

Graph Coloring Below Guarantees via Co-Triangle Packing

Shyan Akmal   

INSAIT, Sofia University “St. Kliment Ohridski”, Bulgaria

Tomohiro Koana  

Keio University, Yagami, Japan

Abstract

In the ℓ -COLORING problem, we are given a graph on n nodes, and tasked with determining if its vertices can be properly colored using ℓ colors. In this paper we study *below-guarantee graph coloring*, which tests whether an n -vertex graph can be properly colored using $g - k$ colors, where g is a trivial upper bound such as n . We introduce an algorithmic framework that builds on a packing of co-triangles $\overline{K_3}$ (independent sets of three vertices): the algorithm greedily finds co-triangles and employs a win-win analysis. If many are found, we immediately return *yes*; otherwise these co-triangles form a small *co-triangle modulator*, whose deletion makes the graph co-triangle-free.

Extending the work of [Gutin et al., SIDMA 2021], who solved ℓ -COLORING (for any ℓ) in randomized $O^*(2^k)$ time when given a $\overline{K_2}$ -free modulator of size k , we show that this problem can likewise be solved in randomized $O^*(2^k)$ time when given a $\overline{K_3}$ -free modulator of size k .

This result in turn yields a randomized $O^*(2^{3k/2})$ algorithm for $(n - k)$ -COLORING (also known as DUAL COLORING), improving the previous $O^*(4^k)$ bound. We then introduce a smaller parameterization, $(\omega + \bar{\mu} - k)$ -COLORING, where ω is the clique number and $\bar{\mu}$ is the size of a maximum matching in the complement graph; since $\omega + \bar{\mu} \leq n$ for any graph, this problem is strictly harder. Using the same co-triangle-packing argument, we obtain a randomized $O^*(2^{6k})$ algorithm, establishing its fixed-parameter tractability for a smaller parameter. Complementing this finding, we show that no fixed-parameter tractable algorithm exists for $(\omega - k)$ -COLORING or $(\bar{\mu} - k)$ -COLORING under standard complexity assumptions.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Graph algorithms analysis; Theory of computation $\rightarrow$ Parameterized complexity and exact algorithms

Keywords and phrases coloring, parameterized algorithms, algebraic algorithms, above-guarantee, below-guarantee, subset convolution, determinants

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2025.5

Related Version *Full Version*: <https://arxiv.org/abs/2509.12347>

Funding *Shyan Akmal*: Partially funded by the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure).

Tomohiro Koana: Supported by JST ERATO Grant Number JPMJER2301, Japan.

Acknowledgements We thank the anonymous reviewers for helpful feedback on this work.

1 Introduction

Graph coloring is a cornerstone of theoretical computer science and discrete mathematics: it connects to deep structure theorems and has driven the development of numerous algorithmic techniques [3, 5, 6, 8, 14]. Given an undirected graph G on n vertices, a *proper coloring* of G is an assignment of colors to its nodes such that any two adjacent vertices receive different colors. The canonical decision problem for this concept is k -COLORING: given an integer k and a graph G , decide whether G admits a proper coloring with at most k colors.



© Shyan Akmal and Tomohiro Koana;

licensed under Creative Commons License CC-BY 4.0

36th International Symposium on Algorithms and Computation (ISAAC 2025).

Editors: Ho-Lin Chen, Wing-Kai Hon, and Meng-Tsung Tsai; Article No. 5; pp. 5:1–5:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

While 2-COLORING is polynomial-time solvable, 3-COLORING (and therefore k -COLORING for every $k \geq 3$) is already NP-hard [18]. The field of parameterized complexity tackles such NP-hard tasks by measuring the complexity of solving these problems in terms of secondary measures. In this paper we focus on a particularly important subarea of this field known as *below-guarantee parameterization*.

Below-Guarantee Parameterization. In the guarantee parameterization paradigm, we reformulate problems by first identifying classes of instances that are trivially solvable – typically because a solution is guaranteed to exist – and then parameterize general instances by their distance from these “easy” cases. See [13] for a survey of this area.

Dual Coloring. In the context of coloring graphs with n nodes, observe that the n -COLORING problem is trivial, because we can obtain a proper coloring by assigning each vertex a distinct color. The framework of below-guarantee parameterization described above then motivates studying the $(n - k)$ -COLORING problem, also referred to as DUAL COLORING, where the parameter k captures how many colors we are trying to “save” compared to the trivial coloring using different colors for all vertices. The DUAL COLORING problem has been influential in parameterized complexity, with its study in [7] introducing the concept of crown reductions, by now a basic technique in the field of kernelization [10, Chapter 4]. Given an n -node instance G of DUAL COLORING and any constant $\varepsilon > 0$, [17] showed that in polynomial time one can kernelize the instance to a graph G' on $(2 + \varepsilon)k$ vertices such that the answer to the DUAL COLORING problem is the same on G and G' , for sufficiently large k in terms of $\varepsilon > 0$. Combining this kernelization with existing $2^n \text{poly}(n)$ time algorithms for graph coloring [5], we immediately recover a $O^*(4^{(1+\varepsilon)k})$ time algorithm for DUAL COLORING for any constant $\varepsilon > 0$, where we write $O^*(f(k))$ to denote $f(k) \text{poly}(n)$. Without going through this kernelization, one can also use the algorithms of [12] to solve DUAL COLORING directly in $O^*(4^k)$ time. We improve upon this result:

► **Theorem 1.** DUAL COLORING can be solved in randomized $O^*(2^{3k/2})$ time.

Let us mention a closely related problem called $(n - k)$ -SET COVER. In this problem, we are given a family $\mathcal{F}$ of subsets of $[n]$, and are tasked with determining if there exist $(n - k)$ sets from $\mathcal{F}$ whose union is $[n]$. Very recently, Alferov et al. [2] showed that $(n - k)$ -SET COVER can be solved in $O^*(2^{3k/2}|\mathcal{F}|)$ time. Even though the $(n - k)$ -COLORING problem can be cast as an instance of $(n - k)$ -SET COVER, where $\mathcal{F}$ is the collection of all independent vertex sets of the graph, this result of [2] does not imply an $O^*(2^{3k/2})$ time algorithm for $(n - k)$ -COLORING, because $\mathcal{F}$ can have exponentially many sets in n . The dynamic programming approach used in the algorithm of [2] appears to inherently require this runtime dependence on $|\mathcal{F}|$ even when the sets $\mathcal{F}$ can be implicitly described as in $(n - k)$ -COLORING, so that Theorem 1 is not directly implied by this previous work. However, we note it seems plausible that one could combine the ideas of [2] with the subset convolution arguments of [4] to obtain fast algorithms for DUAL COLORING as well, and recover Theorem 1.

A Stronger Structural Guarantee. Given a graph G , we let $\overline{G}$ denote its complement graph, which has the same vertex set as G , but an edge $\{u, v\}$ if and only if u and v are distinct, non-adjacent nodes in G . Let $\omega = \omega(G)$ denote the size of the maximum clique in G . Let $\overline{\mu} = \mu(\overline{G})$ denote the size of the maximum matching in the complement $\overline{G}$. Then we have the following structural parameter-based guarantee for coloring G .

► **Observation 2.** The graph G admits a proper coloring using at most $(\omega + \overline{\mu})$ colors.

Proof. Let M be a maximal matching in $\overline{G}$. By definition, $|M| \leq \overline{\mu}$. By maximality of M in $\overline{G}$, the vertices that are not incident to any edge in M form an independent set in $\overline{G}$. Thus the set C of vertices outside M form a clique in G , so $|C| \leq \omega$. We color G as follows. For each edge in M , we assign a unique color to both of its endpoints, and we assign distinct new colors to all vertices in C . In total, we use at most $(\omega + \overline{\mu})$ colors. By construction this coloring is proper, which proves the claim. $\blacktriangleleft$

Note that in any graphs with n vertices, we always have $(\omega + \overline{\mu}) \leq n$. This is because if we take a clique C in G of size ω and a matching M in $\overline{G}$ of size $\overline{\mu}$, then by definition each edge of M is incident to at least one distinct vertex outside of C , which forces $n \geq |C| + |M| = (\omega + \overline{\mu})$. Thus the guarantee of Observation 2 is stronger than the trivial fact that any graph with n vertices always has a proper coloring using n colors.

In the same way this latter observation motivated the DUAL COLORING problem, it is natural to ask if we can obtain efficient algorithms while parameterizing below the stronger guarantee provided by Observation 2. We prove that this is indeed possible.

► **Theorem 3** (Parameterization from Stronger Guarantee). *Given an integer $k \geq 1$, and a graph G that has maximum clique size ω and whose complement has maximum matching size $\overline{\mu}$, we can solve $(\omega + \overline{\mu} - k)$ -COLORING in randomized $O^*(2^{6k})$ time.*

Our approach. The key ideas behind Theorems 1 and 3 are sketched below; see Section 1.1 for more details. We first greedily compute a maximal packing of co-triangles $\overline{K}_3$ (independent triplets). If the packing holds many co-triangles, saving two colors per co-triangle already yields the desired proper coloring. Otherwise, the packing's vertices form a small *co-triangle modulator* S , whose deletion results in a $\overline{K}_3$ -free graph. In the latter case, we extend the randomized procedure of Gutin et al. [12], to show that we can solve the desired coloring problem in $O^*(2^{|S|})$ time.

Hardness results. We also note that algorithms which are fixed-parameter tractable with respect to k are unlikely to exist for $(\omega - k)$ -COLORING and $(\overline{\mu} - k)$ -COLORING, so that combining ω and $\overline{\mu}$ together in Theorem 3 appears to be necessary in order to achieve efficient parameterized algorithms with respect to these parameters. For the former, this is because 3-COLORING is NP-hard in planar graphs [11]. Since a planar graph does not have a clique of size five, this means that assuming $P \neq NP$ we cannot hope to solve $(\omega - k)$ -COLORING in $f(k) \text{ poly}(n)$ time for any function $f(\cdot)$. To show hardness for the latter problem, we prove the following.

► **Theorem 4.** *The $(n/2 - k)$ -COLORING problem is W[1]-hard in the parameter k on graphs with n vertices whose complement has a perfect matching.*

In view of Theorem 4, we cannot hope for an $O^*(f(k))$ -time algorithm for $(\overline{\mu} - k)$ -COLORING, for any function $f(\cdot)$.

1.1 Technical Overview

As a starting point, we give a brief overview for how the work of [12] implies a $O^*(4^k)$ time algorithm for DUAL COLORING.

Given a host graph G and pattern graph H , we say G is *H-free* if G does not contain H as an induced subgraph. We say a set of vertices S in G is an *H-free modulator* if deleting the vertices in S from G makes the graph H -free. For $\ell \in \mathbb{N}$, we let K_ℓ denote the complete graph on ℓ vertices, and $\overline{K}_\ell$ denote its complement, the independent set on ℓ vertices.

The main relevant result from previous work is the following, proven in [12, Theorem 3.5].

► **Proposition 5** (Clique Modulator). *Given a graph G together with a $\overline{K_2}$ -free modulator of size p (i.e., a set of at most p vertices, whose deletion from G turns the graph into a clique), we can solve k -COLORING on G for any k in randomized $O^*(2^p)$ time.*

Recall that given a pattern graph H and a host graph G , we say an H -packing in G is a collection $\mathcal{H}$ of vertex-disjoint, induced copies of H in G . Using Proposition 5 we can solve DUAL COLORING in $O^*(4^k)$ time using the following three steps.

Step 1: Identify Matching

Extract a maximal $\overline{K_2}$ -packing M in G .

Step 2: Large Matching $\Rightarrow$ Easy Instance

If $|M| \geq k$, then report that a proper coloring using at most $(n - k)$ colors exists.

Step 3: Small Matching $\Rightarrow$ Small Modulator

If $|M| < k$, the vertices participating in M form a $\overline{K_2}$ -free modulator of size $p < 2k$. Then apply Proposition 5 to solve DUAL COLORING in $O^*(4^k)$ time.

In **Step 1** we obtain M by running a polynomial time algorithm for finding a maximum matching in the complement graph $\overline{G}$. In **Step 2** we observe that if $|M| \geq k$, then we can assign a distinct color for each $\overline{K_2}$ in M to color its two vertices, and then color all remaining vertices in G with different colors to obtain a proper coloring using at most $k + (n - 2k) = n - k$ colors overall. In **Step 3** we observe that if $|M| < k$, by maximality deleting each $\overline{K_2}$ in M results in a $\overline{K_2}$ -free graph, so Proposition 5 solves the problem.

This strategy of extracting a matching and then performing casework on its size to either identify a solution or enforce more structure is similar to previous work on kernelization for DUAL COLORING [7, 17] and applications of crown decomposition [10, Chapter 4].

To solve DUAL COLORING faster in $O^*(2^{3k/2})$ time, we work with a structure more complicated than a matching: a triangle packing. The core technical lemma powering our new algorithms is the following result.

► **Lemma 6.** *Given a $\overline{K_3}$ -free modulator of size p and any positive integer k , the k -COLORING problem can be solved in randomized $O^*(2^p)$ time.*

Since any $\overline{K_3}$ -free modulator is also a $\overline{K_2}$ -free modulator by definition, Lemma 6 is a stronger version of Proposition 5. We remark that an extension to $\overline{K_4}$ -free modulator is unlikely because COLORING is NP-hard on $\overline{K_4}$ -free graphs [16].

Step 1: Identify Triangle Packing

Extract a maximal $\overline{K_3}$ -packing $\mathcal{T}$ in G .

Step 2: Large Packing $\Rightarrow$ Easy Instance

If $|\mathcal{T}| \geq k/2$, then report that a proper coloring using at most $(n - k)$ colors exists.

Step 3: Small Packing $\Rightarrow$ Small Modulator

If $|\mathcal{T}| < k/2$, the vertices participating in $\mathcal{T}$ form a $\overline{K_3}$ -free modulator of size $p < 3k/2$. Then apply Lemma 6 to solve DUAL COLORING in $O^*(2^{3k/2})$ time.

In **Step 1** we obtain $\mathcal{T}$ in polynomial time by running a greedy algorithm. In **Step 2** we observe that if $|\mathcal{T}| \geq k/2$, then we can assign each $\overline{K_3}$ in $\mathcal{T}$ a different color to color its three vertices, and then color all remaining vertices in G with different colors to obtain a proper coloring using at most $k/2 + (n - 3k/2) = n - k$ colors overall. In **Step 3** we observe that if $|\mathcal{T}| < k/2$, by maximality deleting each $\overline{K_3}$ in $\mathcal{T}$ results in a $\overline{K_3}$ -free graph, so Lemma 6 solves the problem.

The proof of Proposition 5 from [12] is established using algebraic techniques. The main idea is that given a $\overline{K_2}$ -free modulator S of size p , the induced subgraph on $V \setminus S$ is a clique, so every vertex outside of S must be assigned a different color. Solving the k -COLORING problem can then be reduced to a problem similar to bipartite matching, where one must determine for each vertex outside of S the subset of vertices in S it shares a color with (intuitively, the subset of S it is “matched to”). The authors use determinants to enumerate bipartite matchings in an auxiliary graph capturing this question, and apply algorithms for *polynomial sieving* and *fast subset convolution* to find the best matching (corresponding to an optimal coloring) and solve this problem in $O^*(2^p)$ time, comparable to the number of subsets of S that must be considered.

Our proof of Lemma 6 uses similar technical ingredients. Here, because we are given a $\overline{K_3}$ -free modulator S instead of a $\overline{K_2}$ -free modulator, we can no longer assume that the vertices in $\overline{S} = V \setminus S$ all get different colors. However, we can assume that in a proper coloring of the graph, any given color appears at most twice across the nodes in $\overline{S}$. This then lets us reduce the k -COLORING problem in this context to a problem related to perfect matchings, albeit in a nonbipartite graph. At a high level, we do this by considering all partitions of $\overline{S}$ into sets of size at most two, where these parts represent distinct color classes in $\overline{S}$, and then consider all ways of matching these parts to different subsets of S , corresponding to extending these color classes from $\overline{S}$ to the full graph.

Since we work with matchings in general undirected graphs, we use Pfaffians instead of determinants in our arguments. We also used a Pfaffian-based algebraic approach in recent work on EDGE COLORING [1]. On top of this framework, we use fast subset convolution to ensure that we achieve the same $O^*(2^p)$ runtime as before.

Our algorithm for $(\omega + \overline{\mu} - k)$ -COLORING follows by employing a similar strategy to the procedure for DUAL COLORING outlined above, with a more refined analysis with respect to the relevant structural parameters.

Comparison with the $(n - k)$ -SET COVER Result. Alferov et al. [2] achieved an $O^*(2^{3k/2}|\mathcal{F}|)$ algorithm for $(n - k)$ -SET COVER on input families $\mathcal{F}$ via a different toolkit. Their procedure likewise begins by extracting a maximal triangle packing and immediately reports a yes-instance when that packing is large (**Step 2** in our outline). When the packing is small, they study the matching structure in the remaining part via the Gallai-Edmonds decomposition. A sufficiently large matching again certifies a solution, and they carefully use this fact to obtain a runtime bound of $O^*(2^{3k/2}|\mathcal{F}|)$. These arguments are tailored to $(n - k)$ -SET COVER, and do not lead to an efficient algorithm parameterized by $\overline{K_3}$ -free modulator size as in Lemma 6, which is central to proving Theorem 3. For the special case of $(n - k)$ -COLORING, where the set $\mathcal{F}$ can be succinctly described as the collection of independent sets in a given graph, the dynamic program employed in [2] still suffers from this issue. Due to this limitation, this previous approach does not directly prove Theorem 1 either.

Organization. In Section 2, we review notation and basic facts about graphs, matrices, and polynomials that will be useful to us. In Section 3 we prove Lemma 6. In Section 4, we apply Lemma 6 to design our algorithm for DUAL COLORING and $(\omega + \overline{\mu} - k)$ -COLORING (our harder structural parameter variant of DUAL COLORING) and prove Theorem 3, and also establish a lower bound for a related problem by proving Theorem 4. We conclude in Section 5 by summarizing our work and mentioning relevant open problems.

2 Preliminaries

General Notation. Given a positive integer a , we let $[a] = \{1, \dots, a\}$ denote the set of the first a consecutive positive integers.

Graph Notation. We let G denote the input graph on n vertices and m edges. We let V denote the vertex set of G . Given a subset of nodes $S \subseteq V$, we let $G[S]$ denote the induced subgraph of G on S . We let $\mu(G)$, $\omega(G)$, and $\alpha(G)$ denote the size of a maximum matching, clique, and independent set in G respectively. We let K_3 denote the complete graph on three nodes, which we refer to as a *triangle*.

From Coloring to Clique Covers. Given a graph G , we say a *clique cover* of G is a collection $\mathcal{C}$ of vertex-disjoint cliques in G such that every vertex appears in some clique of $\mathcal{C}$. The *size* of this cover is simply $|\mathcal{C}|$, the number of cliques used in the clique cover.

In the p -CLIQUE COVER problem, we are given a graph G on n vertices and are tasked with determining if G has a clique cover of size at most p . For convenience, we refer to $(n - k)$ -CLIQUE COVER as the DUAL CLIQUE COVER problem. By complementing the graph it is easy to see that p -COLORING and p -CLIQUE COVER are equivalent computational tasks.

► **Observation 7 (CLIQUE COVER).** *For any integer $p \geq 1$, solving p -COLORING on a graph G is equivalent to solving p -CLIQUE COVER on the complement graph $\overline{G}$.*

Proof. Suppose we have a proper coloring of G . Then the set of vertices assigned any fixed color form an independent set in G , which means they form a clique in $\overline{G}$. Thus the coloring induces a clique cover of the same size. The reverse direction, that a solution to p -CLIQUE COVER on $\overline{G}$ implies a solution to p -COLORING on G , follows by symmetric reasoning. ◀

For convenience, in our proofs we will use Observation 7 to frame our algorithms as solving problems related to CLIQUE COVER instead of COLORING.

Algebraic Preliminaries. Throughout we work over a finite field $\mathbb{F}$ of size $\text{poly}(n)$. Arithmetic operations over $\mathbb{F}$ take $\text{poly}(\log n)$ time.

Given a set S , let $\Pi(S)$ denote the set of perfect matchings on S (i.e., the collection of partitions of S into sets of size exactly two). Given a skew-symmetric matrix $\mathbf{A}$ (i.e., a matrix that equals $\mathbf{A} = -\mathbf{A}^\top$ the negative of its transpose) with rows and columns indexed by a set S , its Pfaffian is defined to be

$$\text{Pf } \mathbf{A} = \sum_{M \in \Pi(S)} \text{sgn}(M) \prod_{\{u,v\} \in M} \mathbf{A}[u,v] \quad (1)$$

for a function $\text{sgn}: \Pi(S) \rightarrow \{-1, 1\}$ whose definition is not relevant here [20, Section 7.3.2].

An arithmetic circuit is a way of constructing a polynomial by starting with its input variables, and iteratively building up more complicated expressions by using the standard operations of addition, multiplication, and division. The size of an arithmetic circuit is the total number of non-scalar operations used to construct its final output polynomial in this way. Our algorithms use the well known fact that Pfaffians admit polynomial-size arithmetic circuits that only use addition and multiplication operations [22].

► **Proposition 8 (Pfaffian Construction).** *In $\text{poly}(n)$ time we can construct a division-free arithmetic circuit of $\text{poly}(n)$ size over $\mathbb{F}$ whose inputs are indeterminate entries of an $n \times n$ skew-symmetric matrix $\mathbf{A}$, and whose output is the polynomial $\text{Pf } \mathbf{A}$.*

We make use of the following classic result, proven for example in [19, Theorem 7.2].

► **Proposition 9** (Identity Testing). *Let P be a nonzero polynomial over a finite field $\mathbb{F}$ of degree at most d . If each variable of P is assigned an independent, uniform random value from $\mathbb{F}$, then the corresponding evaluation of P is nonzero with probability at least $1 - d/|\mathbb{F}|$.*

Subset Convolution. Let S be a set of size p . Let α and β be functions from the collection of subsets of S to $\mathbb{F}$.

The *subset convolution* $(\alpha * \beta)$ of α and β is the function defined by setting

$$(\alpha * \beta)(T) = \sum_{\substack{A, B \subseteq S \\ T = A \sqcup B}} \alpha(A) \beta(B) \quad \text{for all } T \subseteq S.$$

For each $v \in S$, introduce a variable y_v . Let $Y = \{y_v\}_{v \in S}$ be this set of variables. Consider polynomials

$$P = \sum_{A \subseteq S} \alpha(A) \prod_{a \in A} y_a \quad \text{and} \quad Q = \sum_{B \subseteq S} \beta(B) \prod_{b \in B} y_b$$

over the quotient ring $R = \mathbb{F}[Y]/\langle (y^2)_{y \in Y} \rangle$. That is, R is the usual ring of polynomials over the variables in Y with coefficients from $\mathbb{F}$, except that whenever we multiply the same variable with itself the resulting product vanishes.

We refer to R as the *squarefree ring* over Y . By definition of polynomial multiplication and the fact that only squarefree monomials survive in R , we get that

$$P \cdot Q = \sum_{T \subseteq S} (\alpha * \beta)(T) \prod_{t \in T} y_t$$

over R . This shows that computing subset convolutions is equivalent to computing products of polynomials over R . The following result then follows from known $O^*(2^p)$ time algorithms for subset convolution [4].

► **Proposition 10** (Fast Subset Convolution). *We can compute addition and multiplication over the squarefree ring R on p variables in $O^*(2^p)$ time.*

3 Triangle Modulator

In this section, we prove the following key lemma.

► **Lemma 6.** *Given a $\overline{K_3}$ -free modulator of size p and any positive integer k , the k -COLORING problem can be solved in randomized $O^*(2^p)$ time.*

Proof. Replace the input graph with its complement. Then by Observation 7, it suffices to show that given a K_3 -free modulator S of size p for G and a positive integer k , we can solve k -CLIQUE COVER on G in $O^*(2^p)$ time.

We write $\overline{S} = V \setminus S$. By definition, $G[\overline{S}]$ contains no triangle (i.e., a copy of K_3). This means that for every clique C in G , we have $|C \cap \overline{S}| \in \{0, 1, 2\}$. Given a clique cover $\mathcal{C}$ of G , we say it has type $\vec{t} = (t_0, t_1)$ if for each $i \in \{0, 1\}$ the cover $\mathcal{C}$ contains exactly t_i cliques C satisfying $|C \cap \overline{S}| = i$. Note that if a cover has type $\vec{t}$, then because every vertex appears in a unique clique in the cover, exactly $t_2 = (|\overline{S}| - t_1)/2$ cliques in the cover intersect $\overline{S}$ at two nodes. Consequently, the size of a clique cover with type $\vec{t}$ is exactly

$$t_0 + t_1 + t_2 = t_0 + (t_1 + |\overline{S}|)/2 = t_0 + (t_1 + n - p)/2.$$

We say a type $\vec{t} = (t_0, t_1)$ is *valid* if and only if we have $t_0 + (t_1 + n - p)/2 \leq k$ so that a clique cover of type $\vec{t}$ has size at most k .

Our task is to determine if G contains a clique cover of size at most k . Such a cover has at most $O(k^2) \leq \text{poly}(n)$ valid types $\vec{t}$, and we try out each possible such type and look for a clique cover with exactly that type.

To that end, fix a valid type $\vec{t} = (t_0, t_1)$. We now construct an auxiliary graph H , such that perfect matchings in H encode how clique covers of G with type $\vec{t}$ may restrict to clique covers of $G[\bar{S}]$. We include the vertex set $\bar{S}$ in H , and add all edges from $G[\bar{S}]$ to H . We then introduce a set U of t_1 new vertices, and add edges from every node in U to every vertex in $\bar{S}$. Intuitively, when we take a perfect matching M of H , it will correspond to a clique cover $\mathcal{C}$ of G such that

- for each edge $\{u, v\}$ with $u \in U$, there is a clique $C \in \mathcal{C}$ with $C \cap \bar{S} = \{v\}$, and
- for each edge $\{v, w\} \in M$ with $v, w \in \bar{S}$, there is a clique $C \in \mathcal{C}$ with $C \cap \bar{S} = \{v, w\}$.

With this correspondence in mind, we move to constructing a polynomial that will enumerate clique covers of G .

Let $E(H)$ denote the edge set of H . For each $e \in E(H)$, introduce an indeterminate variable x_e . Now consider the matrix $\mathbf{A}$ whose rows and columns are indexed by nodes of H , with entries defined by

$$\mathbf{A}[v, w] = \begin{cases} x_{vw} & \text{if } \{v, w\} \in E(H) \text{ and } v \prec w \\ -x_{vw} & \text{if } \{v, w\} \in E(H) \text{ and } w \prec v \\ 0 & \text{otherwise} \end{cases}$$

where $(\prec)$ is some fixed ordering on the nodes of H .

By Equation (1) and the definition of $\mathbf{A}$ above, we have

$$\text{Pf } \mathbf{A} = \sum_{M \in \Pi(H)} \text{sgn}(M) \prod_{e \in M} x_e \quad (2)$$

where here $\Pi(H)$ denotes the set of perfect matchings in H , and $\text{sgn}(M) \in \{-1, 1\}$ for each choice of M .

Next, we will introduce additional, larger polynomial expressions that we will substitute in for the x_e variables, with the end goal of transforming the matching polynomial $\text{Pf } \mathbf{A}$ into a polynomial that enumerates collections of cliques in G .

Go through all subsets $T \subseteq S$, and for each check if T is a clique in G . This lets us determine the collection $\mathcal{F}$ of all cliques in S in $O^*(2^p)$ time overall.

Then for each edge $e \in E(H)$ and clique $C \subseteq S$, we introduce a variable z_{eC} . For each $i \in [t_0]$ and clique $C \subseteq S$, also introduce a variable z_{iC} . Let Z be the set of all these z_{eC} and z_{iC} variables.

Additionally, for each node $v \in S$, we introduce a variable y_v . Let $Y = \{y_v\}_{v \in S}$ denote this set of k variables.

Now for each $e \in E(H)$ and subset $T \subseteq S$, define the polynomial

$$\Phi_e[T] = \sum_{\substack{C \in \mathcal{F} \\ C \subseteq T}} z_{eC} \prod_{v \in C} y_v. \quad (3)$$

Intuitively this polynomial enumerates those cliques $C \subseteq T$ that can be extended using vertices in $e \cap \bar{S}$ to obtain larger cliques in G .

Also for each $i \in [t_0]$, define the polynomial

$$\Phi_i = \sum_{C \in \mathcal{F}} z_{iC} \prod_{v \in C} y_v. \quad (4)$$

Intuitively these polynomials enumerate cliques $C \subseteq S$ that we do not plan on extending with nodes in $\bar{S}$.

For each vertex $v \in V$, let $N(v)$ denote the set of vertices in S adjacent to v in G . We construct a new matrix $\mathbf{B}$, by starting with the matrix $\mathbf{A}$ and

- for each $e = \{u, v\} \in E(H)$ with $u \in U$ and $v \in \bar{S}$, substituting x_{uv} with $\Phi_e[N(v)]$, and
- for each $e = \{v, w\} \in E(H)$ with $v, w \in \bar{S}$, substituting x_{vw} with $\Phi_e[N(v) \cap N(w)]$.

Now define the polynomial

$$F = (\text{Pf } \mathbf{B}) \cdot \prod_{i=1}^{t_0} \Phi_i \quad (5)$$

over $R[Z]$, where $R = \mathbb{F}[Y]/\langle (y^2)_{y \in Y} \rangle$ is the squarefree ring defined in Section 2.

▷ **Claim 11 (Polynomial Characterization).** The polynomial F has a monomial divisible by $\prod_{v \in S} y_v$ if and only if G has a clique cover of type $\vec{t} = (t_0, t_1)$.

Proof. Suppose first that F has a monomial divisible by $\prod_{v \in S} y_v$. Since we work over the squarefree ring R , this monomial is of the form

$$f(Z) \prod_{v \in S} y_v \quad (6)$$

for some polynomial $f \in \mathbb{F}[Z]$. Moreover, from Equation (5) such a monomial in F must have been generated by taking the product of monomials from $\text{Pf } \mathbf{B}$ and each of the Φ_i such that the appearance of variables from Y in these monomials partition Y .

For each $i \in [t_0]$, let C_i be the set of vertices $v \in S$ such that the y_v variable was used by the monomial from Φ_i selected to help generate the expression from Equation (6) in the expansion of the product from Equation (5). Then by the partition property, the C_i are pairwise disjoint. From the definition of Φ_i in Equation (4), we get that each C_i is a clique.

Now, consider the monomial selected from $\text{Pf } \mathbf{B}$ to help generate the expression from Equation (6) in the expansion of the product from Equation (5). By Equation (2) and the definition of $\mathbf{B}$ in terms of $\mathbf{A}$, this monomial corresponds to a matching $M \in \Pi(H)$. Split $M = M_1 \sqcup M_2$ by letting M_b retain the edges in M which have exactly b endpoints in $\bar{S}$ for each $b \in [2]$. Since H has vertex set $\bar{S} \sqcup U$ and $|U| = t_1$, we get that M_1 has exactly t_1 edges, and each of its edges has one endpoint in U and one endpoint in $\bar{S}$. Let $v_1, \dots, v_{t_1}$ be the nodes in $\bar{S}$ appearing as endpoints of edges in M_1 . For each $j \in [t_1]$, let e_j be the edge containing v_j in M_1 .

Since in $\mathbf{B}$ we take $\mathbf{A}$ and replace each x_{e_j} variable with $\Phi_{e_j}[N(v_j)]$, the product over the edges of M_1 from Equation (2) contributes the variables y_v precisely for those vertices v appearing in the union of some choice of cliques $D_1, \dots, D_{t_1} \subseteq S$ with the property that the set $D_j \sqcup \{v_j\}$ is a clique in G for each j . Since we work over R , the cliques C_i and D_j are mutually disjoint. Moreover, since $C_i \subseteq \bar{S}$ for all i , we actually have that the cliques C_i and $D_j \sqcup \{v_j\}$ are collectively vertex-disjoint.

Set $t_2 = (|\bar{S}| - t_1)/2$, and let $\tilde{e}_1, \dots, \tilde{e}_{t_2}$ be the edges of M_2 . For each $\ell \in [t_2]$, let $\tilde{N}_\ell$ be the set of common neighbors in S of the endpoints of the edge $\tilde{e}_\ell$. Since in $\mathbf{B}$ we take $\mathbf{A}$ and replace each $x_{\tilde{e}_\ell}$ variable with $\Phi_{\tilde{e}_\ell}[\tilde{N}_\ell]$, the product over the edges of M_2 from Equation (2) contributes the variables y_v precisely for those vertices v appearing in the union of some choice of cliques $\tilde{D}_1, \dots, \tilde{D}_{t_2} \subseteq S$ with the property that the set $\tilde{D}_\ell \sqcup \tilde{e}_\ell$ is a clique in G for each ℓ . Since we work over R , the C_i , D_j , and $\tilde{D}_\ell$ are mutually disjoint. Moreover, since $M = M_1 \sqcup M_2$ is a matching in S , we in fact know that the C_i , $D_j \sqcup \{v_j\}$, and $\tilde{D}_\ell \sqcup \tilde{e}_\ell$ are all vertex-disjoint cliques in G .

Combining this vertex-disjoint condition with the facts that M is a perfect matching on $\bar{S}$ and that the monomial in Equation (6) generated by the products of the monomials of all the cliques we have listed so far is divisible by $\prod_{v \in S} y_v$, we get that the C_i , $D_j \sqcup \{v_j\}$, and $\tilde{D}_\ell \sqcup \tilde{e}_\ell$ cliques taken together form a clique cover of G of type $\vec{t} = (t_0, t_1)$ as claimed.

Conversely, suppose G has a clique cover of type $\vec{t} = (t_0, t_1)$. Set $t_2 = (|\bar{S}| - t_1)/2$.

Let $C_1, \dots, C_{t_0}$ be the cliques in this cover intersecting $\bar{S}$ at zero nodes, $D_1, \dots, D_{t_1}$ be the cliques in this cover intersecting $\bar{S}$ at one node, and $\tilde{D}_1, \dots, \tilde{D}_{t_2}$ be the cliques in this cover intersecting $\bar{S}$ at two nodes. For each $i \in [t_0]$, select the monomial from Φ_i in Equation (4) corresponding to the clique C_i .

For each $j \in [t_1]$, let v_j be the unique vertex from $\bar{S}$ in D_j . For each $\ell \in [t_2]$, let $\tilde{e}_\ell$ be the subset of two vertices in $\bar{S}$ contained in $\tilde{D}_\ell$. Since $\tilde{D}_\ell$ is a clique, $\tilde{e}_\ell$ is an edge. Let $\tilde{N}_\ell$ be the set of common neighbors in $\bar{S}$ of the endpoints of $\tilde{e}_\ell$. The v_j and $\tilde{e}_\ell$ are vertex-disjoint and account for all vertices in $\bar{S}$, because we are assuming we are starting with a clique cover. So we can take a perfect matching M of H which includes all the $\tilde{e}_\ell$ edges, and pairs each v_j with a different node in U (here we use the fact that U has exactly t_1 nodes). Let e_j be the edge containing v_j in M for each $j \in [t_1]$.

Then we can select the monomial of $\text{Pf } \mathbf{B}$ corresponding to choosing the summand for M in Equation (2), choosing the summand for the cliques $D_j \setminus \{v_j\}$ in the sums for $\Phi_{e_j}[N(v_j)]$ given by Equation (3), and choosing the summands for the cliques $\tilde{D}_\ell \setminus \tilde{e}_\ell$ in the sums for $\Phi_{\tilde{e}_\ell}[\tilde{N}_\ell]$ given by Equation (3) again.

Then from the clique cover assumption, the product of all the monomials we selected from the Φ_i and $\text{Pf } \mathbf{B}$ will be divisible by $\prod_{v \in S} y_v$. Moreover, the product of the variables from Z appearing in this monomial uniquely recover the cliques C_i , D_j , and $\tilde{D}_\ell$, as well as the matching M , because the z_{iC} variables record the ordering of the C_i cliques, and the z_{eC} variables annotate the edges e of the matchings corresponding to the $D_j \cap \bar{S}$ and $\tilde{D}_\ell \cap \bar{S}$ sets. Thus the monomial generated in this way cannot be cancelled out by any other term in the expansion of F in Equation (5), which proves the desired result. $\triangleleft$

Having established Claim 11, it suffices to show that we can test that F has a monomial divisible by $\prod_{v \in S} y_v$ in $O^*(2^p)$ time. To do this, we will take a random evaluation of the variables in Z , and then use fast subset convolution over the variables in Y .

For all $i \in [t_0]$ and cliques $C \subseteq S$, we pick independent, uniform random $\xi_{iC} \in \mathbb{F}$. Independently from those values, for all $e \in E(H)$ and $C \subseteq S$ we also pick independent, uniform random $\xi_{eC} \in \mathbb{F}$. For each $i \in [t_0]$, let

$$\varphi_i = \sum_{C \in \mathcal{F}} \xi_{iC} \prod_{v \in C} y_v. \quad (7)$$

be the result of evaluating the Z variables of Φ_i on the ξ_{iC} values. We can compute each φ_i in $O^*(2^p)$ time because we have already precomputed the collection $\mathcal{F}$ of cliques in S .

Similarly, for each $e \in E(H)$ and $T \subseteq S$, let

$$\varphi_e[T] = \sum_{\substack{C \in \mathcal{F} \\ C \subseteq T}} \xi_{eC} \prod_{v \in C} y_v \quad (8)$$

be the result of evaluating the Z variables of $\Phi_e[T]$ on the ξ_{eC} values. Since we precomputed all the cliques contained in S , for any fixed $e \in E(H)$ and $T \subseteq S$ we can use Equation (8) to compute $\varphi_e[T]$ in $O^*(2^p)$ time. In particular, we can compute $\varphi_e[T]$ for all $e \in E(H)$ and $T \subseteq S$ of the form $T = N(v)$ for some $v \in \bar{S}$ or $T = N(v) \cap N(w)$ for some $v, w \in \bar{S}$ in $O^*(2^p)$ time, because there are $\text{poly}(n)$ choices for e and T in this case.

Having computed all these values, we can substitute the x_e for the relevant values $\varphi_e[T]$ values needed to turn $\mathbf{A}$ into $\mathbf{B}$ (with respect to our random evaluation of the variables in Z). Now use Proposition 8 to obtain a polynomial-size, division-free arithmetic circuit for the Pfaffian, and feed the entries of $\mathbf{B}$ we have just computed into it. Each addition and multiplication operation for this arithmetic circuit is now over the squarefree ring R , which by Proposition 10 takes $O^*(2^p)$ time. Since there are only $\text{poly}(n)$ such operations, we compute this random evaluation of $\text{Pf } \mathbf{B}$ over R in $O^*(2^p)$ time overall. We then compute the product of this with all the φ_i again in $O^*(2^p)$ time by Proposition 10, which by Equation (5) gives us the value of the polynomial F under our random evaluation to its Z variables.

Since F has degree at most $\text{poly}(n)$ in its Z variables, applying Proposition 9 to the coefficient of $\prod_{v \in S} y_v$ in F (viewed as a polynomial in Z), we see that by picking the size of the field $\mathbb{F}$ to be a sufficiently large polynomial in n , with high probability the random evaluation of F we computed has the monomial $\prod_{v \in S} y_v$ if and only if the original polynomial F has a monomial divisible by $\prod_{v \in S} y_v$. So by Claim 11 checking if this random evaluation of F has the monomial $\prod_{v \in S} y_v$ solves the k -COLORING problem in $O^*(2^p)$ time with high probability. This proves the desired result. $\blacktriangleleft$

4 Below Guarantee Parameterizations

In this section, we present our algorithm for DUAL COLORING.

► **Theorem 1.** DUAL COLORING can be solved in randomized $O^*(2^{3k/2})$ time.

Proof. We replace the graph G with its complement. Then by Observation 7, it suffices to solve the DUAL CLIQUE COVER problem.

We first construct a maximal collection $\mathcal{T}$ of vertex-disjoint triangles in G . Since we can find a triangle in G if one exists in polynomial time, we can construct $\mathcal{T}$ by repeatedly finding a triangle T in G , including T in $\mathcal{T}$, then deleting the vertices of T from G and repeating this process, until we determine that no triangles are left. We find at most $n/3$ triangles in this way, so this takes polynomial time overall.

Let $t = |\mathcal{T}|$. Suppose $t \geq k/2$. Then taking the triangles in $\mathcal{T}$ together with the single-node sets consisting of each vertex v not used by a triangle in $\mathcal{T}$ yields a clique cover for G of size $t + (n - 3t) = n - 2t \leq n - k$. Thus, we can solve DUAL CLIQUE COVER by returning the cover described above.

Otherwise, we have $t < k/2$. In this case, maximality of $\mathcal{T}$ implies that the set $S = \bigsqcup_{T \in \mathcal{T}} T$ is a K_3 -free modulator for G of size $3t$. Then by Observation 7 and Lemma 6 we can solve DUAL COLORING in this case in $O^*(2^{3t}) \leq O^*(2^{3k/2})$ time, as desired. $\blacktriangleleft$

Next, we establish (randomized) fixed-parameter tractability for a smaller parameter.

► **Theorem 3** (Parameterization from Stronger Guarantee). *Given an integer $k \geq 1$, and a graph G that has maximum clique size ω and whose complement has maximum matching size $\bar{\mu}$, we can solve $(\omega + \bar{\mu} - k)$ -COLORING in randomized $O^*(2^{6k})$ time.*

Proof. We replace the graph G with its complement. Let $\mu = \mu(G)$ and $\alpha = \alpha(G)$ be the size of a maximum matching and maximum independent set in the new graph respectively. Since complementing the graph turns cliques into independent sets, by Observation 7 it suffices to show that we can solve $(\alpha + \mu - k)$ -CLIQUE COVER in $O^*(2^{6k})$ time.

We first construct a maximal collection $\mathcal{T}$ of vertex-disjoint triangles in G . Since finding a triangle in G , if it exists, takes polynomial time, and $\mathcal{T}$ can have at most $n/3$ triangles, we can obtain $\mathcal{T}$ in polynomial time using a greedy algorithm that repeatedly finds triangles, includes them in $\mathcal{T}$, deletes the obtained triangles from G , and continues in this fashion until we are left with a triangle-free graph.

Let $t = |\mathcal{T}|$. Suppose that $t \leq 2k$. In this case, define the set $S = \bigsqcup_{T \in \mathcal{T}} T$ of all vertices participating in the triangles of $\mathcal{T}$. Then by maximality of $\mathcal{T}$, the set S is a K_3 -free modulator of G of size $3t$. In this case, by Observation 7 and Lemma 6 we can solve $(\alpha + \mu - k)$ -CLIQUE COVER in $O^*(2^{3t}) \leq O^*(2^{6k})$ time as claimed.

Otherwise, $t > 2k$. In this case, take $\tilde{\mathcal{T}} \subseteq \mathcal{T}$ consisting of exactly $2k$ triangles, and define $S = \bigsqcup_{T \in \tilde{\mathcal{T}}} T$ to be a set of $6k$ vertices making up $2k$ of the triangles in $\mathcal{T}$. We show that in this case, a clique cover of size at most $(\alpha + \mu - k)$ always exists in G .

Write $\bar{S} = V \setminus S$. Let M_{out} be a maximum matching in $G[\bar{S}]$. We can find M_{out} in polynomial time. Let $\mu_{\text{out}} = |M_{\text{out}}|$ be the size of this matching. Let I be the set of vertices in $\bar{S}$ that do not appear as an endpoint of an edge in M_{out} . By maximality of M_{out} , the set I forms an independent set in G . Write $\alpha_{\text{out}} = |I|$.

▷ **Claim 12.** We have $\mu_{\text{out}} + \alpha_{\text{out}} \leq \alpha + \mu - 3k$.

Proof. We will prove the claim by establishing several inequalities concerning structural parameters in G .

First, construct a maximum matching M_{in} in $G[S \sqcup I]$ in polynomial time. Let $\mu_{\text{in}} = |M_{\text{in}}|$ be the size of this matching. Since M_{out} is in $G[\bar{S} \setminus I]$ by definition, we see that $M_{\text{in}} \sqcup M_{\text{out}}$ is a matching in G of size $(\mu_{\text{in}} + \mu_{\text{out}})$.

Since μ is the size of a maximum matching in G , we get that

$$\mu_{\text{in}} + \mu_{\text{out}} \leq \mu. \quad (9)$$

Note that since α is the maximum size of independent sets in G , we have

$$\alpha_{\text{out}} \leq \alpha. \quad (10)$$

Now, by maximality of M_{in} in $G[S \sqcup I]$, we know that the set of vertices in $S \sqcup I$ not participating as an endpoint in M_{in} forms an independent set in G . This set then has size $(\alpha_{\text{out}} + 6k - 2\mu_{\text{in}})$. Since α is the maximum size of independent sets in G , we have

$$\alpha_{\text{out}} + 6k - 2\mu_{\text{in}} < \alpha. \quad (11)$$

Now, we can write

$$\mu_{\text{out}} + \alpha_{\text{out}} = \mu_{\text{out}} + (1/2)\alpha_{\text{out}} + (1/2)\alpha_{\text{out}}. \quad (12)$$

By applying Equations (9)–(11) to the first through third terms on the right-hand side of Equation (12) respectively, we can bound

$$\mu_{\text{out}} + \alpha_{\text{out}} \leq (\mu - \mu_{\text{in}}) + (1/2)\alpha + (1/2)(\alpha + 2\mu_{\text{in}} - 6k) = \alpha + \mu - 3k$$

which proves the desired result. ◁

By definition, every vertex in $\bar{S}$ either appears as an endpoint of M_{out} , or belongs to I . Then using the triangles in $\tilde{\mathcal{T}}$ to cover the vertices in S , and taking the edges from M_{out} and the individual nodes from I to cover the vertices in $\bar{S}$, we obtain a clique cover for G of size

$$2k + \mu_{\text{out}} + \alpha_{\text{out}} \leq \alpha + \mu - k$$

by Claim 12. So we can solve $(\alpha + \mu - k)$ -CLIQUE COVER just by returning this cover. ◀

Curiously, previous work showed that INDUCED MATCHING (the problem of finding a 1-regular induced subgraph on k vertices) is also fixed-parameter tractable in k for the same below-guarantee parameter $\alpha + \mu - k$ as in the proof above [15].

Hardness. We now present our lower bound, demonstrating that although Theorem 3 shows we can get an efficient parameterized algorithm for graph coloring below $(\omega + \bar{\mu})$, we do not expect an analogous algorithm to exist for coloring below the quantity $\bar{\mu}$ on its own.

► **Theorem 4.** *The $(n/2 - k)$ -COLORING problem is W[1]-hard in the parameter k on graphs with n vertices whose complement has a perfect matching.*

Proof. By replacing the graph with its complement and applying Observation 7, it suffices to show that $(n/2 - k)$ -CLIQUE COVER is W[1]-hard on graphs containing a perfect matching.

We prove this result by reducing from k -COLORED CLIQUE. In this problem, we are given a k -partite graph G on kn vertices, with vertex set $V = V_1 \sqcup \dots \sqcup V_k$ partitioned into parts V_i consisting of n nodes each, and are tasked with determining if G contains a clique on k vertices. This problem is known to be W[1]-hard [9, Theorem 13.25].

Take an instance G of k -COLORED CLIQUE.

For each $i \in [k]$, we order the vertices in $V_i = \{v_{i1}, \dots, v_{in}\}$.

We construct a larger graph $\tilde{G}$ that contains G as a subgraph, in addition to some auxiliary nodes and edges we describe next. For each $i \in [k + 2]$, we introduce a new node u_i in $\tilde{G}$. For each $i \in [k]$ and $j \in [n - 1]$, we introduce a new node w_{ij} in $\tilde{G}$. We add edges between all the u_i nodes. For all $i \in [k]$, we add an edge from u_i to v_{i1} . For all $i \in [k]$ and $j \in [n - 1]$ we add edges from w_{ij} to v_{ij} and $v_{i(j+1)}$.

Conceptually, this construction of $\tilde{G}$ from G involves adding a new clique of size $(k + 2)$ to the graph on the u_i vertices, and for each $i \in [k]$ introducing a path

$$P_i = \langle u_i, v_{i1}, w_{i1}, v_{i2}, w_{i2}, \dots, w_{i(n-1)}, v_{in} \rangle \quad (13)$$

beginning at u_i , and alternating between nodes in V_i and W_i according to the orders of the vertices in these sets. By design, for any $j \in [n]$, we can always split the path

$$P_i = \langle u_i \rangle \diamond A_{ij} \diamond \langle v_{ij} \rangle \diamond B_{ij} \quad (14)$$

into its first node, a path A_{ij} on an even number of vertices, the node v_{ij} , and a suffix B_{ij} on an even number of vertices. Moreover, A_{ij} and B_{ij} consist of $2n - 2$ nodes altogether in this decomposition.

Note that the constructed graph $\tilde{G}$ has a perfect matching, by taking the edges $\{u_i, v_{i1}\}$ for all $i \in [k]$ and $\{w_{ij}, v_{i(j+1)}\}$ for all $i \in [k]$ and $j \in [n - 1]$.

From its definition, $\tilde{G}$ has $N = 2kn + 2$ nodes. Set the parameter $\ell = k(n - 1) + 2$.

► **Claim 13.** G has a clique on k vertices if and only if $\tilde{G}$ has a clique cover of size ℓ .

Proof. Suppose that G has a clique of size k . Let $(v_{1j_1}, \dots, v_{kj_n}) \in V_1 \times \dots \times V_k$ be the k -tuple of vertices participating in this clique.

For each $i \in [k]$, by Equation (14) we can split each path P_i into

$$\langle u_i \rangle \diamond A_{ij_i} \diamond \langle v_{ij_i} \rangle \diamond B_{ij_i}$$

its first node, two paths A_{ij_i} and B_{ij_i} on even numbers of vertices with $2n - 2$ nodes total, and the node v_{ij_i} . For each $i \in [k]$, we can cover the nodes in A_{ij_i} and B_{ij_i} using the endpoints of the edges from a matching of size $(n - 1)$. This shows that with $k(n - 1)$ cliques, we can cover all vertices in $\tilde{G}$ in a disjoint fashion, except the u_i and v_{ij_i} nodes. We include all the u_i nodes in one clique and all the v_{ij_i} nodes in another clique to then obtain a clique cover of size $\ell = k(n - 1) + 2$ of $\tilde{G}$ as claimed.

Conversely, suppose $\tilde{G}$ has a clique cover of size ℓ . Note that the w_{ij} vertices, with indices ranging over $i \in [k]$ and $j \in [n-1]$, form an independent set. Consequently, covering the w_{ij} requires using $k(n-1)$ distinct cliques. Now, each w_{ij} has degree two in $\tilde{G}$, and the neighbors of w_{ij} are non-adjacent in $\tilde{G}$ by definition. Consequently, the cliques used to cover the w_{ij} collectively cover at most $2k(n-1)$ vertices in $\tilde{G}$.

In the assumed clique cover of size ℓ from $\tilde{G}$, this implies that a set of $\ell - k(n-1) = 2$ distinct cliques are used to cover the at least $N - 2k(n-1) = 2k+2$ remaining nodes in $\tilde{G}$ not covered by the cliques that contain the w_{ij} vertices. Since node u_{k+2} is adjacent to only other u_i nodes, the clique that covers u_{k+2} must solely consist of u_i nodes. Without loss of generality, we may then assume the clique covering u_{k+2} contains all u_i nodes. After removing the nodes covered by this clique, we have one clique left, that must be used to cover the at least $(2k+2) - (k+2) = k$ remaining nodes in $\tilde{G}$. Moreover, this clique uses no u_i nodes or w_{ij} nodes. Consequently, the last clique must be a clique of size k in the v_{ij} nodes, which pulls back to the a clique of size k in G . This proves the claim. $\triangleleft$

Observe that $N/2 - \ell = (k+1)$, so that $\ell = N/2 - (k+1)$.

By Claim 13, solving k -COLORED CLIQUE problem on graphs with n vertices reduces to solving $(N/2 - (k+1))$ -CLIQUE COVER on graphs with $N = 2kn + 2$ nodes and a perfect matching. Hence the reduction is efficient enough to imply that $(n/2 - k)$ -CLIQUE COVER is $W[1]$ -hard in graphs with perfect matchings, which proves the desired result. $\blacktriangleleft$

5 Conclusion

In this paper, we presented a faster parameterized algorithm for DUAL COLORING, improving the runtime from $O^*(4^k)$ to $O^*(2^{3k/2}) \leq O^*(2.83^k)$. We also introduced a new below-guarantee parameterization for graph coloring, which can be viewed as a harder version of DUAL COLORING. We showed that this problem can be solved in $2^{O(k)}$ poly(n) time as well, and noted that a closely related parameterization is in contrast $W[1]$ -hard.

The most relevant open question to this work is: what is the true parameterized complexity of DUAL COLORING? The current best algorithms for k -COLORING take $O^*(2^n)$ time for general k , and without improving this runtime we cannot hope to solve DUAL COLORING in faster than $O^*(2^k)$ time. Can we indeed achieve a $O^*(2^k)$ runtime for DUAL COLORING?

A natural strategy to obtain faster algorithms for DUAL COLORING would be to try and strengthen Lemma 6 further, by obtaining efficient coloring algorithms parameterized by H -free modulators for larger pattern graphs H . For example, obtaining an analogue of Lemma 6 for $\overline{K_4}$ -free modulators would immediately imply a faster DUAL COLORING algorithm by following the framework outlined in Section 1.1. Unfortunately, this simple attempt at generalizing Lemma 6 does not seem possible. This is because solving k -COLORING in graphs without induced copies of $\overline{K_4}$ is NP-hard in general [16, Theorem 1], so that assuming $P \neq NP$ we cannot hope to solve k -COLORING in graphs with $\overline{K_4}$ -free modulators of size p in $f(p)$ poly(n) time, for any function $f(\cdot)$. As mentioned in the Section 1, $(n-k)$ -SET COVER can be solved in $O^*(2^{3k/2})$ time using a different method. Can this approach be combined with our triangle packing argument to obtain a faster DUAL COLORING algorithm?

Another interesting research direction is to obtain faster algorithms for *multiplicative* below-guarantee graph coloring. For any parameter $\delta \in [1/3, 1]$, by setting $k = (1 - \delta)n$ in Theorem 1 we get that (δn) -COLORING can be solved in $O^*(2^{(1 - (\delta - 1/3)n)})$ time. Using very different techniques, previous work showed that for $\delta \in [0, 1]$ it is possible to distinguish between the cases where the input graph G admits a proper coloring with $(\delta n - 1)$ colors

and the case where any proper coloring of G needs at least $(\delta n + 1)$ colors in $O^*(2^{(1-\Omega(\delta^4))n})$ time [21, Theorem 1.2]. Can the dependence on δ be improved and the need for additive approximation here be removed, to solve (δn) -COLORING in $O^*(2^{(1-\Omega(\delta))n})$ time in general?

Finally, we showed that for DUAL COLORING and its harder variant, constructing a maximal triangle packing instead of a maximal matching can accelerate algorithms for these problems. Can this method be used more generally to help improve parameterized algorithms or kernelization for other problems where the current best methods rely on matching-based techniques such as crown decomposition?

References

- 1 Shyan Akmal and Tomohiro Koana. Faster edge coloring by partition sieving. In *Proceedings of the 42nd International Symposium on Theoretical Aspects of Computer Science, STACS 2025*, volume 327 of *LIPICs*, pages 7:1–7:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICs.STACS.2025.7.
- 2 Vasily Alferov, Ivan Bliznets, and Kirill Brilliantov. Parameterization of (Partial) Maximum Satisfiability above Matching in a Variable-Clause Graph. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence, AAAI 2024*, pages 7918–7925. AAAI Press, 2024. doi:10.1609/AAAI.V38I8.28628.
- 3 Kenneth Appel and Wolfgang Haken. Every planar map is four colorable. I. Discharging. *Illinois J. Math.*, 21(3):429–490, 1977. URL: <https://projecteuclid.org/euclid.ijm/1256049012>.
- 4 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Fourier meets Möbius: fast subset convolution. In *Proceedings of the 39th annual ACM symposium on Theory of computing, STOC 2007*. ACM, June 2007. doi:10.1145/1250790.1250801.
- 5 Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set Partitioning via Inclusion-Exclusion. *SIAM Journal on Computing*, 39(2):546–563, January 2009. doi:10.1137/070683933.
- 6 Béla Bollobás, Paul A. Catlin, and Paul Erdős. Hadwiger’s Conjecture is True for Almost Every Graph. *Eur. J. Comb.*, 1(3):195–199, 1980. doi:10.1016/S0195-6698(80)80001-1.
- 7 Benny Chor, Mike Fellows, and David W. Juedes. Linear Kernels in Linear Time, or How to Save k Colors in $O(n^2)$ Steps. In *Proceedings of the 30th International Workshop on Graph-Theoretic Concepts in Computer Science*, volume 3353 of *WG 2004*, pages 257–269. Springer, 2004. doi:10.1007/978-3-540-30559-0_22.
- 8 Maria Chudnovsky, Neil Robertson, Paul D. Seymour, and Robin Thomas. The strong perfect graph theorem. *Ann. Math.*, 164(1):51–229, 2006. doi:10.4007/annals.2006.164.51.
- 9 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer International Publishing, 2015. doi:10.1007/978-3-319-21275-3.
- 10 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, December 2018. doi:10.1017/9781107415157.
- 11 M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, February 1976. doi:10.1016/0304-3975(76)90059-1.
- 12 Gregory Gutin, Diptapriyo Majumdar, Sebastian Ordyniak, and Magnus Wahlström. Parameterized Pre-Coloring Extension and List Coloring Problems. *SIAM Journal on Discrete Mathematics*, 35(1):575–596, January 2021. doi:10.1137/20m1323369.
- 13 Gregory Gutin and Matthias Mnich. A Survey on Graph Problems Parameterized Above and Below Guaranteed Values, 2024. arXiv:2207.12278v2.
- 14 Tommy R. Jensen and Bjarne Toft. *Graph Coloring Problems*. Wiley, 2011.
- 15 Tomohiro Koana. Induced matching below guarantees: Average paves the way for fixed-parameter tractability. In *Proceedings of the 40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023*, volume 254 of *LIPICs*, pages 39:1–39:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.STACS.2023.39.

- 16 Daniel Král, Jan Kratochvíl, Zsolt Tuza, and Gerhard J. Woeginger. Complexity of coloring graphs without forbidden induced subgraphs. In *Proceedings of the 27th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2001*, volume 2204 of *Lecture Notes in Computer Science*, pages 254–262. Springer, 2001. doi:10.1007/3-540-45477-2_23.
- 17 Wenjun Li, Yang Ding, Yongjie Yang, and Guozhen Rong. A $(2 + \varepsilon)k$ -vertex kernel for the dual coloring problem. *Theoretical Computer Science*, 868:6–11, May 2021. doi:10.1016/j.tcs.2021.03.035.
- 18 László Lovász. Coverings and colorings of hypergraphs. *Proceedings of the 4th Southeastern Conference of Combinatorics, Graph Theory, and Computing*, pages 3–12, 1973. URL: <https://cir.nii.ac.jp/crid/1572261549354589440>.
- 19 Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, August 1995. doi:10.1017/cbo9780511814075.
- 20 Kazuo Murota. *Matrices and matroids for systems analysis*, volume 20. Springer Science & Business Media, 1999.
- 21 Jesper Nederlof. Finding Large Set Covers Faster via the Representation Method. In *24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.ESA.2016.69.
- 22 Günter Rote. *Division-Free Algorithms for the Determinant and the Pfaffian: Algebraic and Combinatorial Approaches*, pages 119–135. Springer Berlin Heidelberg, 2001. doi:10.1007/3-540-45506-x_9.