Dynamic Coloring on Restricted Graph Classes

Sriram Bhyravarapu[1](✉), Swati Kumari[2], 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, Raipur, India
{swatik,vinod}@iitbhilai.ac.in

Abstract. A proper k-coloring of a graph is an assignment of colors from the set $\{1, 2, \ldots, k\}$ to the vertices of the graph such that no two adjacent vertices receive the same color. Given a graph $G = (V, E)$, the Dynamic Coloring problem asks to find a proper k-coloring of G such that for every vertex $v \in V(G)$ of degree at least two, there exists at least two distinct colors appearing in the neighborhood of v. The minimum integer k such that there is a dynamic coloring of G using k colors is called the dynamic chromatic number of G and is denoted by $\chi_d(G)$.

The problem is NP-complete in general, but solvable in polynomial time on several restricted families of graphs. In this paper, we study the problem on restricted classes of graphs. We show that the problem can be solved in polynomial time on chordal graphs and biconvex bipartite graphs. On the other hand, we show that it is NP-complete on star-convex bipartite graphs, comb-convex bipartite graphs and perfect elimination bipartite graphs. Next, we initiate the study on Dynamic Coloring from the parameterized complexity perspective. First, we show that the problem is fixed-parameter tractable when parameterized by neighborhood diversity or twin-cover. Then, we show that the problem is fixed-parameter tractable when parameterized by the combined parameters clique-width and the number of colors.

Keywords: proper coloring · fixed-parameter tractable · dynamic coloring · neighborhood diversity · twin-cover · bipartite graphs

1 Introduction

A *vertex coloring* (or proper coloring) of a graph G is an assignment of colors to the vertices of the graph such that no two adjacent vertices are assigned the same color. The minimum number of colors required for a proper coloring of G is called the chromatic number of G denoted by $\chi(G)$. Given a graph $G = (V, E)$ and an integer $k \in \mathbb{N}$, a proper k-coloring $f : V(G) \to [k]$ is called a *dynamic coloring*, if for every vertex $v \in V(G)$ of degree at least 2, there are at least two distinct colors appearing in the neighborhood of v, i.e., $|f(N(v))| \geq 2$, where the

M. Mavronicolas (Ed.): CIAC 2023, LNCS 13898, pp. 112–126, 2023.
https://doi.org/10.1007/978-3-031-30448-4_9

set $N(v)$ denotes the neighbors of v in G. The smallest integer k such that there is a dynamic coloring of G using k colors is called the *dynamic chromatic number* of G and is denoted by $\chi_d(G)$. Note that for any graph G, $\chi_d(G) \geq \chi(G)$.

Dynamic coloring was introduced by Montgomery [14] and it is NP-complete on general graphs [13]. The problem has been studied on various restricted graph classes. For example, polynomial time algorithms are obtained for trees [14] and graphs with bounded tree-width [13] while it is NP-hard even for planar bipartite graphs with maximum degree at most three and arbitrarily high girth [16]. Finding upper bounds of $\chi_d(G)$ for planar graphs have been studied in several papers. It was shown in [3] that $\chi_d(G) \leq 5$ if G is a planar graph. Later in [10] it was shown that if G is a connected planar graph with $G \neq C_5$, then $\chi_d(G) \leq 4$. Dynamic coloring of graphs has been studied extensively by several authors, see for instance [1,3,8,14,16].

In this paper, we study the decision version of the Dynamic Coloring problem, which is stated as follows.

Dynamic Coloring
Input: A graph $G = (V, E)$ and a positive integer k.
Question: Does G have a dynamic coloring using at most k colors?

In the first part of this paper, we study the computational complexity of Dynamic Coloring on restricted families of graphs. We show that the problem can be solved in polynomial time on chordal graphs. As the problem is NP-complete on bipartite graphs [13], we study its complexity on sub-classes of bipartite graphs and close the gap between classes of graphs that are NP-complete and P time solvable. It is known that $\chi_d(G)$ is unbounded [8] when G is a bipartite graph. We show that $\chi_d(G) \leq 4$, when G is a biconvex bipartite graph and give a polynomial time algorithm by exploiting its structural properties. We also show that the problem is NP-complete on several sub-classes of bipartite graphs, a hierarchy of which is illustrated in Fig. 1.

In the second part of this paper, we study Dynamic Coloring from the viewpoint of parameterized complexity [4,5]. In parameterized complexity, each problem instance is associated with an integer, say k, called parameter. A parameterized problem is said to be fixed-parameter tractable (FPT) with respect to a parameter k if it can be solved in time $f(k)n^{O(1)}$, where n is the input size and f is a computable function only depending on the parameter k.

There may be many parameterizations for Dynamic Coloring. The most natural parameter to consider is the "solution size", which in this case is the number of colors. As the problem is NP-complete [13] even when the number of colors is three, we do not expect to have an FPT algorithm with solution size as the parameter. There are parameters which are selected based on the structure of the graph, called "structural parameterizations". For instance, vertex cover, tree-width, neighborhood diversity, etc. The hierarchy of a few structural graph parameters is illustrated in Fig. 2.

We study the parameterized complexity of Dynamic Coloring with respect to several structural graph parameters. Tree-width is one of the most used

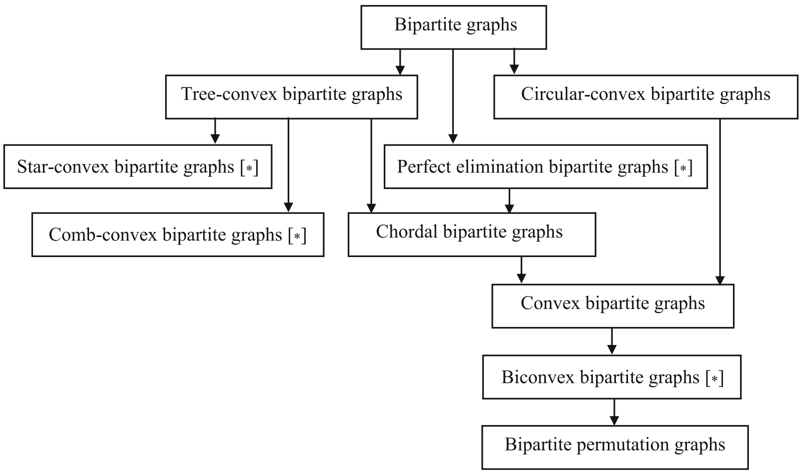
Bipartite graphs
Tree-convex bipartite graphs
Circular-convex bipartite graphs
Star-convex bipartite graphs [*]
Perfect elimination bipartite graphs [*]
Comb-convex bipartite graphs [*]
Chordal bipartite graphs
Convex bipartite graphs
Biconvex bipartite graphs [*]
Bipartite permutation graphs

Fig. 1. Hierarchy showing the relationship between sub-classes of bipartite graphs. The graph classes considered in this paper are indicated by $*$.

structural parameters when dealing with NP-hard graph problems. Li et al. in [16] showed that Dynamic Coloring is FPT when parameterized by tree-width, following its formulation in monadic second order logic. One disadvantage with tree-width is that it is unbounded for dense graphs (e.g., cluster graphs). In this paper, we consider the parameters twin-cover and neighborhood diversity, which also include dense graphs. We show that Dynamic Coloring is fixed-parameter tractable when parameterized by twin-cover or neighborhood diversity.

Next, we consider the graph parameter clique-width, which is suitable when dealing with hard graph problems on dense graphs. Clique-width is a generalization of the parameters twin-cover and neighborhood diversity, in the sense that graphs of bounded neighborhood diversity or graphs of bounded twin-cover also have bounded clique-width. We show that Dynamic Coloring is FPT when parameterized by the combined parameters clique-width and the number of colors. Hence studying the parameterized complexity of Dynamic Coloring with respect to the above mentioned parameters reduces the gap between tractability and intractability. We summarize our contribution below.

1. In Sect. 3, we show that Dynamic Coloring can be solved in polynomial time on chordal graphs.
2. In Sect. 4, we show that Dynamic Coloring is polynomial time solvable on bipartite permutation graphs, a sub-class of biconvex graphs. We extend this algorithm to design a polynomial time algorithm for biconvex graphs, in Sect. 5.
3. In Sect. 6, we show NP-completeness results on star-convex bipartite graphs, comb-convex bipartite graphs and perfect elimination bipartite graphs

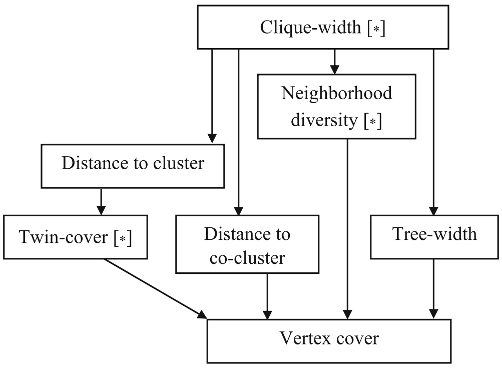

Fig. 2. Hasse diagram of a few structural graph parameters. An edge from a parameter k_1 to a parameter k_2 means that there is a function f such that for all graphs G, we have $k_1(G) \leq f(k_2(G))$. The parameters considered in this paper are indicated by $*$.

strengthening the NP-completeness result of Dynamic Coloring on bipartite graphs.

4. In Sects. 7 and 8, we show that Dynamic Coloring is FPT when parameterized by neighborhood diversity, twin-cover or the combined parameters clique-width and the number of colors.

2 Preliminaries

For $k \in \mathbb{N}$, we use $[k]$ to denote the set $\{1, 2, \ldots, k\}$. 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 a graph. An edge between vertices x and y is denoted as xy for simplicity. For a subset $X \subseteq V(G)$, the graph $G[X]$ denotes the subgraph of G induced by the vertices of X.

For a vertex set $X \subseteq V(G)$, we denote by $G \setminus X$, the graph obtained from G by deleting all vertices of X and their incident edges. For a vertex $v \in V(G)$, by $N(v)$, we denote the set $\{u \in V(G) \mid vu \in E(G)\}$ and we use $N[v]$ to denote the set $N(v) \cup \{v\}$. The neighborhood of a vertex set $S \subseteq V(G)$ is $N(S) = (\cup_{v \in V(G)} N(v)) \setminus S$. A graph is *bipartite* if its vertex set can be partitioned into two disjoint sets such that no two vertices in the same set are adjacent. We say a vertex v of degree at least two to have satisfied *dynamic coloring property* if there exist at least two vertices in the neighborhood of v that are colored distinctly.

Due to space constraints, the proofs of the Theorems marked ($\star$) are presented in the full version of the paper.

3 Chordal Graphs

A vertex v of a graph G is called a *simplicial* vertex if the subgraph of G induced by the vertex set $\{v\} \cup N(v)$ is a complete graph. A *perfect elimination ordering* is a vertex ordering $v_1, \ldots, v_n$ of $V(G)$ such that each v_i is simplicial in $G[i]$, the subgraph induced by the vertices $\{v_1, \ldots, v_i\}$. A graph G is *chordal* if and only if it has a perfect elimination ordering. Given a graph, a perfect elimination ordering of G can be done in polynomial time [15]. In this section, we show Dynamic Coloring is polynomial time solvable on chordal graphs.

Theorem 1. *Let G be a chordal graph with at least two edges, then*

$$\chi_d(G) = \begin{cases} 3 & \text{if } \omega(G) = 2 \\ \omega(G) & \text{if } \omega(G) \geq 3 \end{cases}$$

where $\omega(G)$ is the size of a largest clique of G.

Proof. Let $G = (V, E)$ be a chordal graph. Hence G is C_k-free, for $k \geq 4$. If $\omega(G) = 2$, then G is C_k-free for $k \geq 3$. That is, G is a tree, hence from [14] we know $\chi_d(G) = 3$. We now deal with the case when $\omega(G) \geq 3$.

Let $v_1, \ldots, v_n$ be a perfect elimination ordering (PEO) of G. That is, each v_i is simplicial in $G[i]$. Let $f : V(G) \to [\omega(G)]$ be a coloring of G defined as follows: assign $f(v_1) = 1$ and $f(v_2) = 2$. For each v_i, where $i \geq 3$, let $D_i = N(v_i) \cap \{v_1, \ldots, v_{i-1}\}$ be the neighbors of v_i in $G[i]$ and T_i be the set of colors used to color the vertices of D_i. Then,

$$f(v_i) = \begin{cases} \min\{[\omega(G)] \setminus \{f(v_j), f(v_k)\}\} & \text{if } D_i = \{v_j\} \text{ and } D_j = \{v_k\}, \text{ where } k < j < i \\ \min\{[\omega(G)] \setminus T_i\} & \text{otherwise} \end{cases}$$

Clearly, f is a proper coloring as each v_i is greedily assigned a color (minimum) not appearing in T_i. Next, we show that f is dynamic coloring. For a vertex v_i, if $|D_i| \geq 2$ then $|T_i| \geq 2$ (i.e., $|f(N(v_i))| \geq 2$). If $|D_i| = 1$, and let $D_i = \{v_j\}$, then v_i has only one neighbor v_j in $G[i]$.

If v_i has no neighbor in the set $\{v_{i+1}, \ldots, v_n\}$ then the degree of v_i in G is one. Suppose, v_i has a neighbor in the set $\{v_{i+1}, \ldots, v_n\}$. Let v_p be the first neighbor (according to PEO) of v_i in the set $\{v_{i+1}, \ldots, v_n\}$. If $|D_p| = 1$, then by case 1, $f(v_p) \neq f(v_j)$, $f(v_p) \neq f(v_i)$ and $f(v_i) \neq f(v_j)$, hence v_i has at least two neighbors with distinct colors. Else if $|D_p| = 2$, then $D_p = \{v_j, v_i\}$, again by case 2, the three vertices v_j, v_i and v_p are colored with three distinct colors. Hence v_i has at two neighbors with distinct colors. Note that $|D_p|$ cannot be greater than 2 because of the choice of p.

It is easy to see that $\chi_d(G) \geq \omega(G)$ as we need at least $\omega(G)$ colors to properly color the largest clique of G. Since any vertex v_i has $\ell = |D_i|$ many neighbors in $G[i]$, at least one of the colors $1, 2, \cdots, \ell+1$ is not used in T_i. Hence our algorithm finds a coloring of G with at most $\max_i\{|T_i| + 1\}$ colors, which is at most $\omega(G)$.

Hence $\chi_d(G) = \omega(G)$. The time required for the above coloring procedure is $O(n^2)$. $\square$

4 Bipartite Permutation Graphs

In this section, we show that Dynamic Coloring is polynomial time solvable on bipartite permutation graphs. We start the section with some basic definitions and notations that are needed to describe the algorithm.

Definition 1. (Chain Graph [6]). *A bipartite graph $G = (A \cup B)$ is called a chain graph if for every two vertices $u_1, u_2 \in A$ we have either $N(u_1) \subseteq N(u_2)$ or $N(u_2) \subseteq N(u_1)$.*

That is, there is an ordering of the vertices of A, say $u_1, u_2, \dots u_{|A|}$, such that $N(u_i) \subseteq N(u_{i+1})$, $1 \leq i < |A|$. As a consequence, we can also find an ordering of the vertices of B, say $v_1, v_2, \dots v_{|B|}$, such that $N(v_i) \subseteq N(v_{i+1})$, $1 \leq i < |B|$.

We can see that each part of a chain graph can be linearly ordered under the inclusion of their neighborhoods. We say a vertex ordering σ of A is increasing if $x <_\sigma y$ implies $N(x) \subseteq N(y)$, and decreasing if $x <_\sigma y$ implies $N(y) \subseteq N(x)$.

Definition 2. (Multi-chain Ordering [2,6]). *Given a connected graph $G = (V, E)$, we arbitrarily choose a vertex as $v_0 \in V(G)$ and construct distance layers $L_0, L_1, \dots, L_p$ from v_0. The layer L_i, where $i \in [p]$, represents the set of vertices that are at a distance i from v_0 and p is the largest integer such that $L_p \neq \emptyset$.*

We say that these layers form a multi-chain ordering *of G if for every two consecutive layers L_i and L_{i+1}, where $i \in \{0, 1, \dots, p-1\}$, we have that $G[L_i \cup L_{i+1}]$ forms a chain graph.*

We say a graph G *admits multi-chain ordering* if there exists a vertex $v_0 \in V(G)$ such that the distance layers form a multi-chain ordering. An illustration of a multi-chain ordering of a graph is given in Fig. 3. It is known [6] that all connected permutation graphs and interval graphs admit multi-chain ordering. We first observe the following on multi-chain ordering.

Observation 2. *If a graph G admits a multi-chain ordering with $p+1$ layers, then there exists a vertex v in L_i, $i \in [p]$, such that $N(v) \supseteq L_{i+1}$.*

Definition 3. (Bipartite Permutation Graph [2]). *A connected graph $G = (V, E)$ is bipartite permutation if and only if $V(G)$ can be partitioned into $q+1$ disjoint independent sets $L_0, L_1, \dots, L_q$ (in this order) in such a way that*

1. *Any two vertices in non-consecutive sets are non-adjacent.*
2. *Any two consecutive sets L_{i-1} and L_i induce a chain graph, denoted by G_i.*
3. *For each $i \in \{1, 2, \dots, q-1\}$, there is an ordering of vertices of the set L_i such that it is non-increasing in G_i and non-decreasing in G_{i+1}. For the set L_0 (resp. L_q), there is a non-decreasing (resp. non increasing) ordering of vertices of L_0 (resp. L_q) in G_1 (resp. G_q).*

Observation 3. *If c is any dynamic coloring of a bipartite permutation graph that uses exactly three colors then at most two colors are used in any layer L_i, i.e., $|c(L_i)| \leq 2$.*

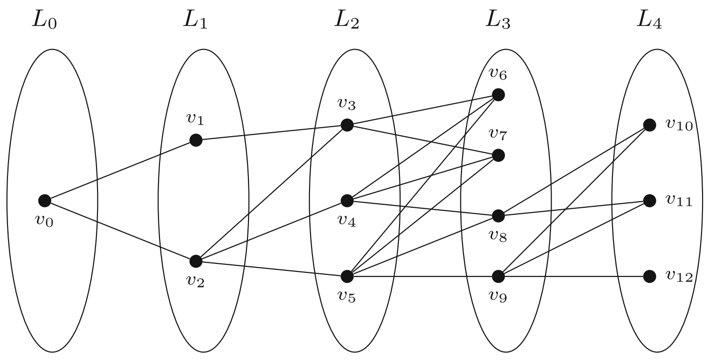

Fig. 3. An illustration of a multi-chain ordering of a graph. The ordering of the vertices in L_2 is $\sigma_2 = v_3, v_4, v_5$. In the chain graph $G[L_1 \cup L_2]$ we have that $N(v_3) \supseteq N(v_4) \supseteq N(v_5)$ and in $G[L_2 \cup L_3]$ we have that $N(v_3) \subseteq N(v_4) \subseteq N(v_5)$.

Proof. Suppose there exists a layer L_i such that $|c(L_i)| = 3$. From Observation 2, there is a vertex $v \in L_{i-1}$ such that $N(v) \supseteq L_i$. Since v is adjacent to all vertices in L_i and $|c(L_i)| = 3$, there exists a color assigned to a vertex in L_i that is same as $c(v)$. This is a contradiction to the fact that c is a proper coloring and thus a dynamic coloring. □

We now show a polynomial time algorithm to decide if the dynamic chromatic number of a bipartite permutation graph is three.

Lemma 1. *Given a bipartite permutation graph G, there is a polynomial algorithm to decide if $\chi_d(G) = 3$.*

Proof. Let $G = (V, E)$ be a bipartite permutation graph and $L_0, L_1, \ldots, L_p$ be the distance layers in a multi-chain ordering of G constructed from an arbitrarily chosen vertex $v_0 \in V(G)$, according to Definition 3. If $\chi_d(G) = 3$, then from Observation 3, we have that in any dynamic coloring $c : V(G) \to \{c_1, c_2, c_3\}$ of G, at most two colors are used for assigning colors in any layer L_i of G. This leaves us with the following cases: either $|c(L_i)| = 1$ or $|c(L_i)| = 2$. At each layer L_i, $0 \le i \le p$, we maintain all possible colorings of L_i in the following manner.

If L_i uses exactly one color, then we have three possible ways of coloring L_i. That is, we have three colorings where each coloring of L_i represents all its vertices being assigned the same color (one of the three colors).

If L_i uses exactly two colors, say c_1 and c_2, then we guess four vertices: the first and the last vertices in the ordering σ_i of L_i that are assigned the colors c_1 and c_2. Using this guess, we extend the coloring to the remaining vertices of L_i as follows. Let x_1^i and y_1^i (resp. x_2^i and y_2^i) be the first and the last vertices in σ_i which are colored with c_1 (resp. c_2). Let $w \in L_i \setminus \{x_1^i, y_1^i, x_2^i, y_2^i\}$. From the above description, we have that either $x_1^i < w < y_1^i$ or $x_2^i < w < y_2^i$. If $x_1^i < w < y_1^i$ then we color w with c_1 else we color it with c_2.

Let $c : V(G) \to \{c_1, c_2, c_3\}$ be a dynamic coloring of G. Let $c(L_i) = \{c_1, c_2\}$ and $\{x_1^i, y_1^i, x_2^i, y_2^i\}$ be the first and the last vertices in σ_i that are assigned the colors c_1 and c_2. Then the coloring obtained by applying the above extension procedure, on each layer L_i, on vertices $\{x_1^i, y_1^i, x_2^i, y_2^i\}$ is also a dynamic coloring of G. Hence it is enough to know the first and the last vertices in the ordering σ_i of L_i that are assigned the colors c_1 and c_2.

The number of colorings for a pair of colors c_1 and c_2 is at most $|L_i|^4$. Since there are three such pairs, the total number of colorings arising out of L_i is at most $3|L_i|^4$. Using these local colorings at the levels, we check for the existence of a dynamic coloring of G using a dynamic programming routine.

Let C_i be the set of colorings of L_i obtained from the above description. We call a triplet coloring (c_{i-1}, c_i, c_{i+1}), where $c_{i-1} \in C_{i-1}$, $c_i \in C_i$ and $c_{i+1} \in C_{i+1}$, as a *feasible coloring* (or simply *feasible*) for L_i, where $1 \leq i \leq p-1$, if every vertex of L_i admits dynamic coloring property when the colorings c_{i-1}, c_i, and c_{i+1} are assigned to L_{i-1}, L_i and L_{i+1} respectively. Similarly, we call a pair (c_0, c_1) (resp. (c_{p-1}, c_p)) as feasible for the layer L_0 (resp. L_p). Let T_i denote the set of all feasible colorings corresponding to the layer L_i.

We now use a dynamic programming approach to check if there exists a dynamic coloring of G using three colors. We have an entry $d[i, f_i]$ for each feasible coloring f_i at layer L_i that defines the existence of a coloring in $G[L_0 \cup L_1 \cup \cdots \cup L_{i+1}]$ such that all vertices in $L_0 \cup L_1 \cup \cdots \cup L_i$ satisfy dynamic coloring property given the feasible coloring f_i at L_i.

Let $f_i = (c_{i-1}, c_i, c_{i+1})$. We set the entry $d[i, f_i] = true$ if f_i is feasible and there exists a feasible coloring $f_{i-1} = (x, c_{i-1}, c_i)$ at L_{i-1} such that $d[i-1, f_{i-1}] = true$. Otherwise, we set $d[i, f_i] = false$. We initialize $d[0, f] = true$, for each feasible coloring $f = (c_0, c_1)$ of L_0 if the vertex $v_0 \in L_0$ satisfies dynamic coloring when the colorings c_0 and c_1 are assigned to the layers L_0 and L_1 respectively. Otherwise, we initialize $d[0, f] = false$. If there exists an entry $d[p, f_p]$, for some feasible coloring f_p of L_p, such that $d[p, f_p] = true$, we decide that there exists a dynamic coloring of G using three colors. If $d[p, f_p] = false$ for every feasible coloring f_p of L_p, then we decide that G does not have a dynamic coloring using three colors.

The correctness of the algorithm follows from the description of the algorithm. We now compute the running time of the algorithm which includes guessing the colorings at each layer and applying the dynamic programming routine. Computing the colorings for all the layers takes $O(p \cdot |L_i|^4) \leq O(n^5)$ time. The number of feasible colorings is at most $|L_{i-1}|^4 \cdot |L_{i+1}|^4 \cdot |L_{i+1}|^4 \leq n^{12}$. Computing if a coloring is feasible coloring can be done in $O(n^2)$ time. All the entries $d[i, \cdot]$ pertaining a layer can be computed in $O(n^{14})$ time. Since $i \leq p \leq n$, the total time taken is $O(n^{15})$. □

Theorem 4. Dynamic Coloring *can be solved in polynomial time on bipartite permutation graphs.*

Proof. Let $G = (V, E)$ be a connected bipartite permutation graph. Since bipartite permutation graphs are a sub-class of biconvex graphs, it follows from Lemma 2 that $\chi_d(G) \leq 4$. It is easy to see that (i) $\chi_d(G) = 1$ if and only if $G = K_1$, and (ii) $\chi_d(G) = 2$ if and only if $G = K_2$. If G does not belong to any of the above cases, then $\chi_d(G) \in \{3, 4\}$. We check whether $\chi_d(G) = 3$ using Lemma 1. If $\chi_d(G) \neq 3$, we decide that $\chi_d(G) = 4$. □

5 Biconvex Graphs

Definition 4 (Biconvex Graph). *An ordering σ of X in a bipartite graph $B = (X, Y, E)$ has the adjacency property if for every vertex $y \in Y$, the neighborhood $N(y)$ consists of vertices that are consecutive (an interval) in the ordering σ of X. A bipartite graph (X, Y, E) is biconvex if there are orderings of X (with respect to Y) and Y (with respect to X) that fulfills the adjacency property.*

Theorem 5. Dynamic Coloring *can be solved in polynomial time on biconvex graphs.*

Towards showing Theorem 5, we first show that the dynamic chromatic number of a biconvex graph is at most 4.

Lemma 2. *If G is a biconvex graph then $\chi_d(G) \leq 4$.*

Proof. Let $G = (X, Y, E)$ be a biconvex graph. We assume that G has at least five vertices, otherwise, trivially $\chi_d(G) \leq 4$.

We use the property that G is biconvex. Let $\sigma = x_1, x_2, \ldots, x_p$ be an enumeration of vertices of X and $\pi = y_1, y_2, \ldots, y_q$ be an enumeration of vertices of Y. For each $i \in [p]$, color x_i with 1 if i is odd, else color it with 2. For each $j \in [q]$, color y_j with 3 if i is odd, else color it with 4. Consider any vertex $x_i \in X$ with degree at least two. As G is convex over Y, the vertices adjacent to x_i are consecutive with respect to the ordering π. That is, if y_j is a neighbor of x_i, then at least one of y_{j+1} or y_{j-1} is a neighbor of x_i. Hence the neighborhood of x_i contains a vertex of color 3 and a vertex of color 4. Similarly, we can show that the neighborhood of every vertex $y_j \in Y$ contains a vertex of color 1 and a vertex of color 2. Hence, the above coloring is a dynamic coloring of G. □

We now proceed to the proof of Theorem 5 which is similar to the proof of bipartite permutation graphs in Theorem 4. Hence, we give a short proof highlighting the key differences.

Proof. (Short Proof of Theorem 5). Let G be a biconvex graph. We know that $\chi_d(G) \leq 4$, from Lemma 2. Similar to the proof of bipartite permutation graphs, it is sufficient to check whether $\chi_d(G) \in \{3, 4\}$. Since all connected biconvex graphs admit multi-chain ordering [6], it is possible to extend our algorithm in Theorem 4 to biconvex graphs.

The difference between bipartite permutation graphs and biconvex graphs is that the latter has two vertex orderings in a multi-chain ordering, say $\sigma_{i,1}$ and $\sigma_{i,2}$, for each layer L_i, one corresponding to L_{i-1} and the other corresponding to L_{i+1}. For each of the two orderings, we guess the first and last vertices in the respective ordering that are assigned colors based on how many colors are seen in each layer. However, we need to ensure that the guesses obtained in the orderings should complement each other in the sense that a vertex $v \in L_i$ cannot be assigned the color c_1 in one ordering and the color c_2 in the other ordering. The rest of the algorithm is similar to Theorem 4. The number of colorings at each layer is at most $|L_i|^8$. The number of feasible colorings is at most $|L_{i-1}|^8 \cdot |L_i|^8 \cdot |L_{i+1}|^8 \leq n^{24}$. Considering the number of entries in the dynamic programing table, the total time taken is $O(n^{27})$. $\square$

6 Hardness Results on Sub-classes of Bipartite Graphs

Theorem 6 ($\star$). Dynamic Coloring *is* NP*-complete on perfect elimination bipartite graphs, star-convex bipartite graphs and comb-convex graphs.*

7 Parameterization by Neighborhood Diversity

The graph parameter neighborhood diversity was introduced by Lampis [11], and it is a generalization of the parameter vertex cover. In this section, we show that Dynamic Coloring is fixed-parameter tractable parameterized by neighborhood diversity. Our main idea is to reduce our problem to the integer linear programming problem that is fixed-parameter tractable when parameterized by the number of variables.

Definition 5. (Neighborhood Diversity [11]). *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 the vertices in each set have the same type.*

Observe that each V_i either forms a clique or an independent set in G, for all $i \in [t]$. We call the set V_i as a *clique type* (resp. *independent type*) if $G[V_i]$ is a clique (resp. independent set). If $|V_i| = 1$, then we consider V_i as an independent type. For each $i, j \in [t]$, $i \neq j$, it is the case that either every vertex in V_i is adjacent to every vertex in V_j or no vertex in V_i is adjacent to any vertex in V_j.

For each $A \subseteq \{1, 2, \cdots, t\}$, we denote a *subset type* of G by $T_A = \{V_i : i \in A\}$. We denote the set of types neighboring the type V_i in G by $adj(V_i)$. That is, $V_j \in adj(V_i)$, if every vertex in V_i is adjacent to every vertex in V_j. Given a proper coloring $f : V(G) \to [k]$, we say V_i *admits dynamic coloring* with respect to f if for all $v \in V_i$, $|f(N(v))| \geq 2$.

Given a graph $G = (V, E)$, there exists an algorithm that runs in polynomial time [11] and finds a minimum sized neighborhood partition of $V(G)$. So, we assume that the types $V_1, V_2, \ldots, V_t$ are given as input. If $t = 1$, then G is a complete graph and the problem can be solved easily. Hence, we assume $t \geq 2$.

Observation 7. *If there exists a proper coloring of G and V_i is a clique type, then V_i admits dynamic coloring.*

Proof. Recall that each clique type has at least two vertices. Let V_i be a clique type. Since $t \geq 2$ and G is connected, there exists a type $V_j \in adj(V_i)$ such that every vertex in V_i is part of a triangle (with two vertices from V_i and a vertex from V_j) that uses three distinct colors in any proper coloring. Thus each vertex in V_i has at least two differently colored neighbors which implies that V_i admits dynamic coloring. $\square$

We now present the main theorem of the section.

Theorem 8. Dynamic Coloring *can be solved in* $O(q^{2.5q+o(q)}n)$ *time, where* $q = 2^t$ *and* t *is the neighborhood diversity of* G*.*

We use *Integer Linear Programming* (ILP) to show that Dynamic Coloring is FPT when parameterized by neighborhood diversity. The following result shows that ILP is FPT when parameterized by the number of variables.

Theorem 9. **([7,9,12]).** An ILP feasibility instance of size n can be solved in $O(q^{2.5q+o(q)}n)$ time and $n^{O(1)}$ space, where q is the number of variables.

The crux of the proof is to distribute the colors across the type sets in a dynamic coloring of G (if one exists). Instead of looking at the list of colors featuring in the types of T_A in a dynamic coloring, where $A \subseteq [t]$, we are only interested in the number of colors that appear exclusively in each of the types of T_A in the coloring.

We now define variables and constraints for ILP. For each subset $A \subseteq [t]$, we have a variable n_A that denotes the number of colors used exclusively in all the types of T_A and not used in any of the types $\{V_1, V_2, \ldots, V_t\} \setminus T_A$. For example, if $A = \{4, 6\}$ (i.e., $T_A = \{V_4, V_6\}$) and $n_A = 3$, then there are three colors say c_1, c_2, c_3 (the exact values of which will be decided later) where each of the colors is used in both V_4 and V_6. Moreover, the colors c_1, c_2, c_3 are not assigned to any of the vertices in types $\{V_1, V_2, \ldots, V_t\} \setminus \{V_4, V_6\}$. Notice that the number of variables is at most 2^t. With this, we proceed to describe the constraints of ILP.

(C0) Consider only those subsets types T_A such that there do not exist types $V_i, V_j \in T_A$ such that $V_i \in adj(V_j)$.
We only consider those subset types T_A which do not have a pair of adjacent types. This constraint ensures that, if two types V_i and V_j are adjacent then the set of colors used in V_i is disjoint from the set of colors used in V_j.

(C1) The sum of all the variables is at most k. That is $\sum\limits_{A} n_A \leq k$.
This constraint ensures that the number of colors used in any coloring is at most k.

(C2) For each clique type V_i, $1 \leq i \leq t$, 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 $\sum\limits_{A: V_i \in T_A} n_A = |V_i|$.
This constraint ensures that the number of colors used for coloring a clique type V_i in any coloring is equal to $|V_i|$.

(C3) For each independent type V_i, $1 \leq i \leq t$, the sum of the variables n_A for which $V_i \in T_A$ is at most the minimum of k and $|V_i|$. That is, $\sum\limits_{A: V_i \in T_A} n_A \leq \min\{k, |V_i|\}$. Also the sum of variables n_A for which $V_i \in T_A$ is at least one. That is, $\sum\limits_{A: V_i \in T_A} n_A \geq 1$.
This constraint ensures that, in any coloring the number of colors used for coloring an independent type V_i is (i) at most the minimum of k and $|V_i|$, and (ii) at least one.

(C4) For each independent type V_i, $1 \leq i \leq t$, if the degree of every vertex in V_i is at least two then the sum of variables n_A for which there exists $V_j \in adj(V_i) \cap T_A$ is at least 2. That is, $\sum\limits_{A: \exists V_j \in adj(V_i) \cap T_A} n_A \geq 2$.
This constraint ensures that the number of colors used in the neighborhood of any vertex (with degree at least two) in an independent type is at least two.

(C5) For each $A \subseteq [t]$, $n_A \geq 0$.
The number of colors used exclusively in all the types in T_A is at least 0.

We use Theorem 9 to obtain a feasible assignment for ILP, if one exists. We claim the following: there is a feasible assignment of ILP if and only if there is a dynamic coloring of G using at most k colors.

Feasibility Implies Colorability: Using a feasible assignment of variable values returned by ILP, we construct a dynamic coloring $f : V(G) \rightarrow [k]$ that assigns colors greedily to the vertices of G, a pseudo-code is presented as Algorithm 1.

We now show that f is a dynamic coloring of G. To show this, we need to show that (a) every vertex is colored, (b) f is a proper coloring and (c) every vertex with degree at least two has two distinctly colored neighbors. Every vertex is assigned a color in Algorithm 1. The constraint (C0) ensures that no color is used in both V_i and V_j if they are adjacent. The constraint (C2) ensures that no two vertices in a clique type are assigned the same color. Hence f is a proper coloring of G.

We now show that for each vertex v with degree at least two, $|f(N(v))| \geq 2$. Since f is a proper coloring, from Observation 7, we have that each clique type V_i admits dynamic coloring. Consider an independent type V_j. Since the graph is connected, we have that $|adj(V_j)| \geq 1$. If there exists a clique type $V_\ell \in adj(V_j)$ then each vertex v in V_j has $|f(N(v))| \geq 2$. This is because the size of a clique type V_ℓ is at least two and f is a proper coloring. Otherwise, if all the types in $adj(V_j)$ are independent types, then there exists at least two distinctly colored vertices in the neighborhood of every vertex in V_j due to constraint (C4). Thus f is a dynamic coloring of G.

Algorithm 1: A dynamic coloring from a feasible assignment of ILP.

```
   Input: $T_A$ and $n_A$, for each $A \subseteq [t]$
   Output: A dynamic coloring of $G$
 1 $C(T_A) = \emptyset$, for each $A \subseteq [t]$      ▷ $C(T_A)$: set of colors associated to $T_A$
 2 $c = 0$                                              ▷ $c$: color counter
 3 for each $A \subseteq [t]$ do
 4     $C(T_A) = \{c+1, c+2, \ldots, c+n_A\}$
 5     $c = c + n_A$
 6 $C(V_i) = \emptyset$, for each $i \in [t]$           ▷ $C(V_i)$: set of colors associated to $V_i$
 7 for each $i \in [t]$ do
 8     $C(V_i) = \bigcup_{A:V_i \in T_A} C(T_A)$
 9 for each $i \in [t]$ do
10     if $V_i$ is a clique type then
11         Color the vertices of $V_i$ from $C(V_i)$ such that each vertex is
            assigned a distinct color
12     else
13         Color the vertices of $V_i$ from $C(V_i)$ such that each color of $C(V_i)$ is
            used at least once
14 return (Coloring of $G$)
```

Colorability Implies Feasibility: Given a dynamic coloring $f : V(G) \to [k]$ of G, we find a feasible assignment to the ILP. For each $A \subseteq [t]$, we set n_A to be the number of colors that are assigned exclusively to each of the types in T_A, that is

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

We now show that such an assignment satisfies the constraints (C0)-(C5). As f is a proper coloring, no two adjacent types share the same color. Hence the constraint (C0) is satisfied.

Each color $c \in [k]$ is uniquely associated with a type T_A, where c is used in all the types of T_A and not used in any of the types $\{V_1, V_2, \ldots, V_t\} \setminus T_A$. The color c is therefore contributed to the variable n_A and not contributed to any other variable $n_{A'}$ where $A \neq A'$. Hence we get that the sum of variables is at most k. Hence the constraints (C1) and (C5).

Every vertex in a clique type V_i is assigned a distinct color in f and hence the sum of variables n_A such that $V_i \in T_A$ is equal to $|V_i|$. Hence the constraint (C2) is satisfied.

Consider an independent type V_j. Clearly $|f(V_j)| \leq \min\{k, |V_j|\}$. Hence the constraint (C3) is satisfied. For each vertex $v \in V_j$ of degree at least two, we have that $|f(N(v))| \geq 2$. That is there are two distinct colors in the neighboring types of V_j. Since these colors contribute to some variables, the constraint (C4) is satisfied.

Running Time: The running time of the algorithm is proportional to the time needed to (i) reduce dynamic coloring problem to the ILP problem, and (ii) find a feasible solution to the created ILP instance. The constraint (C0) considers the subset types from the 2^t subset types such that no two types in a subset type are adjacent. This can be done in $O(2^t t^2)$ time. The constraints (C1) and (C5) can be constructed in $O(2^t)$ time. The constraints (C2) and (C3) can be constructed in $O(t2^t)$ time. The constraint (C4) can be constructed in $O(t2^t)$ time. Finding a feasible assignment of ILP using Theorem 9 takes $O(q^{2.5q+o(q)}n)$ time, where $q = 2^t$. The latter part dominates the former and hence the overall running time is $O(q^{2.5q+o(q)}n)$ time, where $q = 2^t$.

This completes the proof of Theorem 8.

8 Parameterizations by Twin-Cover and Clique-Width

Theorem 10 (⋆). Dynamic Coloring *can be solved in* $O(q^{2.5q+o(q)}n)$ *time, where* $q = 2^{t+2^t}$ *and* t *is the size of the twin-cover of* G.

Theorem 11 (⋆). Dynamic Coloring *can be solved in* $O(3^{O(wk)}poly(n))$ *time where* w *is the clique-width of the graph* G *and* k *is the number of colors.*

9 Conclusion

In this paper, we study Dynamic Coloring on various restricted graph classes and from the viewpoint of parameterized complexity. We presented polynomial time algorithms for chordal graphs and biconvex graphs. We have strengthened the NP-completeness result for bipartite graphs by showing that the problem remains NP-complete for star-convex bipartite graphs, comb-convex bipartite graphs and perfect elimination bipartite graphs. We show that the problem is FPT when parameterized by neighborhood diversity, twin-cover or the combined parameters clique-width and the number of colors.

We conclude the paper with the following list of open problems.

1. What is the parameterized complexity of Dynamic Coloring parameterized by (a) distance to cluster, and (b) distance to co-cluster?
2. What is the complexity of Dynamic Coloring for (a) convex bipartite graphs, and (b) permutation graphs?
3. Will our results hold for a generalized version of the problem called r-Dynamic Coloring [14], where every vertex v of degree at least one, is adjacent to at least $\min\{r, d\}$ many colors, where d is the degree of v?.

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

References

1. Alishahi, M.: Dynamic chromatic number of regular graphs. Discret. Appl. Math. **160**(15), 2098–2103 (2012)
2. Brandstädt, A., Lozin, V.V.: On the linear structure and clique-width of bipartite permutation graphs. Ars Comb. **67**, 273–281 (2003)
3. Chen, Y., Fan, S., Lai, H.-J., Song, H., Sun, L.: On dynamic coloring for planar graphs and graphs of higher genus. Discret. Appl. Math. **160**(7–8), 1064–1071 (2012)
4. Cygan, M., et al.: Parameterized Algorithms, vol. 4. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21275-3
5. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity, vol. 4. Springer, Cham (2013). https://doi.org/10.1007/978-1-4471-5559-1
6. Enright, J.A., Stewart, L., Tardos, G.: On list coloring and list homomorphism of permutation and interval graphs. SIAM J. Discret. Math. **28**(4), 1675–1685 (2014)
7. Frank, A., Tardos, É.: An application of simultaneous diophantine approximation in combinatorial optimization. Combinatorica **7**(1), 49–65 (1987)
8. Jahanbekam, S., Kim, J., Suil, O., West, D.B.: On r-dynamic coloring of graphs. Discrete Appl. Math. **206**, 65–72 (2016)
9. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)
10. Kim, S.J., Lee, S.J., Park, W.J.: Dynamic coloring and list dynamic coloring of planar graphs. Discrete Appl. Math. **161**(13), 2207–2212 (2013)
11. Lampis, M.: Algorithmic meta-theorems for restrictions of treewidth. Algorithmica **64**(1), 19–37 (2012)
12. Lenstra, H.W.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4), 538–548 (1983)
13. Li, X., Yao, X., Zhou, W., Broersma, H.: Complexity of conditional colorability of graphs. Appl. Math. Lett. **22**(3), 320–324 (2009)
14. Montgomery, B.: Dynamic Coloring of Graphs. West Virginia University, Morgantown (2001)
15. Rose, D.J., Tarjan, R.E., Lueker, G.S.: Algorithmic aspects of vertex elimination on graphs. SIAM J. Comput. **5**(2), 266–283 (1976)
16. Saqaeeyan, S., Mollaahamdi, E.: Dynamic chromatic number of bipartite graphs. Sci. Ann. Comput. Sci. **26**(2), 249 (2016)