On Structural Parameterizations of Star Coloring

Sriram Bhyravarapu[1](✉) and I. Vinod Reddy[2]

[1] The Institute of Mathematical Sciences, HBNI, Chennai, India
sriramb@imsc.res.in

[2] Department of Electrical Engineering and Computer Science,
IIT Bhilai, Bhilai, India
vinod@iitbhilai.ac.in

Abstract. A *star coloring* of a graph G is a proper vertex coloring such that every path on four vertices uses at least three distinct colors. The minimum number of colors required for such a star coloring of G is called star chromatic number, denoted by $\chi_s(G)$. Given a graph G and a positive integer k, the Star Coloring Problem asks whether G has a star coloring using at most k colors. This problem is NP-complete even on restricted graph classes such as bipartite graphs.

In this paper, we initiate a study of Star Coloring from the parameterized complexity perspective. We show that Star Coloring is fixed-parameter tractable when parameterized by (a) neighborhood diversity, (b) twin-cover, and (c) the combined parameters clique-width and the number of colors.

1 Introduction

A coloring $f : V(G) \rightarrow \{1, 2, \ldots, k\}$ of a graph $G = (V, E)$ is a *star coloring* if (i) $f(u) \neq f(v)$ for every edge $uv \in E(G)$, and (ii) every path on four vertices uses at least three distinct colors. The *star chromatic number* of G, denoted by $\chi_s(G)$, is the smallest integer k such that G is star colorable using k colors. Given a graph G and a positive integer k, the Star Coloring problem asks whether G has a star coloring using at most k colors. The name star coloring is due to the fact that the subgraph induced by any two color classes (subset of vertices assigned the same color) is a disjoint union of stars.

Star Coloring [14] is used in the computation of the Hessian matrix. A Hessian matrix is a square matrix of second order partial derivatives of a scalar-valued function. Hessian matrices are used in large-scale optimization problems, parametric sensitivity analysis [3], image processing, computer vision [22], and control of dynamical systems in real time [3]. Typically, Hessian matrices that arise in a large-scale application are sparse. The computation of a sparse Hessian matrix using the automatic differentiation technique requires a seed matrix. Coleman and Moré [5] showed that the computation of a seed matrix can be formulated using a star coloring of the adjacency graph of a Hessian matrix.

Star Coloring was first introduced by Grünbaum in [15]. The computational complexity of the problem is studied on several graph classes. The problem

A. Bagchi and R. Muthu (Eds.): CALDAM 2023, LNCS 13947, pp. 159–171, 2023.
https://doi.org/10.1007/978-3-031-25211-2_12

is polynomial time solvable on cographs [23] and line graphs of trees [24]. It is NP-complete to decide if there exists a star coloring of bipartite graphs [4] using at most k colors, for any $k \geq 3$. It has also been shown that Star Coloring is NP-complete on planar bipartite graphs [1] and line graphs of subcubic graphs [20] when $k = 3$. Recently, Shalu and Cyriac [26] showed that k-Star Coloring is NP-complete for graphs of degree at most four, where $k \in \{4, 5\}$.

To the best of our knowledge, the problem has not been studied in the framework of parameterized complexity. In this paper, we initiate the study of Star Coloring from the viewpoint of parameterized complexity. In parameterized complexity, the running time of an algorithm is measured as a function of input and a secondary measure called a parameter. A parameterized problem is said to be fixed-parameter tractable (FPT) with respect to a parameter k, if the problem can be solved in $f(k)n^{O(1)}$ time, where f is a computable function independent of the input size n and k is a parameter associated with the input instance. For more details on parameterized complexity, we refer the reader to the texts [9]. As Star Coloring is NP-complete even when $k = 3$, the problem is para-NP complete when parameterized by the number colors k. This motivates us to study the problem with respect to structural graph parameters, which measure the structural properties of the input graph. The parameter tree-width [25] introduced by Robertson and Seymour is one of the most investigated structural graph parameters for graph problems.

The Star Coloring problem is expressible in monadic second order logic (MSO) [17]. Using the meta theorem of Courcelle [6], the problem is FPT when parameterized by the tree-width of the input graph. Clique-width [8] is another graph parameter which is a generalization of tree-width. If a graph has bounded tree-width, then it has bounded clique-width, however, the converse may not always be true (e.g., complete graphs). Courcelle's meta theorem can also be extended to graphs of bounded clique-width. It was shown in [7] that all problems expressible in MSO logic that does not use edge set quantifications (called as MS_1-logic) are FPT when parameterized by the clique-width. However, the Star Coloring problem cannot be expressed in MS_1 logic [10,17]. Motivated by this, we study the parameterized complexity of the problem with respect to the combined parameters clique-width and the number of colors and show that Star Coloring is FPT.

Next, we consider the parameters neighborhood diversity [19] and twin-cover [13]. These parameters are weaker than clique-width in the sense that graphs of bounded neighborhood diversity (resp. twin-cover) have bounded clique-width, however, the converse may not always be true. Moreover, these two parameters are not comparable with the parameter tree-width and they generalize the parameter vertex cover [13] (see Fig. 1). We show that Star Coloring is FPT with respect to neighborhood diversity or twin-cover.

2 Preliminaries

For $k \in \mathbb{N}$, we use $[k]$ to denote the set $\{1, 2, \ldots, k\}$. If $f : A \rightarrow B$ is a function and $C \subseteq A$, $f|_C$ denotes the restriction of f to C, that is $f|_C : C \rightarrow B$ such

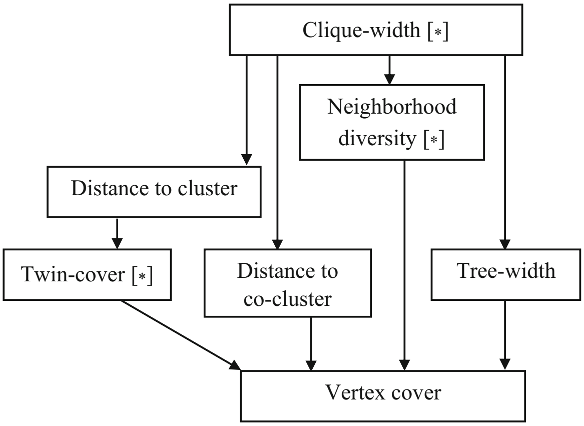

Fig. 1. Hasse diagram of some structural graph parameters. An edge from a parameter k_1 to a parameter k_2 means that there is a function f such that for all graphs G, we have $k_1(G) \leq f(k_2(G))$. The parameters considered in this paper are indicated by $*$.

that for all $x \in C$, $f|_C(x) = f(x)$. All graphs we consider in this paper are undirected, connected, finite and simple. For a graph $G = (V, E)$, we denote the vertex set and edge set of G by $V(G)$ and $E(G)$ respectively. We use n to denote the number of vertices and m to denote the number of edges of the graph. For simplicity, an edge between vertices x and y is denoted as xy. For a subset $X \subseteq V(G)$, the graph $G[X]$ denotes the subgraph of G induced by vertices of X. If $f : V(G) \rightarrow [k]$ is a coloring of G using k colors, then we use $f^{-1}(i)$ to denote the subset of vertices of G which are assigned the color i. For a subset $U \subseteq V(G)$, we use $f(U)$ to denote the set of colors used to color the vertices of U, i.e., $f(U) = \bigcup\limits_{u \in U} f(u)$.

For a vertex set $X \subseteq V(G)$, we denote $G - X$, the graph obtained from G by deleting all vertices of X and their incident edges. The open neighborhood of a vertex v, denoted $N(v)$, is the set of vertices adjacent to v and the set $N[v] = N(v) \cup \{v\}$ denotes the closed neighborhood of v. The neighbourhood of a vertex set $S \subseteq V(G)$ is $N(S) = (\cup_{v \in S} N(v)) \setminus S$. For a fixed coloring of G, we say a path is *bi-colored* if there exists a proper coloring of the path using two colors.

The proofs of the theorems marked with ($\star$) are presented in the full version of this paper [2].

3 Neighborhood Diversity

In this section, we show that Star Coloring is FPT when parameterized by neighborhood diversity. The key idea is to reduce star coloring on graphs of bounded neighborhood diversity to the integer linear programming problem (ILP). The latter is FPT when parameterized by the number of variables.

Theorem 1 ([11,18,21]). *The q-variable* INTEGER LINEAR PROGRAMMING FEASIBILITY *problem can be solved using $O(q^{2.5q+o(q)}n)$ arithmetic operations and space polynomial in n, where n is the number of bits of the input.*

We now define the parameter neighborhood diversity and state some of its properties.

Definition 1 (Neighborhood Diversity [19]). *Let $G = (V, E)$ be a graph. Two vertices $u, v \in V(G)$ are said to have the* same type *if and only if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. A graph G has neighborhood diversity at most t if there exists a partition of $V(G)$ into at most t sets $V_1, V_2, \ldots, V_t$ such that all vertices in each set have same type.*

Observe that each V_i either forms a clique or an independent set in G. Also, for any two distinct types V_i and V_j, either each vertex in V_i is adjacent to each vertex in V_j, or no vertex in V_i is adjacent to any vertex in V_j. We call a set V_i as a *clique type* (resp, independent type) if $G[V_i]$ is a clique (resp, independent set). It is known that a smallest sized partition of $V(G)$ into clique types and independent types can be found in polynomial time [19]. Hence, we assume that the types $V_1, V_2, \ldots, V_t$ of the graph G are given as input.

We now present the main result of the section.

Theorem 2. STAR COLORING *can be solved in $O(q^{2.5q+o(q)}n)$ time, where $q = 2^t$ and t is the neighborhood diversity of the graph.*

Let $G = (V, E)$ be a graph with the types $V_1, V_2, \ldots, V_t$. For each $A \subseteq \{1, 2, \ldots, t\}$, we denote a *subset type* of G by $T_A = \{V_i \mid i \in A\}$. We denote the set of all types adjacent to type V_i by $adj(V_i)$. That is, $V_j \in adj(V_i)$ if every vertex in V_j is adjacent to every vertex of V_i. Given a graph G and its types, we construct the ILP instance in the following manner.

Construction of the ILP Instance: For each $A \subseteq [t]$, let n_A be the variable that denotes the number of colors assigned to vertices in every type of T_A and not used in any of the types from $\{V_1, V_2, \ldots, V_t\} \setminus T_A$. For example, if $A = \{1, 3, 4\}$ (i.e., $T_A = \{V_1, V_3, V_4\}$) and $n_A = 2$, then there are two colors, say c_1 and c_2, such that both c_1 and c_2 are assigned to vertices in each of the types V_1, V_3 and V_4 and not assigned to any of the vertices in types $\{V_1, V_2, \ldots, V_t\} \setminus \{V_1, V_3, V_4\}$. This is the critical part of the proof where we look at how many colors are used exclusively in each type of T_A rather than what colors are used. Since we have a variable n_A for each $A \subseteq [t]$, the number of variables is 2^t.

We now describe the constraints for ILP with a short description explaining the significance or the information being captured by the constraints.

(C0) Discard all subset types T_A containing two types V_i, V_j where $V_j \in adj(V_i)$. To ensure that no two adjacent vertices are assigned the same color, we introduce this constraint that only considers T_A in which no two types in T_A are adjacent.

(C1) The sum of all the variables is at most k. That is $\sum_{A \subseteq [t]} n_A \leq k$.

We introduce this constraint to ensure that the number of colors used in any coloring is at most k.

(C2) For each clique type V_i, $i \in [t]$, the sum of the variables n_A for which $V_i \in T_A$ is equal to the number of vertices in V_i. That is, $\sum_{A: V_i \in T_A} n_A = |V_i|$.
To ensure that no two vertices in the clique type V_i are assigned the same color, we introduce this constraint.

(C3) For each independent type V_i, where $i \in [t]$, the sum of the variables n_A for which $V_i \in T_A$ is at least one and at most the minimum of k and the number of vertices in V_i. That is, $1 \leq \sum_{A: V_i \in T_A} n_A \leq \min\{k, |V_i|\}$.
To ensure that the number of colors used for coloring an independent type V_i is at least one and at most the minimum of k and $|V_i|$, we introduce this constraint.

(C4) For each combination of four distinct types, say $V_{i_1}, V_{i_2}, V_{i_3}$ and V_{i_4}, where $i_1, i_2, i_3, i_4 \in [t]$, with $V_{i_1}, V_{i_3} \in adj(V_{i_2})$ and $V_{i_4} \in adj(V_{i_3})$, we have the following constraint:
If the sum of the variables n_A for which $V_{i_1}, V_{i_3} \in T_A$ is at least one, then sum of variables n_B for which $V_{i_2}, V_{i_4} \in T_B$ should be equal to zero. That is,

$$\sum_{\substack{A: V_{i_1}, V_{i_3} \in T_A \text{ where} \\ V_{i_1}, V_{i_3} \in adj(V_{i_2}) \text{ and } V_{i_4} \in adj(V_{i_3})}} n_A \geq 1 \implies \sum_{B: V_{i_2}, V_{i_4} \in T_B} n_B = 0.$$

This constraint ensures that if there exists a vertex in V_{i_1} and a vertex in V_{i_3} that are assigned the same color, then the sets of colors used to color the vertices of V_{i_2} and V_{i_4} are disjoint.

(C5) For every combination of three distinct types, say $V_{i_1}, V_{i_2}, V_{i_3}$, where $i_1, i_2, i_3 \in [t]$, with V_{i_1} being an independent type and $V_{i_2}, V_{i_3} \in adj(V_{i_1})$, we have the following constraint:
If the sum of the variables n_A for which $V_{i_1} \in T_A$ is strictly less than the number of vertices in V_{i_1}, then the sum of variables n_B for which $V_{i_2}, V_{i_3} \in T_B$ is equal to zero.

$$\sum_{\substack{A: V_{i_1} \in T_A, \text{ where } V_{i_2}, V_{i_3} \in adj(V_{i_1}) \text{ and} \\ V_{i_1} \text{ is an independent type}}} n_A < |V_{i_1}| \implies \sum_{B: V_{i_2}, V_{i_3} \in T_B} n_B = 0.$$

This constraint ensures that if there exist two vertices in V_{i_1} that are assigned the same color, then every vertex in V_{i_2} is assigned a color different from every vertex in V_{i_3}.

(C6) For every combination of two distinct independent types V_{i_1}, V_{i_2}, where $i_1, i_2 \in [t]$ with $V_{i_1} \in adj(V_{i_2})$, if the sum of the variables n_A for which $V_{i_1} \in T_A$ is less than the number of vertices in V_{i_1}, then the sum of variables n_B for which $V_{i_2} \in T_B$ is equal to the number of vertices in V_{i_2}, and vice-versa. The former constraint is illustrated below while the latter constraint can be constructed by swapping V_{i_1} and V_{i_2} in the former constraint.

$$\sum_{\substack{A:V_{i_1}\in T_A \text{ where } V_{i_1}\in adj(V_{i_2})\\ \text{and } V_{i_1},V_{i_2} \text{ are independent types}}} n_A < |V_{i_1}| \implies \sum_{B:V_{i_2}\in T_B} n_B = |V_{i_2}|.$$

This constraint ensures that if there exist two vertices in V_{i_1} that are assigned the same color then all vertices in V_{i_2} are assigned distinct colors. We can say similar things for the latter constraint.

(C7) For each $A \subseteq [t]$, $n_A \geq 0$.
The number of colors used exclusively in all the types of T_A is at least 0.

The construction of the ILP instance is complete. We use Theorem 1 to obtain a feasible assignment for ILP. Using this, we find a star coloring of G. We now show that G has a star coloring using at most k colors if and only if there exists a feasible assignment to ILP.

Lemma 1. *If there exists a feasible assignment to ILP then G has a star coloring using at most k colors.*

Proof. Using a feasible assignment returned by the ILP, we construct a star coloring $f : V(G) \to [k]$ of G. Let $A_1, A_2, \ldots A_{2^t}$ be the subsets of $[t]$ in some fixed order. For each A_i, we associate the set of colors $c(A_i) = \{\sum_{j=0}^{i-1} n_{A_j} + 1, \sum_{j=0}^{i-1} n_{A_j} + 2, \ldots, \sum_{j=0}^{i-1} n_{A_j} + n_{A_i}\}$, where $n_{A_0} = 0$.

Now, for each V_j, we associate the set of colors $c(V_j) = \cup_{j\in A_i} c(A_i)$. If V_j is a clique type, then from constraint (C2), $|c(V_j)| = |V_j|$ for every j. Therefore, we color the vertices of V_j with distinct colors from the set $c(V_j)$. If V_j is an independent type, then from constraint (C3), $1 \leq |c(V_j)| \leq \min\{k, |V_j|\}$. In this case, we greedily color the vertices of V_j with colors from the set $c(V_j)$ such that each color is used at least once in V_j. This finishes the description of the coloring f of G.

We now argue that f is a star coloring of G. To show that f is a proper coloring, we need to show that every vertex is assigned a color and adjacent vertices do not receive the same color. The coloring process described above ensures that every vertex is colored. Also, f is a proper coloring because of the constraints (C0) and (C2). The former constraint ensures that subset types T_A considered do not contain a pair of adjacent types in it while the latter constraint ensures that no two vertices in a clique type are assigned the same color. Thus f is a proper coloring.

We now show that there is no bi-colored path of length 3. Suppose that there exists a path $u_1 - u_2 - u_3 - u_4$ on four vertices such that $f(u_1) = f(u_3)$ and $f(u_2) = f(u_4)$.

- **Vertices u_1, u_2, u_3, u_4 belong to four distinct types.**
  WLOG, let u_1, u_2, u_3, u_4 belong to V_1, V_2, V_3, V_4 respectively. From the definition of neighborhood diversity, we have $V_1, V_3 \in adj(V_2)$ and $V_4 \in adj(V_3)$. As $f(u_1) = f(u_3)$ and $f(u_2) = f(u_4)$, there exists two sets $A \subseteq [t]$ and $B \subseteq [t]$ such that $V_1, V_3 \in T_A$, $V_2, V_4 \in T_B$, $n_A \geq 1$ and $n_B \geq 1$. This cannot happen because of the constraint (C4).
- **Vertices u_1, u_2, u_3, u_4 belong to three distinct types.**
  WLOG, let u_1, u_2, u_3, u_4 belong to V_1, V_2, V_1, V_3 respectively. Since $f(u_1) = f(u_3)$, it is the case that $\sum\limits_{A: V_1 \in T_A} n_A < |V_1|$ implying V_1 is an independent type. Since V_1 is an independent type with two vertices assigned the same color and $f(u_2) = f(u_4)$, there exists $B \subseteq [t]$ such that $V_2, V_3 \in T_B$ and $n_B \geq 1$. This cannot happen because of the constraint (C5).
- **Vertices u_1, u_2, u_3, u_4 belong to two distinct types.**
  WLOG, let $u_1, u_3 \in V_1$ and $u_2, u_4 \in V_2$. Similar arguments as in the above case can be applied to show that V_1 and V_2 are independent types and this case cannot arise due to constraint (C6).

Thus f is a star coloring of G using at most k colors. □

Lemma 2. *If G has a star coloring using at most k colors then there exists a feasible assignment to ILP.*

Proof. Let $f : V(G) \to [k]$ be a star coloring of G using k colors. For each $A \subseteq [t]$, we set

$$n_A = |\bigcap_{V_i \in T_A} f(V_i) - \bigcup_{V_i \notin T_A} f(V_i)|.$$

That is n_A is the number of colors that appear in each of the types in T_A and does not appear in any of the types from $\{V_1, \ldots, V_t\} \setminus T_A$. We now show that such an assignment satisfies the constraints (C0)–(C7).

1. Since f is a proper coloring of G, no two vertices in two adjacent types are assigned the same color. Hence the constraint (C0) is satisfied.
2. Using the fact that f is a star coloring that uses k colors and from the definition of n_A, where each color is counted towards only exactly one variable, we see that the constraint (C1) is satisfied. For each of the remaining variables n_A for which no color is associated with it, we have that $n_A = 0$. Hence the constraint (C7) is satisfied.
3. When V_i is a clique type, we have that $|f(V_i)| = |V_i|$. The expression $\sum\limits_{A: V_i \in T_A} n_A$ denotes the number of colors used in V_i in the coloring f, which equals $|V_i|$. Hence the constraint (C2) is satisfied.
4. When V_i is an independent type, the number of colors used in V_i is at most the minimum of k and $|V_i|$. In addition, we need at least one color to color the vertices of V_i. Hence $1 \leq |f(V_i)| \leq \min\{k, |V_i|\}$. Since $\sum\limits_{A: V_i \in T_A} n_A = |f(V_i)|$, the constraint (C3) is satisfied.

5. Since f is a star coloring, there is no bi-colored P_4. Thus for every combination of four types, say V_1, V_2, V_3 and V_4, if there exists a color assigned to a vertex in V_1 and a vertex in V_3 with $V_1, V_3 \in adj(V_2)$ and $V_4 \in adj(V_3)$, then all the vertices in $V_2 \cup V_4$ should be assigned distinct colors. That is, there is no $B \subseteq [t]$ for which $V_2, V_4 \in T_B$ and $n_B \geq 1$. Hence the constraint (C4) is satisfied.

Similarly, we can show that constraints (C5) and (C6) are also satisfied. □

The running time of the algorithm depends on the time taken to construct an ILP instance and obtain a feasible assignment for the ILP using Theorem 1. The former takes polynomial time while the latter takes $O(q^{2.5q+o(q)}n)$ time where $q = 2^t$ is the number of variables. This completes the proof of Theorem 2.

4 Twin Cover

In this section, we show that Star Coloring is FPT when parameterized by twin cover. Ganian [13] introduced the notion of twin-cover which is a generalization of vertex cover. Note that the parameters neighborhood diversity and twin-cover are not comparable (see Sect. 3.4 in [13]). We now define the parameter twin-cover and state some of its properties.

Definition 2 (Twin Cover [13]). *Two vertices u and v of a graph G are said to be twins if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$ and* true twins *if $N[u] = N[v]$. A twin-cover of a graph G is a set $X \subseteq V(G)$ of vertices such that for every edge $uv \in E(G)$ either $u \in X$ or $v \in X$, or u and v are true twins.*

Remark 1. If $X \subseteq V(G)$ is a twin-cover of G then (i) $G - X$ is disjoint union of cliques, and (ii) for each clique K in $G - X$ and each pair of vertices u, v in K, $N(u) \cap X = N(v) \cap X$.

Theorem 3. Star Coloring *can be solved in $O(q^{2.5q+o(q)}n)$ time where $q = 2^{2^t}$ and t is the size of a twin-cover of the graph.*

Overview of the Algorithm: Given an instance (G, k, t) of Star Coloring, and a twin cover $X \subseteq V(G)$ of size t in G, the goal is to check if there exists a star coloring of G using at most k colors. The algorithm consists of the following four steps.

1. We guess the coloring $f : X \rightarrow [t']$ of X in a star coloring of G (where $t' \leq t$). Then construct an auxiliary graph G' from G where the neighborhood diversity of G' is bounded by a function of t.
2. We show that G has a star coloring g, using at most k colors, such that $g|_X = f$ if and only if G' has a star coloring h, using at most k colors, such that $h|_X = f$.
3. We construct a graph $\mathcal{B}$, which is a subgraph of G' such that G' has a star coloring h, using at most k colors, such that $h|_X = f$ if and only if $\mathcal{B}$ has a proper coloring using at most $k - t'$ colors, where $t' = |f(X)|$.

4. We show that the neighborhood diversity of $\mathcal{B}$ is bounded by a function of t. Then we use the FPT algorithm parameterized by neighborhood diversity from [13] to check whether $\mathcal{B}$ has a proper coloring using at most $k - t'$ colors and decide if there exists a star coloring of G' using at most k colors.

Given a graph G, there exists an algorithm to compute a twin-cover of size at most t (if one exists) in $O(1.2738^t + tn)$ time [13]. Hence we assume that we are given a twin-cover $X = \{v_1, v_2, \dots, v_t\} \subseteq V(G)$ of size t.

Let (G, k, t) be an instance of Star Coloring and $X = \{v_1, v_2, \dots, v_t\} \subseteq V(G)$ be a twin-cover of size t in G. That is, $G[V \setminus X]$ is a disjoint union of cliques. By the definition of twin cover, all vertices in a clique K from $G[V \setminus X]$ has the same neighborhood in X. Similar to the proof of Theorem 2, we define subset types. For each $A \subseteq [t]$, let $T_A = \{v_i \mid i \in A\} \subseteq X$ denote a subset type of G. For every subset type T_A, we denote a *clique type* of G by $K_A = \{K \mid K$ is a clique in $G[V \setminus X]$ and $N(K) \cap X = T_A\}$.

Step 1 of the algorithm is to initially guess the colors of the vertices in X in a star coloring of G. Let $f : X \to [t']$ be such a coloring, where $t' \leq t$. The rest of the proof is to check if f could be extended to a coloring $g : V(G) \to [k]$ such that $g|_X = f$. Let $X_i = f^{-1}(i) \subseteq X$ be the set of vertices from X that are assigned the color $i \in [t']$ in f. We now construct an auxiliary graph G' from G by repeated application of the Claims 1, 2 and the Reduction Rule 1.

Claim 1. *Let K_A be a clique type with $|K_A| \geq 2$ and there exist two vertices in $X_i \cap T_A$ for some $i \in [t']$. Let $G^\star$ be the graph obtained from G by adding additional edges between every pair of non-adjacent vertices in $\bigcup\limits_{K \in K_A} V(K)$. Then (G, k, t) is a yes-instance of* Star Coloring *if and only if $(G^\star, k, t)$ is a yes-instance of* Star Coloring.

Proof. Let $K, K' \in K_A$ be two cliques and $u, v \in X_i \cap T_A$ (i.e., $f(u) = f(v) = i$). For the forward direction, let g be a star coloring of (G, k, t). Since $g|_X = f$ and g is a star coloring, no two vertices in $\bigcup\limits_{K \in K_A} V(K)$ are assigned the same color. Suppose not, there exists two vertices $w, w' \in \bigcup\limits_{K \in K_A} V(K)$ such that $g(w) = g(w')$, then $w - u - w' - v$ is a bi-colored P_4. Observe that g is also a star coloring of $(G^\star, k, t)$.

For the reverse direction, let h be a star coloring of $(G^\star, k, t)$ that uses at most k colors. Since G is a subgraph of $G^\star$, we have that h is also a star coloring of (G, k, t). □

Notice that a clique type K_A satisfying the assumptions of Claim 1 will now have $|K_A| = 1$. We now look at the clique types K_A such that $|K_A| \geq 2$ and apply the following reduction rule. Let $K \in K_A$ be an arbitrarily chosen clique with maximum number of vertices.

Reduction Rule 1. *Let K_A be a clique type with $|K_A| \geq 2$ and $|X_i \cap T_A| \leq 1$, for all $i \in [t']$. Also, let $K \in K_A$ be an arbitrarily chosen clique with maximum* *number of vertices over all cliques in K_A. Then (G, k, t) is a yes-instance of* Star Coloring *if and only if $(G - \bigcup_{K' \in K_A \setminus \{K\}} V(K'), k, t)$ is a yes-instance of* Star Coloring.

Lemma 3 (⋆). *Reduction Rule 1 is safe.*

We repeatedly apply Reduction Rule 1 on the clique types K_A for which $|K_A| \geq 2$ after the application of Claim 1. Thereby ensuring $|K_A| = 1$ for all clique types K_A for which $|K_A| \geq 2$ in G. Thus for all clique types K_A, we have that $|K_A| \leq 1$. Notice that after the application of Claim 1 and Reduction Rule 1, the resulting graph has bounded neighborhood diversity. However, a proper coloring of the resulting graph may not yield a star coloring. The following claim help us to reduce our problem to proper coloring parameterized by neighborhood diversity.

Claim 2. *Let K_A and K_B, with $A \neq B$, be two clique types such that there exists two vertices $u, v \in X_i$ such that $u \in T_A \cap T_B$ and $v \in T_B$, for some $i \in [t]$. Let $G^\star$ be the graph obtained from G by adding additional edges between every pair of non-adjacent vertices in $V(K_A) \cup V(K_B)$. Then (G, k, t) is a yes-instance of* Star Coloring *if and only if $(G^\star, k, t)$ is a yes-instance of* Star Coloring.

Proof. For the forward direction, let g be a star coloring of (G, k, t). This implies that no two vertices in $V(K_A) \cup V(K_B)$ are assigned the same color because of the vertices u and v. Hence g is also a star coloring of $(G^\star, k, t)$.

The reverse direction is trivial. Since G is a subgraph of $G^\star$, the star coloring of $(G^\star, k, t)$ is also a star coloring of (G, k, t). □

We are now ready to explain the steps of our algorithm in detail.

Step 1: Given an instance (G, k, t) of Star Coloring, we construct an auxiliary graph. The graph constructed after repeated application of Claim 1, Reduction Rule 1 and Claim 2, is the auxiliary graph G'. We now argue that the neighborhood diversity of G' is bounded by a function of t. Consider the partition $\{V(K_A) \mid A \subseteq [t]\} \cup \{\{v_i\} \mid v_i \in X\}$ of $V(G')$. Notice that each clique type K_A of G' is a clique type (see Sect. 3 for more details). That is, all the vertices in K_A have the same neighborhood in X. This is true because initially all vertices in K_A have the same neighborhood (by definition of twin cover) and during the process of adding edges (Claims 1 and 2), either all the vertices in K_A are made adjacent to all the vertices in a type K_B ($A \neq B$) or none of them are adjacent to any vertex in K_B. Thus the number of such types is at most 2^t. Including the vertices of X, we have that the neighborhood diversity of G' is at most $2^t + t$.

Step 2: We need to show that (G, k, t) is a yes-instance of Star Coloring if and only if (G', k, t) is a yes-instance of Star Coloring. This is accomplished by the correctness of the Claims 1, 2 and the Reduction rule 1.

Step 3: The next step of the algorithm is to find a set of colors from $[t']$ that can be assigned to the vertices in $V \setminus X$. Towards this, for each $A \subseteq [t]$, we guess a subset of colors $D_A \subseteq [t']$ of size at most $|V(K_A)|$, that can be assigned to the vertices in the clique type K_A in a star coloring of G (extending the coloring f of X) that uses at most k colors. For the guess D_A, we arbitrarily (as it does not matter which vertices are assigned a specific color) assign colors from D_A to vertices in K_A such that $|D_A|$ vertices in K_A are colored distinctly. Given the guess D_A for each K_A, we can check in $2^{O(t)}$ time if the color set D_A associated with K_A is indeed a proper coloring (considering the coloring f of X and its neighboring types). In a valid guess, some vertices $Q \subseteq V(G') \setminus X$ are assigned colors from $[t']$. The uncolored vertices of G' should be given a color from $[k] \setminus [t']$. Let $g : X \cup Q \rightarrow [t']$ be a coloring such that $g(v) = f(v)$ if $v \in X$, and $g(v) = \ell$, where ℓ is the assigned color as per the above greedy assignment, if $v \in Q$

We now extend this partial coloring g of $Q \cup X$ to a full coloring of G', where the vertices in $V(G') \setminus (Q \cup X)$ are assigned colors from $[k] \setminus [t']$. Let $\mathcal{B}$ be the subgraph of G' obtained by deleting the vertices $Q \cup X$ from G'. Notice that $\mathcal{B}$ has neighborhood diversity at most 2^t.

Claim 3. *There exists a star coloring of G' extending g using at most k colors if and only if there exists a proper coloring of $\mathcal{B}$ using at most $k - t'$ colors.*

Proof. Let $h : V(G') \rightarrow [k]$ be a star coloring of G' such that $h|_{Q \cup X} = g$. Clearly $|h(\mathcal{B})| = |h(V \setminus (Q \cup X))| \leq k - t'$. That is, h restricted to the vertices of $\mathcal{B}$ is a proper coloring of $\mathcal{B}$ which uses at most $k - t'$ colors.

For the reverse direction, let $c : V(\mathcal{B}) \rightarrow [k - t']$ be a proper coloring. We construct a coloring $h : V(G') \rightarrow [k]$ using the coloring c as follows: $h(v) = c(v)$ if $v \in \mathcal{B}$, and $h(v) = g(v)$ otherwise. We show that h is a star coloring of G'. Suppose not, without loss of generality, let $u_1 - u_2 - u_3 - u_4$ be a bi-colored P_4, with $u_1 \in K_{A_1}$, $u_2 \in X$, $u_3 \in K_{A_2}$ and $u_4 \in X$ (notice that this is the only way a bi-colored P_4 exists), for some $A_1, A_2 \subseteq [t]$. That is, $c(u_1) = c(u_3)$ and $c(u_2) = c(u_4)$. Also, $A_1 \neq A_2$ because of the proper coloring. If this were the case, we would have applied Claim 2 as K_{A_1} and K_{A_2} satisfy the assumptions along with coloring of the vertices u_2 and u_4 in X. As a consequence, each vertex in K_{A_1} would have been adjacent to each vertex in K_{A_2}. □

Step 4: It is known that proper coloring is FPT parameterized by neighborhood diversity [12]. The algorithm in [12] uses integer linear programming with 2^{2k} variables, where k is the neighborhood diversity of the graph. Since $\mathcal{B}$ has neighborhood diversity at most 2^t, we have that the number of variables $q \leq 2^{2^t}$. We use the algorithm to test whether $\mathcal{B}$ has a proper coloring using at most $k - t'$ colors.

Running Time: Step 1 of the algorithm takes $O(t^t)$ time to guess a coloring of X. Reduction Rule 1, Claims 1 and 2 can be applied in $2^{O(t)}n^{O(1)}$ time. Step 2 can be processed in $2^{O(t)}n^{O(1)}$ time. Step 3 involves guessing the colors that the clique types can take from the colors used in X and this takes $O(2^{2^t})$ time. Constructing $\mathcal{B}$ takes polynomial time. Step 4 is applying the FPT algorithm parameterized by neighborhood diversity from [12] on $\mathcal{B}$ which takes $O(q^{2.5q+o(q)}n)$ time where

$q \leq 2^{2^t}$. The latter dominates the running time and hence the running time of the algorithm is $O(2^{2^t} q^{2.5q+o(q)} n^{O(1)})$, where $q \leq 2^{2^t}$.

This completes the proof of Theorem 3.

5 Clique-Width

Theorem 4 ($\star$). *Given a graph G, its nice w-expression and an integer k, we can decide if there exists a star coloring of G using k colors in $O((3^{w^3k^2+w^2k^2})^2 n^{O(1)})$ time.*

6 Conclusion

In this paper, we study the parameterized complexity of Star Coloring with respect to several structural graph parameters. We show that Star Coloring is FPT when parameterized by (a) neighborhood diversity, (b) twin cover, and (c) the combined parameter clique-width and the number of colors.

We conclude the paper with the following open problems for further research.

1. What is the parameterized complexity of Star Coloring when parameterized by distance to cluster or distance to co-cluster?
2. It is known that graph coloring admits a polynomial kernel when parameterized by distance to clique [16]. Does Star Coloring also admit a polynomial kernel parameterized by distance to clique?

Acknowledgments. We would like to thank anonymous referees for their helpful comments. The first author and the second author acknowledges SERB-DST for supporting this research via grants PDF/2021/003452 and SRG/2020/001162 respectively for funding to support this research.

References

1. Albertson, M., Chappell, G., Kierstead, H., Kündgen, A., Ramamurthi, R.: Coloring with no 2-colored P_4's. Electr. J. Comb. **11**, 03 (2004)
2. Bhyravarapu, S., Reddy, I.V.: On structural parameterizations of star coloring. arXiv preprint arXiv:2211.12226 (2022)
3. Büskens, C., Maurer, M.: Sensitivity analysis and real-time optimization of parametric nonlinear programming problems. In: Grotschel, M., Krumke, S.O., Rambau, J. (eds.) Online Optimization of Large Scale Systems, pp. 3–16. Springer, Heidelberg (2001)
4. Coleman, T.F., Moré, J.J.: Estimation of sparse Jacobian matrices and graph coloring blems. SIAM J. Numer. Anal. **20**(1), 187–209 (1983)
5. Coleman, T.F., Moré, J.J.: Estimation of sparse hessian matrices and graph coloring problems. Math. Program. **28**(3), 243–270 (1984). https://doi.org/10.1007/BF02612334
6. Courcelle, B.: The monadic second-order logic of graphs III: tree-decompositions, minors and complexity issues. RAIRO-Theor. Inf. Appl. **26**(3), 257–286 (1992)

7. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory Comput. Syst. **33**(2), 125–150 (2000). https://doi.org/10.1007/s002249910009
8. Courcelle, B., Olariu, S.: Upper bounds to the clique width of graphs. Discret. Appl. Math. **101**(1–3), 77–114 (2000)
9. Cygan, M., et al.: Parameterized Algorithms, vol. 4. Springer, Cham (2015)
10. Fomin, F.V., Golovach, P.A., Lokshtanov, D., Saurabh, S.: Intractability of clique-width parameterizations. SIAM J. Comput. **39**(5), 1941–1956 (2010)
11. Frank, A., Tardos, É.: An application of simultaneous diophantine approximation in combinatorial optimization. Combinatorica **7**(1), 49–65 (1987). https://doi.org/10.1007/BF02579200
12. Ganian, R.: Using neighborhood diversity to solve hard problems. CoRR, abs/1201.3091 (2012)
13. Ganian, R.: Improving vertex cover as a graph parameter. Discrete Math. Theor. Comput. Sci. **17**(2), 77–100 (2015)
14. Gebremedhin, A.H., Tarafdar, A., Pothen, A., Walther, A.: Efficient computation of sparse hessians using coloring and automatic differentiation. INFORMS J. Comput. **21**(2), 209–223 (2009)
15. Grünbaum, B.: Acyclic colorings of planar graphs. Israel J. Math. **14**(4), 390–408 (1973)
16. Gutin, G., Majumdar, D., Ordyniak, S., Wahlström, M.: Parameterized pre-coloring extension and list coloring problems. SIAM J. Discrete Math. **35**(1), 575–596 (2021)
17. Harshita, K., Mishra, S., Sadagopan, N.: FO and MSO approach to some graph problems: approximation and poly time results. arXiv preprint arXiv:1711.02889 (2017)
18. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)
19. Lampis, M.: Algorithmic meta-theorems for restrictions of treewidth. Algorithmica **64**(1), 19–37 (2012)
20. Lei, H., Shi, Y., Song, Z.-X.: Star chromatic index of subcubic multigraphs. J. Graph Theory **88**(4), 566–576 (2018)
21. Lenstra, H.W.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4), 538–548 (1983)
22. Lorenz, C., Carlsen, I.-C., Buzug, T.M., Fassnacht, C., Weese, J.: A multi-scale line filter with automatic scale selection based on the Hessian matrix for medical image segmentation. In: ter Haar Romeny, B., Florack, L., Koenderink, J., Viergever, M. (eds.) Scale-Space 1997. LNCS, vol. 1252, pp. 152–163. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63167-4_47
23. Lyons, A.: Acyclic and star colorings of cographs. Discrete Appl. Math. **159**(16), 1842–1850 (2011)
24. Omoomi, B., Roshanbin, E., Dastjerdi, M.V.: A polynomial time algorithm to find the star chromatic index of trees. Electron. J. Comb. **28**(1), 1 (2021)
25. Robertson, N., Seymour, P.D.: Graph minors. I. excluding a forest. J. Comb. Theory Ser. B **35**(1), 39–61 (1983)
26. Shalu, M.A., Antony, C.: The complexity of star colouring in bounded degree graphs and regular graphs. In: Balachandran, N., Inkulu, R. (eds.) CALDAM 2022. LNCS, vol. 13179, pp. 78–90. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-95018-7_7