.4. *Given an evaluation of the variables X, the value of $P_{-I}(v\ell)$ can be computed for all $I \subseteq D$ and all $v\ell \in E_H$ in time and space $\mathcal{O}^*(2^k)$.*

Proof. Consider an arbitrary polynomial $P_{-I}(v\ell)$.
Recalling $P(v\ell) = \sum_{S \in S(v\ell)} \prod_{s \in S} x_s$, we have:

$$P_{-I}(v\ell) = \sum_{S \in S(v\ell)} [S \cap I = \emptyset] \prod_{s \in S} x_s = \sum_{S \subseteq (D-I)} [S \in S(v\ell)] \prod_{s \in S} x_s,$$

using Iverson bracket notation.[1] Using $f(S) = [S \in S(v\ell)] \prod_{s \in S} x_s$, this clearly fits the form of Lemma 3.3, with $\hat{f}(D - I) = P_{-I}(v\ell)$. Hence we apply Lemma 3.3 for every edge $v\ell \in E_H$, for $\mathcal{O}^*(2^k)$ time per edge, making $\mathcal{O}^*(2^k)$ time in total to compute all values. ∎

Having access to these values, it is now easy to complete the algorithm.

Theorem 3.5. List Coloring with Clique Modulator *can be solved by a randomized algorithm in time $\mathcal{O}^*(2^k)$.*

[1] Recall that for a logical proposition P, $[P] = 1$ if P is true and 0, otherwise.

Proof. Let A be the matrix defined above (but do not explicitly construct it yet). By Lemma 3.2, we need to check whether $\det A$ contains a monomial divisible by $\prod_{x \in X} x$, and by Lemma 2.5 this is equivalent to testing whether

$$\sum_{I \subseteq D} (-1)^{|I|} \det A_{-I} \not\equiv 0.$$

By the Schwartz-Zippel lemma (Lemma 2.4), it suffices to randomly evaluate the variables X and Y occurring in A and evaluate this sum once; if G has a proper list coloring and if the values of X and Y are chosen among sufficiently many values, then with high probability the result is non-zero, and if not, then the result is guaranteed to be zero. Thus the algorithm is as follows.

1. Instantiate variables of X and Y uniformly at random from $[N]$ for some sufficiently large N. Note that for an error probability of ε with $0 < \varepsilon < 1$, it suffices to use $N = \Omega(n^2(1/\varepsilon))$.
2. Use Lemma 3.4 to fill in a table with the value of $P_{-I}(v\ell)$ for all I and $v\ell$ in time $\mathcal{O}^*(2^k)$.
3. Compute
$$\sum_{I \subseteq D} (-1)^{|I|} \det A_{-I},$$
constructing A_{-I} from the values $P_{-I}(v\ell)$ in polynomial time in each step.
4. Answer YES if the result is non-zero, NO otherwise.

Clearly this runs in total time and space $\mathcal{O}^*(2^k)$ and the correctness follows from the arguments above. ∎

3.2. Refuting Polynomial Kernel.

In this section, we prove that List Coloring with Clique Modulator does not admit a polynomial kernel. We prove this result by a polynomial parameter transformation from Hitting Set where the parameter is the number of sets, which is known not to have a polynomial kernel [17]. Notice that Hitting Set parameterized by number of sets is equivalent to Set Cover parameterized by the universe size.

Theorem 3.6. List Coloring with Clique Modulator *parameterized by k does not admit a polynomial kernel unless NP $\subseteq$ coNP/poly.*

Proof. Let us recall the formal definition of the Hitting Set problem.

Hitting Set parameterized by m

Input:	A universe U of n elements, a family $\mathcal{F} \subseteq 2^U$ of m subsets of U, and an integer k.
Problem:	Is there $X \subseteq U$ with at most k elements such that for every $F \in \mathcal{F}$, it holds that $F \cap X \neq \emptyset$?

Let $(U, \mathcal{F}, k)$ be an instance of Hitting Set problem where $U = [n]$, and $\mathcal{F} = \{F_1, \ldots, F_m\}$. Now, we are ready to describe the construction.

Construction: For every $i \in [m]$, we create a vertex u_i and assign $L(u_i) = F_i$. Let $D = \{u_1, \ldots, u_m\}$. In addition, we create a clique C with $n - k$ vertices $\{v_1, \ldots, v_{n-k}\}$. Moreover, for every $j \in [n-k]$, we set $L(v_j) = U$ and for all $i \in [n-k]$ and $j \in [m]$, let (u_i, v_j) be an edge. This completes the construction, which takes polynomial time. We denote the obtained graph by G. It remains to show that the two instances are equivalent.

Towards showing the forward direction, let $(U, \mathcal{F}, k)$ be an yes-instance. Then, there is a set X of at most k elements from U such that for every $F_i \in \mathcal{F}$, $X \cap F_i \neq \emptyset$.

Using the elements present in X, we can color D as follows. We pick an element arbitrarily from every $F_i \cap X$, and color the vertex u_i using that color. After that, we provide different colors to different vertices in C that are different from the colors used in D as well. Hence, we can color G by n colors.

Towards showing the backwards direction, suppose that G has a proper list coloring. Note that all vertices of C have to get different colors. Hence, the vertices of D must be colorable using only k colors. Suppose that X is the set of k colors used to color the vertices of D. Note that the colors respect the list for every vertex in D where the list represents the sets in the family. Hence, X is a hitting set of size k. ☐

Note that the reduction also shows that if List Coloring with Clique Modulator could be solved in time $\mathcal{O}(2^{\epsilon k} n^{\mathcal{O}(1)})$ for some $\epsilon < 1$, then Hitting Set could be solved in time $\mathcal{O}(2^{\epsilon |\mathcal{F}|} |U|^{\mathcal{O}(1)})$, which in turn would imply that any instance I with universe U and set family $\mathcal{F}$ of the well-known Set Cover problem could be solved in time $\mathcal{O}(2^{\epsilon |U|} |\mathcal{F}|^{\mathcal{O}(1)})$. The existence of such an algorithm is open, and it has been conjectured that no such algorithm is possible under SETH (the strong exponential-time hypothesis); see Cygan et al. [15]. Thus, up to the assumption of this conjecture and SETH, the algorithm for List Coloring with Clique Modulator given in Theorem 3.5 is best possible w.r.t. its dependency on k.

4. Polynomial kernel for Pre-Coloring Extension with Clique Modulator.

In the following let $(G, D, k, \lambda_P, X, Q)$ be an instance of Pre-Coloring Extension with Clique Modulator, let $C = G - D$, let D_P be the set of all pre-colored vertices in D, and let $D' = D \setminus D_P$. W.l.o.g., we can assume that $|Q| \geq |C|$ as otherwise the instance is a trivial no-instance. In the following, we will assume that the instance will be updated with the introduction of every reduction rule, i.e., we will assume that all already introduced reduction rules have already been exhaustively applied to the current instance.

Reduction Rule 1. *Remove any vertex $v \in D'$ that has less than $|Q|$ neighbors in G.*

The proof of the following lemma is obvious and thus omitted.

Lemma 4.1. *Reduction Rule 1 is safe and can be implemented in polynomial time.*

Note that if Reduction Rule 1 can no longer be applied, then every vertex in D' has at least $|Q|$ neighbors, which because of $|Q| \geq |C|$ implies that every such vertex has at most $|D| \leq k$ non-neighbors in G and hence also in C. Let C_N be the set of all vertices in C that are not adjacent to all vertices in D' and let $C' = C - C_N$. Note that $|C_N| \leq |D||D| \leq k^2$.

We show next how to reduce the size of C_N to k. Note that this step is optional if our aim is solely to obtain a polynomial kernel, however, it allows us to reduce the number of vertices in the resulting kernel from $\mathcal{O}(k^2)$ to $\mathcal{O}(k)$. Let J be the bipartite graph with partition (C_N, D) having an edge between $c \in C_N$ and $d \in D$ if $\{c, d\} \notin E(G)$. Our next reduction rule can be seen as a crown reduction rule that uses a crown decomposition of J with crown A and head $N_H(A)$; a similar rule has been employed previously in [3, Reduction Rule 2].

Reduction Rule 2. *If $A \subseteq C_N$ is an inclusion-wise minimal set satisfying $|A| > |N_J(A)|$, then remove the vertices in $D' \cap N_J(A)$ from G.*

Note that after the application of Reduction Rule 2, the vertices in A are implicitly removed from C_N and added to C' since all their non-neighbors in D' (i.e. the vertices in $D' \cap N_J(A)$) are removed from the graph.

LEMMA 4.2. *Reduction Rule 2 is safe and can be implemented in polynomial time.*

Proof. It is clear that the rule can be implemented in polynomial-time. Towards showing the safeness of the rule, it suffices to show that G has a coloring extending λ_P using only colors from Q if and only if so does $G \setminus (D' \cap N_J(A))$. Since $G \setminus (D' \cap N_J(A))$ is a subgraph of G, the forward direction of this statement is trivial. So assume that $G \setminus (D' \cap N_J(A))$ has a coloring λ extending λ_P using only colors from Q. Because the set A is inclusion-minimal, we obtain from Proposition 2.1, that there is a (maximum) matching, say M, between $N_J(A)$ and A in J that saturates $N_J(A)$. Moreover, it follows from the definition of J that every vertex in A is adjacent to every vertex in G apart from the vertices in $N_J(A)$. Therefore, the colors in $\lambda(A)$ can only reappear in $D_P \cap N_J(A)$. We can now use the matching M to reshuffle the colors in A in such a way that the colors of vertices in A that are matched by M to a vertex in D' appear exactly once in the graph; or in other words we reshuffle the colors in A such that all colors that also appear in $D_P \cap N_J(A)$ are assigned to vertices in A that are matched by M to vertices in D_P. That is, let A' be the set of all vertices a in A with $\lambda(a) \in \lambda(D_P \cap N_J(A))$ such that a is matched by M to a vertex in D'. Similarly, let A_P be the set of all vertices a in A with $\lambda(a) \notin \lambda(D_P \cap N_J(A))$ such that a is matched by M to a vertex in D_P. Note that $|A_P| \geq |A'|$ and therefore there is a bijection $\alpha : A' \to A'_P$ from A' to a subset A'_P of A_P. Now, let λ' be the coloring obtained from λ by setting $\lambda'(a) = \lambda(\alpha(a))$ for every $a \in A'$, $\lambda'(a) = \lambda(\alpha^{-1}(a))$ for every $a \in A'_P$, and $\lambda'(a) = \lambda(a)$ otherwise. Then, the color $\lambda'(a)$ appears exactly once for every $a \in A$ that is matched by M to a vertex in D'. Therefore, we can extend λ' into a coloring λ'' for G by coloring the vertices in $D' \cap N_J(A)$ according to the matching M. More formally, let $\lambda_{D' \cap N_J(A)}$ be the coloring for the vertices in $D' \cap N_J(A)$ obtained by setting $\lambda_{D' \cap N_J(A)}(v) = \lambda'(u)$ for every $v \in D' \cap N_J(A)$, where $\{v, u\} \in M$. Then, we obtain λ'' by setting: $\lambda''(v) = \lambda'(v)$ for every $v \in V(G) \setminus (D' \cap N_J(A))$ and $\lambda''(v) = \lambda_{D' \cap N_J(A)}(v)$ for every vertex $v \in D' \cap N_J(A)$. ∎

Note that because of Proposition 2.1, we obtain that there is a set $A \subseteq C_N$ with $|A| > |N_J(A)|$ as long as $|C_N| > |D|$. Moreover, since $N_J(A) \cap D' \neq \emptyset$ for every such set A (due to the definition of C_N), we obtain that Reduction Rule 2 is applicable as long as $|C_N| > |D|$. Hence after an exhaustive application of Reduction Rule 2, we obtain that $|C_N| \leq |D'| \leq k$.

We now introduce our final two reduction rules, which allow us to reduce the size of C'.

Reduction Rule 3. *Let $v \in V(C')$ be a pre-colored vertex with color $\lambda_P(v)$. Then remove $\lambda_P^{-1}(\lambda_P(v))$, i.e., all vertices colored with the same color ($\lambda_P(v)$) as v, from G and $\lambda_P(v)$ from Q.*

LEMMA 4.3. *Reduction Rule 3 is safe and can be implemented in polynomial time.*

Proof. Because $v \in V(C')$, it holds that only vertices in D_P can have color $\lambda_P(v)$, but these are already pre-colored. Hence in any coloring for G that extends λ_P, the vertices in $\lambda_P^{-1}(\lambda_P(v))$ are the only vertices that obtain color $\lambda_P(v)$, which implies the safeness of the rule. ∎

Because of Reduction Rule 3, we can from now on assume that no vertex in C' is pre-colored. Note that the only part of G, whose size is not yet bounded by a polynomial in the parameter k is C'. To reduce the size of C', we need will make use of Proposition 2.2. Let $P = \lambda_P(D_P)$ and H be the bipartite graph with bi-partition (C', P) containing an edge between $c' \in C'$ and $p \in P$ if and only if c' is not adjacent to a vertex pre-colored by p in G.

Reduction Rule 4. *Let M be a maximum matching in H and let C_M be the endpoints of M in C'. Then remove all vertices in $C_{\overline{M}} := C' \setminus C_M$ from G and remove an arbitrary set of $|C_{\overline{M}}|$ colors from $Q \setminus \lambda_P(X)$. (Recall that $\lambda_P : X \to Q$.)*

In the following let C_M and $C_{\overline{M}}$ be as defined in the above reduction rule for an arbitrary maximum matching M of H. To show that the reduction rule is safe, we need the following auxiliary lemma, which shows that if a coloring for G reuses colors from P in C', then those colors can be reused solely on the vertices in C_M.

LEMMA 4.4. *If there is a coloring λ for G extending λ_P using only colors in Q, then there is a coloring λ' for G extending λ_P using only colors in Q such that $\lambda'(C_{\overline{M}}) \cap P = \emptyset$.*

Proof. Let C_P be the set of all vertices v in C' with $\lambda(v) \in P$. If $C_P \cap C_{\overline{M}} = \emptyset$, then setting λ' equal to λ satisfies the claim of the lemma. Hence assume that $C_P \cap C_{\overline{M}} \neq \emptyset$. Let N be the matching in H containing the edges $\{v, \lambda(v)\}$ for every $v \in C_P$; note that N is indeed a matching in H, because C_P is a clique in G. Because of Proposition 2.2, there is a matching N' in $H[C_M \cup P]$ such that N' has exactly the same endpoints in P as N. Let $C_M[N']$ be the endpoints of N' in C_M and let λ_A be the coloring of the vertices in $C_M[N']$ corresponding to the matching N', i.e., a vertex v in $C_M[N']$ obtains the unique color $p \in P$ such that $\{v, p\} \in N'$. Finally, let α be an arbitrary bijection between the vertices in $(V(N) \cap C') \setminus C_M[N']$ and the vertices in $C_M[N'] \setminus (V(N) \cap C')$, which exists because $|N| = |N'|$. We now obtain λ' from λ by setting $\lambda'(v) = \lambda_A(v)$ for every $v \in C_M[N']$, $\lambda'(v) = \lambda(\alpha(v))$ for every vertex $v \in (V(N) \cap C') \setminus C_M[N']$, and $\lambda'(v) = \lambda(v)$ for every other vertex. To see that λ' is a proper coloring note that $\lambda'(C') = \lambda(C')$. Moreover, all the colors in $\lambda(C') \setminus P$ are "universal colors" in the sense that exactly one vertex of G obtains the color and hence those colors can be freely moved around in C'. Finally, the matching N' in H ensures that the vertices in $C_M[N']$ can be colored using the colors from P. ◻

LEMMA 4.5. *Reduction Rule 4 is safe and can be implemented in polynomial time.*

Proof. Note first that the reduction can always be applied since if $Q \setminus \lambda_P(X)$ contains less than $|C_{\overline{M}}|$ colors, then the instance is a no-instance. It is clear that the rule can be implemented in polynomial time using any polytime algorithm for finding a maximum matching [29]. Moreover, if the reduced graph has a coloring extending λ_P using only the colors in Q, then so does the original graph, since the vertices in $C_{\overline{M}}$ can be colored with the colors removed from the original instance.

Hence, it remains to show that if G has a coloring, say λ, extending λ_P using only colors in Q, then $G \setminus C_{\overline{M}}$ has a coloring extending λ_P that uses only colors in $Q' := Q \setminus Q_{\overline{M}}$, where $Q_{\overline{M}}$ is the set of $|C_{\overline{M}}|$ colors from $Q \setminus \lambda_P(X)$ that have been removed from Q.

Because of Lemma 4.4, we may assume that $\lambda(C_{\overline{M}}) \cap P = \emptyset$. Let B be the set of all vertices v in $G - C_{\overline{M}}$ with $\lambda(v) \in Q_{\overline{M}}$. If $B = \emptyset$, then λ is a coloring extending λ_P using only colors from Q'. Hence assume that $B \neq \emptyset$. Let A be the set of all vertices v in $C_{\overline{M}}$ with $\lambda(v) \in Q'$. Then $\lambda(A) \cap \lambda_P(X) = \emptyset$, which implies that every color in $\lambda(A)$ appears only in $C_{\overline{M}}$ (and exactly once in $C_{\overline{M}}$). Moreover, $|\lambda(A)| \geq |\lambda(B)|$. Let α be an arbitrary bijection between $\lambda(B)$ and an arbitrary subset of $\lambda(A)$ (of size $|B|$) and let λ' be the coloring obtained from λ by setting $\lambda'(v) = \alpha(\lambda(v))$ for every $v \in B$, $\lambda'(v) = \alpha^{-1}(\lambda(v))$ for every $v \in A$, and $\lambda'(v) = \lambda(v)$, otherwise. Then λ' restricted to $G - C_{\overline{M}}$ is a coloring for $G - C_{\overline{M}}$ extending λ_P using only colors from Q'. Note that λ' is a proper coloring because the colors in $\lambda(A)$ are not in P and hence do not appear anywhere else in G and moreover the colors in $\lambda(B)$ do not appear in $\lambda(C_{\overline{M}})$. ∎

Note that after the application of Reduction Rule 4, it holds that $|C'| = |C_M| \leq |P| \leq |D_P| \leq |D| \leq k$. Together with the facts that $|D| \leq k$, $|C_N| \leq k$, we obtain that the reduced graph has at most $3k$ vertices.

THEOREM 4.6. PRE-COLORING EXTENSION WITH CLIQUE MODULATOR *admits a polynomial kernel with at most* $3k$ *vertices.*

5. Polynomial kernel and Compression for $(n-k)$-REGULAR LIST COLORING.

We now show our polynomial kernel and compression for $(n-k)$-REGULAR LIST COLORING, which is more intricate than the one for PRE-COLORING EXTENSION WITH CLIQUE MODULATOR. Let (G, k, L) be an input of $(n-k)$-REGULAR LIST COLORING. We begin by noting that we can assume that G has a clique-modulator of size at most $2k$.

LEMMA 5.1 ([3]). *In polynomial-time either we can either solve* (G, k, L) *or compute a clique-modulator for* G *of size at most* $2k$.

Henceforth, we let $V(G) = C \cup D$ where $G[C]$ is a clique and D is a clique modulator, $|D| \leq 2k$. Let $T = \bigcup_{v \in V(G)} L(v)$. We note one further known reduction rule for $(n-k)$-REGULAR LIST COLORING. Consider the bipartite graph H_G with bi-partition $(V(G), T)$ having an edge between $v \in V(G)$ and $t \in T$ if and only if $t \in L(v)$.

Reduction Rule 5 ([3]). *Let* T' *be an inclusion-wise minimal subset of* T *such that* $|N_{H_G}(T')| < |T'|$, *then remove all vertices in* $N_{H_G}(T')$ *from* G.

Note that after an exhaustive application of Reduction Rule 5, it holds that $|T| \leq |V(G)|$ since otherwise Proposition 2.1 would ensure the applicability of the reduction rule. Hence in the following we will assume that $|T| \leq |V(G)|$.

With this preamble handled, let us proceed with the kernelization. We are not able to produce a direct 'crown reduction rule' for LIST COLORING, as for PRE-COLORING EXTENSION (e.g., we do not know of a useful generalization of Reduction Rule 2). Instead, we need to study more closely which list colorings of $G[D]$ extend to list colorings of G. For this purpose, let $H = H_G - D$ be the bipartite graph with bi-partition (C, T) having an edge $\{c, t\}$ with $c \in C$ and $t \in T$ if and only if $t \in L(c)$. Say that a partial list coloring $\lambda_0 \colon A \to T$ is *extensible* if it can be extended to a proper list coloring λ of G. If $D \subseteq A$, then a sufficient condition for this is that $H - (A \cup \lambda_0(A))$ admits a matching saturating $C \setminus A$. (This is not a necessary condition, since some colors used in $\lambda_0(D)$ could be reused in $\lambda(C \setminus A)$, but this investigation will point in the right direction.) By Proposition 2.1, this is characterized by Hall sets in $H - (A \cup \lambda_0(A))$.

A Hall set $S \subseteq U$ in a bipartite graph G' with bi-partition (U, W) is *trivial* if $N(S) = W$. We start by noting that if a color occurs in sufficiently many vertex lists in H, then it behaves uniformly with respect to extensible partial colorings λ_0 as above.

LEMMA 5.2. *Let* $\lambda_0 \colon A \to T$ *be a partial list coloring where* $|A \cap C| \leq p$ *and let* $t \in T$ *be a color that occurs in at least* $k + p$ *lists in* C. *Then* t *is not contained in any non-trivial Hall set of colors in* $H - (A \cup \lambda_0(A))$.

Proof. Let $H' = H - (A \cup \lambda_0(A))$. Consider any Hall set of colors $S \subset (T \setminus \lambda_0(A))$

and any vertex $v \in C \setminus (A \cup N_{H'}(S))$ (which exists assuming S is non-trivial). Then $S \subseteq T \setminus L(v)$, hence $|S| \leq k$, and by assumption $|N_{H'}(S)| < |S|$. But for every $t' \in S$, we have $N_H(t') \subseteq N_{H'}(S) \cup (A \cap C)$, hence t' occurs in at most $|N_{H'}(S) \cup (A \cap C)| < k+p$ vertex lists in C. Thus $t \notin S$. ∎

In the following, we will assume that $n \geq 11k$.[2] This is safe, since otherwise (by Reduction Rule 5) we already have a kernel with a linear number of vertices and colors. We say that a color $t \in T$ is *rare* if it occurs in at most $6k$ lists of vertices in C.

LEMMA 5.3. *If $n \geq 11k$, then there are at most $3k$ rare colors.*

Proof. Let $S = \{t \in T \mid d_H(t) < 6k\}$. For every $t \in S$, there are $|C| - 6k$ "non-occurrences" (i.e., vertices $v \in C$ with $t \notin L(v)$), and there are $|C|k$ non-occurrences in total. Thus

$$|S| \cdot (|C| - 6k) \leq |C|k \quad \Rightarrow \quad |S| \leq \frac{|C|}{|C| - 6k} k = (1 + \frac{6k}{|C| - 6k})k,$$

where the bound is monotonically decreasing in $|C|$ and maximized (under the assumption that $n \geq 11k$ and hence $|C| \geq 9k$) for $|C| = 9k$ yielding $|S| \leq 3k$. ∎

Let $T_R \subseteq T$ be the set of rare colors. Define a new auxiliary bipartite graph H^* with bi-partition $(C, D \cup T_R)$ having an edge between a vertex $c \in C$ and a vertex $d \in D$ if $\{c, d\} \notin E(G)$ and an edge between a vertex $c \in C$ and a vertex $t \in T_R$ if $t \in L(c)$. Let X be a minimum vertex cover of H^*. Refer to the colors $T_R \setminus X$ as *constrained* rare colors. Note that constrained rare colors only occur on lists of vertices in $D \cup (C \cap X)$. Let $T' = T \setminus (T_R \setminus X)$, $V' = (D \setminus X) \cup (C \cap X)$, and set $q = |T'| - |C \setminus X|$. Before we continue, we want to provide some useful observations about the sizes of the considered sets and numbers.

Observation 1. *It holds that:*

- $|X| \leq |D| + |T_R| \leq 5k$,
- $|V'| \leq |D| + |X| \leq 7k$,
- $q \leq |T| - |C| + |C \cap X| \leq |D| + |X| \leq 7k$; *this holds because* $|T| \leq |V| = |C| + |D|$.

LEMMA 5.4. *Assume $n \geq 11k$. Then G has a list coloring if and only if there is a partial list coloring $\lambda_0 \colon V' \to T$ that uses at most $q = |T'| - |C \setminus X|$ colors from T'.*

Proof. The number of colors usable in $C \setminus X$ is $|T'| - p$ where p is the number counted above (since constrained rare colors cannot be used in $C \setminus X$ even if they are unused in λ_0). Thus it is a requirement that $|T'| - p \geq |C \setminus X|$. That is, $p \leq |T'| - |C \setminus X| = q$. Thus necessity is clear. We show sufficiency as well. That is, let λ_0 be a partial list coloring with scope $V' = (C \cap X) \cup (D \setminus X)$ which uses at most q colors of T'. We modify and extend λ_0 to a list coloring of G.

First let H_0 be the bipartite graph with bi-partition $(V, T_R \setminus X)$ and let M_0 be a matching saturating $T_R \setminus X$; note that this exists by reduction rule 5. We modify λ_0 to a coloring λ'_0 so that every constrained rare color is used by λ'_0, by iterating over every color $t \in T_R \setminus X$; for every t, if t is not yet used by λ'_0, then let $vt \in M_0$ and update λ'_0 with $\lambda'_0(v) = t$. Note that the scope of λ'_0 after this modification is contained in $(C \cap X) \cup D$. Next, let M be a maximum matching in H^*. We use M

[2] The constants $11k$ and $6k$ in this paragraph are chosen to make the arguments work smoothly. A smaller kernel is possible with a more careful analysis and further reduction rules.

to further extend λ'_0 in stages to a partial list coloring λ which colors all of D and uses all rare colors. In phase 1, for every color $t \in T_R \cap X$ which is not already used, let $vt \in M$ be the edge covering t and assign $\lambda(v) = t$. Note that M matches every vertex of X in H^* with a vertex not in X, thus the edge vt exists and v has not yet been assigned in λ. Hence, at every step we maintain a partial list coloring, and at the end of the phase all rare colors have been assigned. Finally, as phase 2, for every vertex $v \in D \cap X$ not yet assigned, let $uv \in M$ where $u \in C$; necessarily $u \in C \setminus X$ and u is as of yet unassigned in λ. The number of colors assigned in λ thus far is at most $|X| + |D| \leq |T_R| + 2|D| \leq 7k$, whereas $|L(u) \cap L(v)| \geq n - 2k \geq 9k$, hence there always exists an unused shared color that can be mapped to $\lambda(u) = \lambda(v)$. Let λ be the resulting partial list coloring. We claim that λ can be extended to a list coloring of G.

Let A be the scope of λ and let $H' = H - (A \cap \lambda(A))$. Note that $A \cap C \subseteq V(M)$, hence $|A \cap C| \leq |D| + |T_R| \leq 5k$. Thus by Lemma 5.2, no non-trivial Hall set in H' can contain a rare color. However, all rare colors are already used in λ. Thus H' contains no non-trivial Hall set of colors. Thus the only possibility that λ is not extensible is that H' has a trivial Hall set, i.e., $|T \setminus \lambda(A)| < |C \setminus A|$. However, every modification after λ'_0 added one vertex to A and one color to $\lambda(A)$, keeping the balance between the two sides. Thus already the partial coloring λ'_0 leaves behind a trivial Hall set. However, λ'_0 colors precisely $C \cap X$ in C and leaves at least $|T'| - q$ colors remaining. By design this is at least $|C \setminus X|$, yielding a contradiction. Thus we find that H' contains no Hall set, and λ is a list coloring of G. ∎

Before we give our compression and kernelization results, we need the following auxiliary lemma.

LEMMA 5.5. *T' contains at least $|T'| - |V'|k$ colors that are universal to all vertices in V'.*

Proof. The list of every vertex $v \in V'$ misses at most k colors from T'. Hence all but at most $|V'|k$ colors in T' are universal to all vertices in V'. ∎

For clarity, let us define the output problem of our compression explicitly.

BUDGET-CONSTRAINED LIST COLORING

Input:	A graph G, a set T of colors, a list $L(v) \subseteq T$ for every $v \in V(G)$, and a pair (T', q) where $T' \subseteq T$ and $q \in \mathbb{N}$.
Problem:	Is there a proper list coloring for G that uses at most q distinct colors from T'?

THEOREM 5.6. *$(n-k)$-*REGULAR LIST COLORING *admits a compression into an instance of* BUDGET-CONSTRAINED LIST COLORING *with at most $11k$ vertices and $\mathcal{O}(k^2)$ colors, encodable in $\mathcal{O}(k^2 \log k)$ bits.*

Proof. If $|V(G)| \leq 11k$, then G itself can be used as the output (with a dummy budget constraint). Otherwise, all the bounds above apply and Lemma 5.4 shows that the existence of a list coloring in G is equivalent to the existence of a list coloring in $G[V']$ that uses at most q colors from T'. Since $|V'| \leq 7k$, it only remains to reduce the number of colors in $T_R \cup T'$. Clearly, if $|T'| < |V'|k + q$, then $|T_R \cup T'| \leq 3k + (7k)k \in \mathcal{O}(k^2)$ and there is nothing left to show. So suppose that $|T'| \geq |V'|k + q$. Then, it follows from Lemma 5.5 that T' contains at least q colors that are universal to the vertices in V' and we obtain an equivalent instance by removing all but exactly

q universal colors from T', which leaves us with an instance with at most $|T_R| + q \leq 3k + 7k^2 \in \mathcal{O}(k^2)$ colors, as required. Finally, to describe the output concisely, note that $G[V']$ can be trivially described in $\mathcal{O}(k^2)$ bits, and the lists $L(v)$ can be described by enumerating $T \setminus L(v)$ for every vertex v, which is k colors per vertex, each color identifiable by $\mathcal{O}(\log k)$ bits. ◻

Note that the compression is asymptotically essentially optimal, since even the basic 4-Coloring problem does not allow a compression in $\mathcal{O}(n^{2-\varepsilon})$ bits for any $\varepsilon > 0$ unless the polynomial hierarchy collapses [24]. For completeness, we also give a proper kernel, which can be obtained in a similar manner to the compression given in Theorem 5.6.

Theorem 5.7. $(n-k)$-Regular List Coloring *admits a kernel with* $\mathcal{O}(k^2)$ *vertices and colors.*

Proof. We distinguish two cases depending on whether or not $|T'| < |V'|k + q$. If $|T'| < |V'|k + q$, then $|T| \leq |T_R| + |T'| < 3k + |V'|k + q \leq 3k + (7k)(k+1) \in \mathcal{O}(k^2)$. Since a list coloring requires at least one distinct color for every vertex in C, it holds that $|C| \leq |T| \leq 3k + (7k)(k+1)$ and hence $|V(G)| \leq (3+7k)k + 2k \in \mathcal{O}(k^2)$, implying the desired kernel.

If on the other hand, $|T'| \geq |V'|k + q$, then, because of Lemma 5.5 it holds that T' contains a set U of exactly q colors that are universal to the vertices in V'. Recall that Lemma 5.4 shows that the existence of a list coloring in G is equivalent to the existence of a list coloring in $G[V']$ that uses at most $q = |T'| - |C \setminus X|$ colors from T'. It follows that the graph $G[V']$ has a list coloring using only colors in $(T_R \setminus X) \cup U$ if and only if G has a list coloring. Hence, it only remains to restore the regularity of the instance. We achieve this as follows. First we add a set T_N of $|(T_R \setminus X) \cup U|$ novel colors. We then add these colors (arbitrarily) to the color lists of the vertices in V' such that the size of every list (for any vertex in V') is $|(T_R \setminus X) \cup U|$. This clearly already makes the instance regular, however, now we also need to ensure that no vertex in V' can be colored with any of the new colors in T_N. To achieve this we add a set C_N of $|T_N|$ novel vertices to $G[V']$, which we connect to every vertex in $(C \cap X) \cup C_N$ and whose lists all contain all the new colors in T_N. It is clear that the constructed instance is equivalent to the original instance since all the new colors in T_N are required to color the new vertices in C_N and hence no new color can be used to color a vertex in V'. Moreover, D is still a clique modulator and the number k' of missing colors (in each list of the constructed instance) is equal to $|D| + |C \cap X| \leq 2k + 5k$ because the instance is $(n - |D| - |C \cap X|)$-regular. Finally, the instance has at most $|V' \cup C_N| \leq 7k + 3k + 7k = 17k \in \mathcal{O}(k)$ vertices and at most $2(|T_R| + |U|) \leq 2(3k + 7k) = 20k \in \mathcal{O}(k)$ colors, as required. ◻

6. Saving k colors: Pre-coloring and List Coloring Variants. In this section, we consider natural pre-coloring and list coloring variants of the "saving k colors" problem, defined as:

$(n-k)$-Coloring parameterized by k

Input: A graph G with n vertices and an integer k.
Problem: Does G have a proper coloring using at most $n-k$ colors?

This problem is known to be FPT (it even allows for a linear kernel) [12], when parameterized by k. Notably the problem provided the main motivation for the introduction of $(n-k)$-Regular List Coloring in [3, 2].

We consider the following (pre-coloring and list coloring) extensions of $(n-k)$-Coloring.

$(n-|Q|)$-Pre-Coloring Extension parameterized by $n-|Q|$

Input: A graph G with n vertices and a pre-coloring $\lambda_P : X \to Q$ for $X \subseteq V(G)$ where Q is a set of colors.

Problem: Can λ_P be extended to a proper coloring of G using only colors from Q?

List Coloring with $n-k$ colors parameterized by k

Input: A graph G on n vertices with a list $L(v)$ of colors for every $v \in V(G)$ and an integer k.

Problem: Is there a proper list coloring of G using at most $n-k$ colors?

Note that the following variant seems natural, however, is trivially NP-complete even when the parameter k is equal to 0, since the problem with an empty pre-coloring then corresponds to the problem whether G can be colored by at most $|Q|$ colors.

$(|Q|-k)$-Pre-Coloring Extension parameterized by k

Input: A graph G with n vertices, a pre-coloring $\lambda_P : X \to Q$ for $X \subseteq V(G)$ where Q is a set of colors, and an integer k.

Problem: Can λ_P be extended to a proper coloring of G using at most $|Q|-k$ colors from Q?

Interestingly, we show that $(n-|Q|)$-Pre-Coloring Extension is FPT and even allows a linear kernel. Thus, we generalize the above-mentioned result of Chor et al. [12] (set $Q = [n-k]$ and $X = \emptyset$). However, List Coloring with $n-k$ colors is easily seen to be NP-hard (even for $k = 0$) using a trivial reduction from 3-Coloring.

Theorem 6.1. $(n-|Q|)$-Pre-Coloring Extension *(parameterized by $n-|Q|$) has a kernel with at most $6(n-|Q|)$ vertices and is hence fixed-parameter tractable.*

Proof. Let G' be the graph obtained from G after applying the following reduction rules:

Reduction Rule 6. *If u and v are two distinct vertices in $G \setminus X$ such that $\lambda_P(N_G(u)) \cup \lambda_P(N_G(v)) = Q$, then we add an edge between u and v in G.*

This rule is safe because u and v cannot be colored with the same color.

Reduction Rule 7. *If u is a vertex in $G \setminus X$ that is adjacent to a vertex $v \in X$, then we can safely add all edges between u and every vertex in $\lambda_P^{-1}(\lambda_P(v))$.*

This rule is safe because u cannot be colored by $\lambda_P(v)$.

Reduction Rule 8. *If u and v are two distinct vertices in X such that $\lambda_P(u) \neq \lambda_P(v)$, then we can again safely add an edge between u and v.*

This rule is safe because u and v cannot be colored with the same color.

Let M be a maximal matching in the complement of G'. Note that if $|M| \leq n-|Q|$, then $V(M)$ is a clique modulator for G' of size at most $2(n-|Q|)$ and we obtain a kernel with at most $6(n-|Q|)$ vertices using Theorem 4.6. Thus assume that

$|M| \geq n - |Q|$. In this case we can safe $|M| \geq n - |Q|$ colors by giving the endpoints of every edge in M the same color. Namely, let $\{u, v\} \in M$, then:

- if $u, v \notin X$, then it follows from Reduction Rule 6 that there is a color $q \in Q$ that can be given to both vertices,
- if $u \notin X$ and $v \in X$, then it follows from Reduction Rule 7 that we can color u with color $\lambda_P(v)$,
- if $u, v \in X$, then by Reduction Rule 8 we have that $\lambda_P(u) = \lambda_P(v)$.

Note that after coloring the edges in M with the same color, removing $V(M)$ from G', and removing the colors used for the edges in M from Q, the number of colors in the remaining instance is equal to the number of vertices in the remaining instance, implying that the remaining instance can be properly colored. ∎

7. Conclusions. We have shown several results regarding the parameterized complexity of List Coloring and Pre-Coloring Extension problems. We showed that List Coloring, and hence also Pre-Coloring Extension, parameterized by the size of a clique modulator admits a randomized FPT algorithm with a running time of $\mathcal{O}^*(2^k)$, matching the best known running time of the basic Chromatic Number problem parameterized by the number of vertices. This answers open questions of Golovach et al. [23]. Note that also that List Coloring is already W[1]-hard parameterized by vertex cover [23], i.e., modulator to an independent set, which excludes even quite simple generalizations of our result to, e.g., a modulator to a disjoint union of cliques. Additionally, we showed that Pre-Coloring Extension under the same parameter admits a linear vertex kernel with at most $3k$ vertices and that $(n-k)$-Regular List Coloring admits a compression into a problem we call Budget-Constrained List Coloring, into an instance with at most $11k$ vertices, encodable in $\mathcal{O}(k^2 \log k)$ bits. The latter also admits a proper kernel with $\mathcal{O}(k^2)$ vertices and colors. This answers an open problem of Banik et al. [3].

One obvious open question is whether it is possible to derandomize our algorithm for List Coloring. This seems, however, very challenging as it would require a derandomization of Lemma 2.4, which has been an open problem for some time. It might, however, be possible (and potentially more promising) to consider a different approach than ours. Another open question is to optimize the bound $11k$ on the number of vertices in the $(n-k)$-Regular List Coloring compression, and/or show a proper kernel with $\mathcal{O}(k)$ vertices. Finally, another set of questions is raised by Escoffier [19], who studied the Max Coloring problem from a "saving colors" perspective. In addition to the questions explicitly raised by Escoffier, it is natural to ask whether his problems Saving Weight and Saving Color Weights admit FPT algorithms with a running time of $2^{\mathcal{O}(k)}$ and/or polynomial kernels.

REFERENCES

[1] Noga Alon, Gregory Z. Gutin, Eun Jung Kim, Stefan Szeider, and Anders Yeo. Solving MAX-r-SAT above a tight lower bound. *Algorithmica*, 61(3):638–655, 2011.

[2] Pranav Arora, Aritra Banik, Vijay Kumar Paliwal, and Venkatesh Raman. Some (in)tractable parameterizations of coloring and list-coloring. In Jianer Chen and Pinyan Lu, editors, *Frontiers in Algorithmics - 12th International Workshop, FAW 2018, Proceedings*, volume 10823 of *Lecture Notes in Computer Science*, pages 126–139. Springer, 2018.

[3] Aritra Banik, Ashwin Jacob, Vijay Kumar Paliwal, and Venkatesh Raman. Fixed-parameter tractability of $(n-k)$ List Coloring. In C.J. Colbourn, R. Grossi, and N. Pisani, editors, *IWOCA 2019, Proceedings*, volume 11638 of *Lecture Notes in Computer Science*, pages 61–69, 2019. URL: https://doi.org/10.1007/978-3-030-25005-8_6, doi:10.1007/978-3-030-25005-8_6.

[4] D. Berend and T. Tassa. Improved bounds on bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 30:185–205, 2010.

[5] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Narrow sieves for parameterized paths and packings. *J. Comput. Syst. Sci.*, 87:119–139, 2017.

[6] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusion-exclusion. *SIAM J. Comput.*, 39(2):546–563, 2009.

[7] Andreas Björklund, Petteri Kaski, and Lukasz Kowalik. Constrained multilinear detection and generalized graph motifs. *Algorithmica*, 74(2):947–967, 2016.

[8] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernel bounds for path and cycle problems. *Theor. Comput. Sci.*, 511:117–136, 2013.

[9] James R. Bunch and John E. Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Mathematics of Computation*, 28(125):231–236, 1974.

[10] Leizhen Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003.

[11] Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theor. Comput. Sci.*, 411(40-42):3736–3756, 2010.

[12] Benny Chor, Mike Fellows, and David Juedes. Linear kernels in linear time, or how to save k colors in $O(n^2)$ steps. In *Proceedings of the 30th International Conference on Graph-Theoretic Concepts in Computer Science*, WG'04, pages 257–269. Springer-Verlag, 2004.

[13] Maria Chudnovsky. Coloring graphs with forbidden induced subgraphs. In *2014 International Congress of Mathematicians*, volume IV, pages 291–302, 2014.

[14] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

[15] 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 Trans. Algorithms*, 12(3):41:1–41:24, 2016.

[16] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[17] Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Kernelization Lower Bounds Through Colors and IDs. *ACM Trans. Algorithms*, 11(2):13:1–13:20, 2014.

[18] R.G. Downey and M.R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

[19] Bruno Escoffier. Saving colors and max coloring: Some fixed-parameter tractability results. *Theor. Comput. Sci.*, 758:30–41, 2019. URL: https://doi.org/10.1016/j.tcs.2018.08.002, doi:10.1016/j.tcs.2018.08.002.

[20] F.V. Fomin, D. Lokshtanov, S. Saurabh, and M. Zehavi. *Kernelization*. Cambridge Univ. Press, 2019.

[21] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[22] Petr A. Golovach, Matthew Johnson, Daniël Paulusma, and Jian Song. A survey on the computational complexity of coloring graphs with forbidden subgraphs. *Journal of Graph Theory*, 84(4):331–363, 2017.

[23] Petr A. Golovach, Daniël Paulusma, and Jian Song. Closing complexity gaps for coloring problems on h-free graphs. *Inf. Comput.*, 237:204–214, 2014.

[24] Bart M. P. Jansen and Astrid Pieterse. Sparsification upper and lower bounds for graph problems and not-all-equal SAT. *Algorithmica*, 79(1):3–28, 2017.

[25] Tommy R. Jensen and Bjarne Toft. *Graph Coloring Problems*. Wiley & Sons, 1994.

[26] Ioannis Koutis. *Multilinear Monomial Detection*, pages 1375–1378. Springer New York, New York, NY, 2016. URL: https://doi.org/10.1007/978-1-4939-2864-4_784, doi:10.1007/978-1-4939-2864-4_784.

[27] Robert Krauthgamer and Ohad Trabelsi. The set cover conjecture and subgraph isomorphism with a tree pattern. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPIcs*, pages 45:1–45:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019. URL: https://doi.org/10.4230/LIPIcs.STACS.2019.45, doi:10.4230/LIPIcs.STACS.2019.45.

[28] L. Lovász. Coverings and coloring of hypergraphs. In *4th Southeastern Conference on Combinatorics, Graph Theory, and Computing, Utilitas Math*, pages 3–12, 1973.

[29] Silvio Micali and Vijay V. Vazirani. An o(sqrt(|v|) |e|) algorithm for finding maximum matching in general graphs. In *21st Annual Symposium on Foundations of Computer Science, Syracuse, New York, USA, 13-15 October 1980*, pages 17–27. IEEE Computer Society, 1980.

[30] Daniël Paulusma. Open problems on graph coloring for special graph classes. In Ernst W. Mayr,

editor, *Graph-Theoretic Concepts in Computer Science - 41st International Workshop, WG 2015, Garching, Germany, June 17-19, 2015, Revised Papers*, volume 9224 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 2015.

[31] B. Randerath and I. Schiermeyer. Vertex coloring and forbidden subgraphs - a survey. *Graphs Comb.*, 20:1–40, 2004.

[32] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.

[33] Z. Tuza. Graph colorings with local restrictions - a survey. *Discussiones Mathematicae Graph Theory*, 17:161–228, 1997.

[34] Magnus Wahlström. Abusing the Tutte matrix: An algebraic instance compression for the K-set-cycle problem. In Natacha Portier and Thomas Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, February 27 - March 2, 2013, Kiel, Germany*, volume 20 of *LIPIcs*, pages 341–352. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[35] F. Yates. The design and analysis of factorial experiments. Technical Report Technical Communication no. 35, Commonwealth Bureau of Soils, 1937.

[36] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*, EUROSAM '79, pages 216–226, 1979.