



# Coloring a Dominating Set Without Conflicts: $q$ -Subset Square Coloring

V. P. Abidha<sup>✉</sup>, Pradeesha Ashok, Avi Tomar, and Dolly Yadav

International Institute of Information Technology Bangalore, Bangalore, India  
{abidha.vp,pradeesha,Avi.Tomar,dolly.yadav}@iiitb.ac.in

**Abstract.** The *Square Coloring* of a graph  $G$  refers to coloring of vertices of a graph such that any two distinct vertices which are at distance at most two receive different colors. In this paper, we initiate the study of a related coloring problem called the *subset square coloring* of graphs. Broadly, the subset square coloring of a graph studies the square coloring of a dominating set of a graph using  $q$  colors. Here the aim is to optimize the number of colors used. This also generalizes the well-studied Efficient Dominating Set problem. We show that the  $q$ -SUBSET SQUARE COLORING problem is NP-hard for all values of  $q$  even on bipartite graphs. We further study the parameterized complexity of this problem when parameterized by a number of structural parameters. We further show bounds on the number of colors needed to subset square color some graph classes.

**Keywords:** Graph coloring · Square coloring · Subset square coloring · Parameterized algorithm · Dominating set

## 1 Introduction

Graph coloring is an important problem in the area of graph theory. For a graph  $G(V, E)$ , the vertex coloring of  $G$  refers to a function  $f$  from the vertex set  $V$  to a set of colors. There are different types of graph coloring problems based on the constraints imposed on this function. A very popular graph coloring question is the Proper Coloring where any two adjacent vertices are to be assigned different colors. Also, several other variants of graph coloring exist, like harmonious coloring, sigma coloring, metric coloring and acyclic coloring. In addition to the theoretical interest, graph coloring problems are motivated by applications in various fields like register allocation in compilers, job scheduling, transportation networks, etc. See [6] for a detailed reading of graph coloring.

A number of graph coloring problems are motivated by a problem in Communication called the *Channel Allocation* problem. Here there exist transmitters  $v_1, v_2, \dots, v_n$  and a transmitter may interfere with another transmitter due to a number of reasons. Now the goal is to assign frequencies to the transmitters such that clear reception of signals is guaranteed. This can be represented as a graph where every vertex corresponds to a transmitter and the interference between transmitters is captured by the distance between the corresponding vertices in

the graph. Here the frequency assigned to a transmitter corresponds to the color assigned to the corresponding vertex. In the 90s, Griggs and Yeh [15], introduced a concept of assigning colors (equivalently, non-negative integers) to vertices such that the assignment of colors to any two vertices depends on whether they are at distance at most two. This is called the  $L(h, k)$ -coloring of graphs. A coloring  $c$  of graph  $G$  is an  $L(h, k)$ -coloring if for any two vertices  $u, v \in V(G)$ ,  $|c(u) - c(v)| \geq h$  if  $u$  and  $v$  are at distance 1 and  $|c(u) - c(v)| \geq k$  if  $u$  and  $v$  are at distance 2. Thus,  $L(1, 0)$ -coloring is the proper coloring itself. Other versions of this problem based on different values of  $h$  and  $k$  are well-studied [5]. Note that  $L(1, 1)$ -coloring involves coloring of vertices with non-negative integers such that the colors on adjacent vertices differ by at least 1 and the colors on vertices at distance 2 also differ by at least 1. This graph coloring is also referred to as *Square coloring* [4, 21] since it is equivalent to the proper coloring of the square of a graph.

We initiate the study of a variant of Square coloring called subset square coloring. This is defined as follows:

**Definition 1.** *Let  $G = (V, E)$  be an undirected graph. A coloring function  $c : V(G) \rightarrow \{c_0, c_1 \dots c_q\}$  is called a  $q$ -subset square coloring of  $G$  if it satisfies the following constraints:*

- For every vertex  $v$  and every color  $c_i, 1 \leq i \leq q$ , we have  $|c^{-1}(c_i) \cap N(v)| \leq 1$ .
- A vertex  $v$  can have at most  $\deg(v)$  vertices with color  $c_0$  in  $N[v]$ , where  $\deg(v)$  refers to degree of  $v$ .

Here, intuitively, assigning the color  $c_0$  to a vertex  $v$  corresponds to  $v$  being uncolored. In this paper we refer to a vertex being uncolored and a vertex colored  $c_0$ , interchangeably. Thus the definition implies that every vertex has at least one colored vertex in its closed neighborhood and no color is repeated in the closed neighborhood. Note that the set of colored vertices form a dominating set of the graph  $G$  (A vertex is said to be dominated, if coloured vertices in its closed neighbourhood is from the set  $\{c_1 \dots c_q\}$ ). Therefore the subset square coloring is equivalent to square coloring a dominating set of the graph using  $q$  colors.

For a given graph  $G$ , let  $\chi_{ssc}(G)$  represent the minimum value of  $q$  such that there exists a  $q$ -subset square coloring of  $G$ . We also study the following algorithmic question. Given a graph  $G$ , the  $q$ -SUBSET SQUARE COLORING problem is defined as follows,

**Input:** Graph  $G$  and  $q \in \mathbb{N}$ .

**Question:** Can  $G$  be  $q$ -subset square colored?

The concept of subset square coloring is previously studied in the context of a classic problem in Computational Geometry called the Art Gallery Problem. Given a polygon  $P$ , along with two sets,  $M$  and  $G$ , of points in  $P$ , the Art Gallery problem is to find  $G' \subseteq G$  such that every point in  $M$  is seen by at least one point in  $G'$ . Motivated by applications in Robotics, Erickson and LaValle [9], introduced the *Chromatic Art Gallery Problem*. Here, the aim is to find a subset  $G' \subseteq G$  such that  $G'$  can be colored using  $q$  colors and every point  $m \in M$  is

seen by at least one point in  $G'$  and moreover, every point in  $G'$  that sees  $m$  gets a distinct color. It is easy to see that for the case where  $M$  and  $G$  are the same finite sets, the Chromatic Art Gallery Problem can be reduced to subset square coloring of a visibility graph. We believe there exists many other application areas related to Channel Allocation where the subset square coloring of graphs becomes useful.

Another motivation for studying the  $q$ -SUBSET SQUARE COLORING problem is that many graphs tend to use much smaller number of colors for subset square coloring when compared to number of colors needed for square coloring. For example, complete graphs, star graphs, wheel graphs etc. need  $O(n)$  colors for square coloring whereas subset square coloring can be done using one color. This will be useful in many applications where the number of colors corresponds to a resource that needs to be optimized.

We now explore some problems that are related to the  $q$ -SUBSET SQUARE COLORING problem.

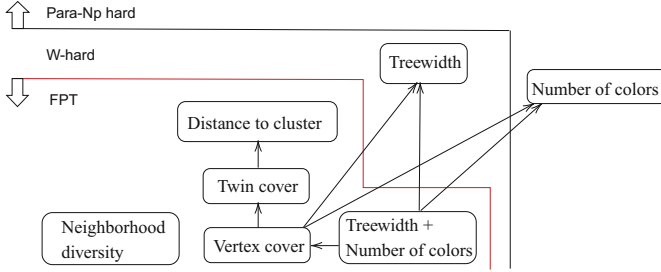
**Related Problems:** The problem of Harmonious coloring, was first introduced in 1983 by Hopcroft and Krishnamoorthy [16] which is defined as follows: The harmonious chromatic number of a graph  $G$ , denoted by  $h(G)$ , is the least number of colors which can be assigned to the vertices of  $G$  such that each vertex has exactly one color, adjacent vertices have different colors, and any two edges have different color pairs. Later, Yue Li Wang et al. [22] developed the concept of  $d$ -Local Harmonious Chromatic problem which generalized the Harmonious Chromatic problem. The  $d$ -Local Harmonious (or just  $d$ -Harmonious) chromatic problem imposes a restriction that the different color-pair requirement is only asked to be satisfied for every edge within distance  $d$  for any vertex. Thus the 1-Harmonious chromatic problem is same as the Square coloring problem.

The problem of Efficient Dominating Set [1] for a given graph is also of interest while we study the subset square coloring problem. An efficient dominating set is one which is simultaneously an independent and a perfect dominating set. A perfect dominating set  $P$  is one in which each vertex  $v \in V(G)$  has exactly one neighbor in  $N(v)$  that belongs to  $P$ , whereas an independent dominating set  $I$  satisfies the condition that set of vertices in  $I$  form an independent set. Specifically, efficient dominating set is a special case of a subset square coloring with  $q = 1$ .

## 2 A Discussion of Results

In this section, we give a summary of our results.

We have already mentioned that the EFFICIENT DOMINATING SET problem is a special case of  $q$ -SUBSET SQUARE COLORING. The EFFICIENT DOMINATING SET problem is already known to be  $NP$ -hard [1]. Thus the  $q$ -SUBSET SQUARE COLORING problem is  $NP$ -hard for  $q = 1$ . We prove that the  $q$ -SUBSET SQUARE COLORING problem with  $q = 2$  is  $NP$ -hard even on planar bipartite graphs and the  $q$ -SUBSET SQUARE COLORING problem is  $NP$ -hard even on bipartite graphs, for all values of  $q$ . Moreover, it is known that for an arbitrary graph  $G$ , it is  $NP$ -hard to check if  $G$  admits an efficient dominating set.



**Fig. 1.** Summary of parameterized results.

We consider the parameterized complexity of the  $q$ -SUBSET SQUARE COLORING problem. For any problem, an interesting parameter to be studied is the size of the solution. For the  $q$ -SUBSET SQUARE COLORING problem, this parameter will be  $q$ , the number of colors used. However, this turns out to be much harder than a  $W$ -hard problem in that it is unlikely to admit an algorithm of running time of the form  $f(q)n^{g(q)}$ .

**Lemma 1.** *The  $q$ -SUBSET SQUARE COLORING problem parameterized by  $q$  is para-NP-hard.*

The lemma follows from the fact that the  $q$ -SUBSET SQUARE COLORING problem is NP-hard even for  $q = 1$ . Moreover, the next result shows that the problem remains  $W$ -hard even on graphs of diameter 2.

**Theorem 1.** *The  $q$ -SUBSET SQUARE COLORING problem parameterized by  $q$  is  $W[2]$ -hard on graphs of diameter 2.*

*Proof.* For a graph  $G$ ,  $\chi_{ssc}(G)$  is bounded by the size of the minimum dominating set of  $G$ . When the diameter of a graph is two, all vertices in any dominating set are at distance at most two. Therefore we can not repeat the colors of vertices in a dominating set. This implies that  $\chi_{ssc}(G)$  is equal to the size of the minimum dominating set. The minimum dominating set problem is known to be  $W[2]$ -hard on graphs of diameter 2 [17]. Thus the result follows.  $\square$

With respect to Theorem 1, we note that the  $q$ -SUBSET SQUARE COLORING problem is polynomial time solvable on planar graphs with diameter 2 [18]. Next, we consider several structural parameters. A well studied structural parameter for graph problems is the treewidth of the graph. Several hard problems are shown to be FPT when parameterized by treewidth. However, the  $q$ -SUBSET SQUARE COLORING problem can be shown to be  $W$ -hard when parameterized by treewidth. Next we consider treewidth and number of colors as a combined parameter and show that this is FPT. This result is proved using a standard technique in the fixed parameter tractable algorithm design called dynamic programming over treewidth.

Next we consider structural parameters which are possibly larger than treewidth. One such well-studied parameter is the size of the vertex cover of the graph. We give an *FPT* algorithm for the  $q$ -SUBSET SQUARE COLORING problem parameterized by the size of the vertex cover. The size of vertex cover is usually a large parameter, especially for dense graphs. Therefore, we study a parameter whose value is small on dense graphs called the neighborhood diversity. It is also a parameter whose value can be computed in polynomial time. We give an *FPT* algorithm for the  $q$ -SUBSET SQUARE COLORING problem parameterized by neighborhood diversity. We further consider a parameter that is provably smaller than the size of vertex cover, called the *distance to cluster graph* and show that the  $q$ -SUBSET SQUARE COLORING problem parameterized by distance to cluster graph is also *FPT*. We further consider a parameter called *twin cover* whose value typically lies between those of distance to cluster graph and size of vertex cover. Since the  $q$ -SUBSET SQUARE COLORING problem parameterized by distance to cluster graph is *FPT*, the  $q$ -SUBSET SQUARE COLORING problem parameterized by twin cover is also *FPT*. However, we give an algorithm with a better running time.

For these problems, as our goal is to only show whether or not the problem is *FPT*, we do not try to optimize the running times. See Fig. 1 for a summary of the results in parameterized complexity of the  $q$ -SUBSET SQUARE COLORING problem.

**Table 1.** Summary of bounds for  $\chi_{ssc}$  for different graph classes.

Graph classes	$\chi_{ssc}$ : Upper bound	$\chi_{ssc}$ : Lower bound
Path $P_n$	1	1
Cycle $C_{3n}$	1	1
Cycle $C_{3n+1}$ or $C_{3n+2}$	2	2
Complete graph $K_n$	1	1
Complete bipartite graph $G_{n,m}$	2	2
Bipartite permutation graph $G_{n,m}$	4	2
Planar graphs with diameter 2	3	2
Grid graph $G_{n \times m}$	2	2
Cograph $G_n$	2	2
Threshold graph $G_n$	1	1
Caterpillar graph $G(n, r_1, r_2, \dots, r_t)$	3	3
Tree $T(V, E)$	$O(\Delta)$	$O(\Delta - 1)$
Tree $T(V, E)$	$O(n)$	$\Omega(\sqrt{n})$
Split graph $T(V, E)$	$O(n)$	$\Omega(\sqrt{n})$

Next we study bounds on  $\chi_{ssc}(G)$  when  $G$  belongs to certain graph classes. It is easy to see that various graph classes (that include many sparse and dense graph classes) like complete graphs, cluster graphs, star graphs, wheel graphs,

paths, cycles, grid graphs etc. are  $q$ -subset square colorable where  $q$  is a constant. However, when we consider trees, we show that there exist trees with  $n$  vertices, that requires  $O(\sqrt{n})$  colors to be subset square colored. As trees form a sub-class of bipartite graphs, this result extends for the class of bipartite graphs also. However, we show that a well-defined sub-class of bipartite graphs, called the Bipartite permutation graphs are 4-subset square colorable. We further show that the class of threshold graphs are 1-subset square colorable. Note that threshold graphs lie in the intersection of split graphs and cographs. We observe that, while cographs, like threshold graphs, are subset square colorable using a constant number of colors, there exist split graphs which require  $O(\sqrt{n})$  colors to be subset square colored. See Table 1 for a summary of results.

### 3 NP–Hardness

In this section, we show that the  $q$ -SUBSET SQUARE COLORING problem is NP-hard, for all values of  $q$ . Note that the result is known for  $q = 1$ . Now, we consider  $q = 2$ .

**Theorem 2.** *The  $q$ -SUBSET SQUARE COLORING problem, where  $q = 2$  is NP-hard, even on planar bipartite graphs.*

*Proof.* We give a reduction from the planar Exact cover by 3-sets(X3C) problem.

#### Planar X3C (Exact cover by 3-sets)

*Input:* A finite set  $X$  with  $|X| = 3n$  and a collection  $S$  of 3–element subsets of  $X$  with  $|S| = m$ .

*Question:* Does  $S$  contain an exact cover for  $X$ , i.e., a sub collection  $S' \subseteq S$  such that every element of  $X$  occurs in exactly one member of  $S'$ ?

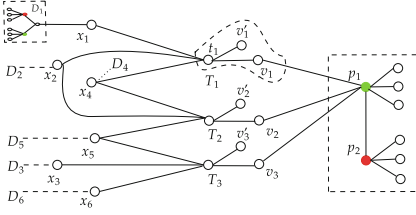
In Planar X3C problem, we have the added constraint that a bipartite graph  $M$  such that  $V(M)$  corresponds to  $X \cup S$  and  $E(M)$  is  $\{(x, s) | x \in X, s \in S, x \in s\}$  is planar.

Let  $(U, S)$  be an instance of the planar X3C problem, where  $U = \{u_1, u_2, \dots, u_{3n}\}$  and  $S = \{S_1, S_2, \dots, S_m\}$ . We construct a planar bipartite graph  $G$  as follows: For every element  $u_i$ , we add a vertex  $x_i$  in  $G$  and connect it with an *element gadget*  $D_i$  in  $G$ . For  $1 \leq i \leq 3n$ ,  $D_i$  is a tree rooted at a vertex  $d_i$ , as shown in Fig. 3(a). Each of the two child nodes of  $d_i$  are connected to three leaves. For every set  $S_j$ ,  $1 \leq j \leq m$ , we add a *set gadget*  $T_j$  with a vertex  $t_j$  attached to two leaves  $v_j$  and  $v'_j$ . We also add a *palette gadget*  $P$  which has two vertices  $p_1$  and  $p_2$  adjacent to each other and each of them attached to three vertices of degree one. See Fig. 3(b).

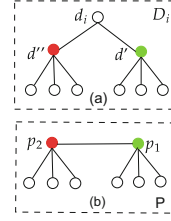
Further, for  $1 \leq i \leq 3n, 1 \leq j \leq m$ , we add an edge between  $x_i$  and  $t_j$  in  $G$ , if  $u_i \in S_j$  in  $(U, S)$ . We also add the edge between  $v_j$  and  $p_1 \in P$ , for all  $1 \leq j \leq m$ .

We claim that  $(U, S)$  has an exact-3-cover if and only if  $G(V, E)$  has a subset square coloring using two colors (proof is given in the full version).

Now, the result follows from the NP-hardness of the planar X3C problem [12] (Fig. 2).  $\square$



**Fig. 2.** Constructed graph  $G_S$  from Exact cover by 3-sets instance of  $S_1 = \{x_1, x_2, x_4\}$ ,  $S_2 = \{x_2, x_4, x_5\}$  and  $S_3 = \{x_3, x_5, x_6\}$ .



**Fig. 3.** (a) Element gadget, (b) Palette gadget

**Theorem 3 (\*).**<sup>1</sup> *The  $q$ -SUBSET SQUARE COLORING problem, where  $q > 2$  is NP-hard, even on bipartite graphs.*

## 4 Parameterized Complexity

In this section, we study the parameterized complexity of the  $q$ -SUBSET SQUARE COLORING problem.

A parameterized problem is a language  $L \subseteq \Sigma^* \times \mathbb{N}$ , where  $\Sigma$  is a fixed, finite alphabet. For an instance  $(x, k) \in \Sigma^* \times \mathbb{N}$ ,  $k$  is called the parameter. The complexity class FPT contains all fixed parameter tractable problems that have an algorithm, a computable function  $f : \mathbb{N} \rightarrow \mathbb{N}$ , and a constant  $c$  such that, given  $(x, k) \in \Sigma^* \times \mathbb{N}$ , the algorithm correctly decides whether  $(x, k) \in L$  in time bounded by  $f(k) \cdot |(x, k)|^c$ . Theory of intractability of parameterized problems orders the problems into a hierarchy called the  $W$ -hierarchy based on its complexity. It is organized as  $\text{FPT} \subseteq W[1] \subseteq W[2] \dots$ . Under standard complexity theoretical assumptions, a problem which is  $W[i]$ -hard does not admit FPT algorithms, where  $i > 0$ . For detailed reading of parameterized complexity refer [8].

### 4.1 Parameterized by Treewidth

We first consider the treewidth of the graph as a parameter. We begin by defining treewidth.

**Tree Decomposition:** [8] A tree decomposition of a graph  $G$  is a tree  $T$  in which each vertex  $i \in T$  has an assigned set of vertices  $X_i \subseteq V$ , called the bag, such that  $\bigcup_{i \in T} X_i = V$ , with some properties: (a) if  $u \in X_i$  and  $u \in X_k$ , then  $u \in X_j$  for all  $j$  on the path from  $i$  to  $k$  in  $T$ . (b) For any edge  $e(u, v) \in E(G)$ , there exists an  $i \in T$  such that  $u, v \in X_i$ .

The width of a tree decomposition  $T$  is the size of the largest bag of  $T$  minus one, and the treewidth of a graph  $G$ , denoted by  $\tau(G)$ , is the minimum width over all possible tree decompositions of  $G$ .

<sup>1</sup> Proofs of results that are marked with a star are given in the full version.

**Theorem 4 (\*).** *The  $q$ -SUBSET SQUARE COLORING problem parameterized by tree-width is  $W[1]$ -hard.*

## 4.2 Parameterized by Treewidth and Number of Colors

We now consider the  $q$ -SUBSET SQUARE COLORING problem parameterized by treewidth and number of colors and give an *FPT* algorithm. We use a standard technique called dynamic programming over treewidth, which gives a constructive proof for the fixed parameter tractability. We use a modified tree decomposition called the nice tree decomposition.

**Nice Tree Decomposition:** [8] A tree decomposition with a distinguished root is called a *nice tree decomposition* if:

- All leaf nodes and the root node have empty bags, i.e.,  $X_l = X_r = \phi$ , where  $r$  is the root node and  $l$  is a leaf node.
- Every other node in the tree decomposition falls in one of the three categories:

**Introduce Node:** An introduce vertex node  $t$  has exactly one child  $t'$  such that  $X_t = X_{t'} \cup \{v\}$  for some  $v \notin X_{t'}$ .

**Forget Node:** A forget node  $t$  has exactly one child  $t'$  such that  $X_t = X_{t'} \setminus \{w\}$  for some  $w \in X_{t'}$ .

**Join Node:** A join node  $t$  has exactly two children  $t_1$  and  $t_2$ , such that  $X_t = X_{t_1} = X_{t_2}$ .

**Introduce Edge Node:** An introduce edge node is labeled with an edge  $uv \in E(G)$  such that  $u, v \in X_t$  and has exactly one child node  $t'$  such that  $X_t = X_{t'}$ .

Note that we assume every edge is introduced exactly once and we say that edge  $uv$  is *introduced* at  $t$ . If a join node contains both  $u$  and  $v$ , and the edge  $uv$  exists in  $E(G)$ , we can note that edge  $uv$  will be introduced in the subtree above the join node. Nice tree decomposition enables us to add edges and vertices one by one and perform operations accordingly. This variant of tree decomposition still has  $O(\tau \cdot n)$  nodes, where  $\tau$  is the treewidth of the graph  $G$ .

The following result is known.

**Proposition 1.** *Given a graph  $G$ , in time  $2^{O(\tau)}n$ , we can compute a nice tree decomposition  $(T, \mathcal{X})$  of  $G$  with  $|V(T)| \in |V(G)|^{O(1)}$  and of width at most  $5\tau$ , where  $\tau$  is the treewidth of  $G$  [3].*

With each node  $t$  of the tree decomposition we associate a subgraph  $G_t$  of  $G$  defined as:  $G_t = (V_t, E_t = \{e : e \text{ is introduced in the subtree rooted at } t\})$ . Here,  $V_t$  is the union of all bags present in the subtree rooted at  $t$ .

**Theorem 5.** *The  $q$ -SUBSET SQUARE COLORING is *FPT* when parameterized by the treewidth  $\tau$  of the input graph and the number of colors  $q$ .*

*Proof.* We give an algorithm based on dynamic programming over nice tree decomposition  $(T, \mathcal{X})$  of  $G$ , computed in time  $2^{O(\tau)}n$ , using Proposition 1, of width at most  $5\tau$ , where  $\tau$  is the treewidth of  $G$ . We define subproblems on  $t \in V(T)$



for the graph  $G_t$ . We consider a *partitioning* of bag  $X_t$  by a mapping  $f : X_t \rightarrow \{B, W, R\}$ . For simplicity, we refer to the vertices in each partition respectively as black, white and grey. Each vertex is also assigned another color by a function  $c : X_t \rightarrow \{c_0, c_1, \dots, c_q\}$  and a  $q$ -length tuple, by a function  $\Gamma : X_t \rightarrow \{0, 1, \hat{1}\}^q$ . Roughly speaking, these functions will determine how the “partial” square coloring looks like, when restricted to  $G_t$  and vertices of  $X_t$ .  $c(v)$  denotes the color assigned to  $v$  and  $c(v) = c_0$  denotes that  $v$  is not colored.  $\Gamma(v)[i]$  indicates whether  $v$  has (either in the current graph, or in the “future”) a vertex in its closed neighborhood that has color  $c_i$ .  $\Gamma(v)[i] = 1$  denotes that vertex  $v$  has a vertex in its closed neighborhood of color  $c_i$  in  $G_t$ ,  $\Gamma(v)[i] = \hat{1}$  denotes that vertex  $v$  has a vertex in its closed neighborhood of color  $c_i$ , that is not present in  $G_t$ , but will appear in the “future”, and  $\Gamma(v)[i] = 0$  denotes the absence of color  $c_i$  in the closed neighborhood of  $v$ . We slightly abuse the notation and use  $\Gamma(v)[c_i]$  and  $\Gamma(v)[i]$  interchangeably. In the following we give a detailed insight into the functions  $f$ ,  $c$  and  $\Gamma$ .

**Black**, represented by  $B$ . Every black vertex  $v$  is given a color  $c(v) \neq c_0$  in a subset square coloring.

**Grey**, represented by  $R$ . A grey vertex  $v$  is not colored, not dominated, i.e.  $c(v) = c_0$  and for each  $i \in [q]$ , it has  $\Gamma(v)[i] \in \{0, \hat{1}\}$ .

**White**, represented by  $W$ . A vertex  $v$  that is neither black nor grey is a white vertex. Note that for a white vertex  $v$ ,  $c(v) = c_0$  and there is  $i \in [q]$ , such that  $\Gamma(v)[i] = 1$ .

A tuple  $(t, c, \Gamma, f)$  is *valid* if the following conditions hold for every vertex  $v \in X_t$ :

1.  $f(v) = B \implies c(v) \neq c_0$  and  $\Gamma(v)[c(v)] = 1$ ,
2.  $f(v) = R \implies c(v) = c_0$  and  $\Gamma(v)[i] \in \{0, \hat{1}\}$ ,  $\forall i \in \{1 \dots q\}$ , and
3.  $f(v) = W \implies c(v) = c_0$  and  $\Gamma(v)[i] = 1$  for some  $i \in \{1 \dots q\}$ .

For a node  $t \in V(T)$ , for each valid tuple  $(t, c, \Gamma, f)$ , we have a table entry denoted by  $D[t, c, \Gamma, f]$ . We have  $D[t, c, \Gamma, f] = \text{true}$  if and only if there is  $\text{col} : V_t \rightarrow \{c_0, c_1, \dots, c_q\}$  (where  $c_0$  denotes no color assignment), such that:

1.  $\text{col}|_{X_t} = c$ ,
2. for each  $v \in X_t$  and  $i \in \{1, 2, \dots, q\}$  with  $\Gamma(v)[i] = 1$ , there is exactly one vertex  $u \in N_{G_t}[v]$ , such that  $\text{col}(u) = c_i$ ,
3. for each  $v \in X_t$  and  $i \in [q]$  with  $\Gamma(v)[i] \in \{0, \hat{1}\}$ , there is no vertex  $u \in N_{G_t}[v]$ , such that  $\text{col}(u) = c_i$ , and
4. for each  $v \in V_t \setminus X_t$ , there is at least one vertex  $u \in N_{G_t}[v]$ , such that  $\text{col}(u) \neq c_0$ , and for all such vertices  $u$ , every other  $u' \in N_{G_t}[v]$  have  $\text{col}(u') \neq \text{col}(u)$ .

In the above, such a coloring  $\text{col}$  is called a  $(t, c, \Gamma, f)$ -*good* coloring. (At any point of time wherever we query an invalid tuple, then its value is *false* by default.) Note that  $D[r, \emptyset, \emptyset, \emptyset] = \text{true}$ , where  $r$  is the root of the tree decomposition, if and only if  $G$  admits a subset square coloring using (at most)  $q$  colors.

We define  $f_{v \rightarrow \gamma}$  where  $\gamma \in \{B, W, R\}$ , as the function where  $f_{v \rightarrow \gamma}(x) = f(x)$ , if  $x \neq v$ , and  $f_{v \rightarrow \gamma}(x) = \gamma$ , otherwise. Similarly, we define the functions  $c_{v \rightarrow c_i}$  and  $\Gamma_{v[i] \rightarrow \alpha}$  where  $\alpha \in \{0, \hat{1}, 1\}$ . We now proceed to define the recursive formulas for the values of  $D$ .

**Leaf Node.** For a leaf node  $t$ , we have  $X_t = \emptyset$ . Hence, the only entry is  $D[t, \emptyset, \emptyset, \emptyset]$ . Moreover, by definition, we have  $D[t, \emptyset, \emptyset, \emptyset] = true$ .

**Introduce Vertex Node.** Let  $t$  be the introduce vertex node with a child  $t'$  such that  $X_t = X_{t'} \cup \{v\}$  for some  $v \notin X_{t'}$ . Since the vertex  $v$  is isolated in  $G_t$ , the following recurrence follows.

$$D[t, c, \Gamma, f] = \begin{cases} D[t', c|_{X'}, \Gamma|_{X'}, f|_{X'}] & \text{if } f(v) = B \text{ and } \Gamma(v)[c_i] \in \{0, \hat{1}\}, \forall c_i \neq c(v) \\ D[t', c|_{X'}, \Gamma|_{X'}, f|_{X'}] & \text{if } f(v) = R \\ \text{False} & \text{otherwise} \end{cases}$$

**Introduce Edge Node.** Let  $t$  be an introduce edge node labeled with an edge  $u^*v^*$  and let  $t'$  be the child of it. Thus  $G_{t'}$  does not have the edge  $u^*v^*$  but  $G_t$  has. Consider distinct  $u, v \in \{u^*, v^*\}$ .

1. If  $f(u) = B$ ,  $f(v) = W$  and  $\Gamma(v)[c(u)] = 1$ . We set  $D[t, c, \Gamma, f] = D[t', c, \Gamma_{v[c(u)] \rightarrow \hat{1}}, f_{v \rightarrow R}] \vee D[t', c, \Gamma_{v[c(u)] \rightarrow \hat{1}}, f_{v \rightarrow W}]$  (if any of the entries are invalid, then it is *false*).
2. If  $f(u) = f(v) = B$  and  $\Gamma(v[c(u)]) = \Gamma(u[c(v)]) = 1$ , set  $D[t, c, \Gamma, f] = D[t', c, \Gamma_{v[c(u)] \rightarrow \hat{1}, u[c(v)] \rightarrow \hat{1}}, f]$ .
3. If  $\{f(u), f(v)\} \cap \{B\} = \emptyset$ , then  $D[t, c, \Gamma, f] = D[t', c, \Gamma, f]$ .
4. If none of the above conditions hold then  $D[t, c, \Gamma, f] = false$ .

**Lemma 2 (\*)**. *Recurrence for introduce edge node is correct.*

**Forget Node.** Let  $t$  be a forget node with child  $t'$  such that  $X_t = X_{t'} \setminus \{v\}$  for some  $v \in X_{t'}$ . Since the vertex  $v$  does not appear again in any bag of a node above  $t$ ,  $v$  must be either black or white (otherwise, we set the entry to *false*).

$$D[t, c, \Gamma, f] = \bigvee_{\substack{1 \leq i \leq q \\ \alpha \in \{0, 1\}^q}} (D[t', c_{v \rightarrow c_0}, \Gamma_{v \rightarrow \alpha}, f_{v \rightarrow W}] \vee D[t', c_{v \rightarrow c_i}, \Gamma_{v \rightarrow \alpha}, f_{v \rightarrow B}])$$

**Join Node.** Let us denote the join node by  $t$ . Let  $t_1$  and  $t_2$  be the children of  $t$ . We know that  $X_t = X_{t_1} = X_{t_2}$  and  $X_t$  induces an independent set in the graphs  $G_t, G_{t_1}$  and  $G_{t_2}$ . We say that the pair of tuples  $[t_1, f_1, c_1, \Gamma_1]$  and  $[t_2, f_2, c_2, \Gamma_2]$  are  $[t, f, c, \Gamma]$ -consistent if for every  $v \in X_t$  the following conditions hold.

- If  $f(v) = B$  then  $(f_1(v), f_2(v)) = (B, B)$  and  $c_1(v) = c_2(v) = c(v)$ .
- If  $f(v) = W$  then  $(f_1(v), f_2(v)) \in \{(W, R), (R, W), (W, W)\}$ .
- If  $f(v) = R$  then  $(f_1(v), f_2(v)) = (R, R)$ .
- If  $\Gamma(v)[i] = 0$  then  $(\Gamma_1(v)[i], \Gamma_2(v)[i]) \in \{(0, 0)\}$  for  $1 \leq i \leq q$ .
- If  $\Gamma(v)[i] = 1$  then  $(\Gamma_1(v)[i], \Gamma_2(v)[i]) \in \{(1, \hat{1}), (\hat{1}, 1)\}$  for  $1 \leq i \leq q$ .
- If  $\Gamma(v)[i] = \hat{1}$  then  $(\Gamma_1(v)[i], \Gamma_2(v)[i]) \in \{(\hat{1}, \hat{1})\}$  for  $1 \leq i \leq q$ .

It is easy to see that a vertex  $v$  belongs to Black partition in  $X_t$  if it is Black in  $X_{t_1}$  and  $X_{t_2}$  and the color  $c(v)$  that is assigned to  $v$  is same in these bags. Similarly, we can understand for a vertex in Grey partition. If vertex  $v$

belongs to White partition, it implies that it is dominated in exactly one of the bags  $X_{t_1}$  and  $X_{t_2}$  or both. However if  $v$  is dominated in both the bags, then the color that dominates it is different which is taken care by  $\Gamma(v)[i]$ . The value of  $\Gamma(v)[i]$  is set to 1 for different values of the  $q$ -length tuple. Therefore, for a vertex to belong to White partition requires one of the three combinations:  $\{(W, R), (R, W), (W, W)\}$ . Now consider the values that  $\Gamma(v)[i]$  can take in  $X_t$ . If color  $i$  is not present in  $N[v]$  in any of the child nodes  $t_1$  and  $t_2$ , then it will naturally not be present in  $N[v]$  in  $G_t$ , implying  $\Gamma(v)[i] = 0$ . However,  $\Gamma(v)[i] = 1$  indicates the presence of a vertex with color  $i$  in  $N[v]$  in  $G_t$ . We further note that color  $i$  appears in exactly one of the child nodes  $X_{t_1}$  or  $X_{t_2}$  but not both because presence of color  $i$  in both child nodes implies the presence of two vertices with color  $i$  in  $N[v]$  in  $G_t$ . This follows from the observation that  $X_t$  induces an independent set in  $G_t$ , by the property of nice tree decomposition. The next possibility of  $\Gamma(v)[i]$  is  $\hat{1}$ . For this to be true, the color  $i$  should not be present in  $G_t$  seen so far but will appear in the tree decomposition eventually.

We set  $D[t, c, \Gamma, f] = \bigvee_{(f_1, f_2)} (D[t_1, c_1, \Gamma_1, f_1] \wedge D[t_2, c_2, \Gamma_2, f_2])$ , where  $[t_1, f_1, c_1, \Gamma_1]$  and  $[t_2, f_2, c_2, \Gamma_2]$  is  $[t, f, c, \Gamma]$ -consistent.

We have described the recursive formulas for the values of  $D[\cdot]$ . Note that we can compute each entry in time bounded by  $2^{O(q\tau)} \cdot q^{O(\tau)} n^{O(1)}$ . Moreover, the number of (valid) entries for a node  $t \in V(T)$  is bounded by  $2^{O(q\tau)} \cdot q^{O(\tau)} n^{O(1)}$ , and  $V(T) \in n^{O(1)}$ . Thus we can obtain that the overall running time of the algorithm is bounded by  $2^{O(q\tau)} n^{O(1)}$ .

### 4.3 Parameterized by the Size of Vertex Cover

In this section, we prove that the  $q$ -SUBSET SQUARE COLORING parameterized by the size of vertex cover is FPT. Let  $X$  be a vertex cover of  $G$ ,  $|X| = k$ . First we prove the following result.

**Lemma 3 (\*).** *The number of colors required to subset square color the vertices of graph  $G$  is at most the size of vertex cover of  $G$ .*

Since both the treewidth and number of colors required to subset square coloring is bounded by  $|X|$ , the result follows from Theorem 5.

**Theorem 6.** *The  $q$ -SUBSET SQUARE COLORING parameterized by the size of vertex cover is FPT.*

### 4.4 Parameterized by Neighborhood Diversity Number

We start by defining the neighborhood diversity number of a graph.

**Definition 2.** [13] *Given a graph  $G = (V, E)$ , two vertices  $u, v \in V$  have the same type if and only if  $N(v) \setminus \{u\} = N(u) \setminus \{v\}$ . The graph  $G$  has a neighborhood diversity  $t$ , if there exists a partition of  $V$  into at most  $t$  sets,  $V_1, V_2, \dots, V_t$  such that all the vertices in  $V_i$  have the same type for  $i = 1, 2, \dots, t$ . The family  $\nu = \{V_1, V_2, \dots, V_t\}$  is called the type partition of  $G$ .*

On creating such a type partition of  $V(G)$ , we observe that the vertices within a partition either induces a clique or is an independent set. Further, for  $1 \leq i, j \leq t$ , each vertex in a partition  $V_i$  is either adjacent to every vertex in another partition  $V_j$  or there are no edges between vertices of  $V_i$  and  $V_j$ .

**Lemma 4 (\*).** *The number of colors required to subset square color a graph  $G$  is at most its neighborhood diversity, that is,  $\chi_{ssc}(G) \leq t$ .*

Now we show that  $q$ -SUBSET SQUARE COLORING parameterized by  $t$  is FPT by giving a polynomial kernel.

**Theorem 7.** *The  $q$ -SUBSET SQUARE COLORING problem parameterized by neighborhood diversity admits a polynomial kernel of size  $O(t^2)$ .*

*Proof.* Let  $G$  be a connected graph, along with a type partition of size  $t$ ,  $t > 1$ . Let  $G'(V', E')$  be the graph obtained from  $G(V, E)$  by deleting all but  $q + 1$  vertices from each type partition. We will show that  $G'$  can be  $q$ -subset square colored if and only if  $G$  can be  $q$ -subset square colored. Note that  $G'$  is also a connected graph.

Let  $\chi$  be a  $q$ -subset square coloring of  $G'$ . We claim  $\chi$  is a  $q$ -subset square coloring of  $G$  as well. Let  $V'_i = \{v_1, v_2, \dots, v_{q+1}\}$  be a vertex set in the type partition of  $V(G')$ . If  $v_i$  and  $v_j$ ,  $1 \leq i, j \leq q + 1, i \neq j$ , are colored, then  $\chi(v_i) \neq \chi(v_j)$ . Otherwise, there exists at least one common neighbor for all vertices in  $V'_i$ , since  $G'$  is connected and this is a contradiction. Since there are at most  $q$  colors, there exists at least one uncolored vertex, say  $v_i$  in  $V'_i$ . Now, every vertex in  $V_i \setminus V'_i$  is dominated in  $G$  by the same vertices that dominate  $v_i$  in  $G'$ .

In the reverse direction, assume that  $G$  admits a  $q$ -subset square coloring,  $\chi$ . Then we color the vertices  $v_1, v_2, \dots, v_{q+1}$  in  $V'_i$  arbitrarily using the colors, if any, used by  $\chi$  on  $V_i$ . Now it is easy to see that this is a valid  $q$ -subset square coloring for  $G'$  as well.

Now  $|V(G')| \leq (q + 1)t$ . If  $q \geq t$ , the problem is trivially a YES instance, by Lemma 4. Therefore  $|V(G')| = O(t^2)$ . Now the result follows.

## 4.5 Parameterized by Distance to Cluster Graph

**Definition 3.** *A cluster graph is a disjoint union of complete graphs.*

It is easy to see that cluster graphs can be 1-subset square colored. In this section, let  $X \subseteq V(G)$  such that  $G[V \setminus X]$  is a cluster graph and  $|X| = k$ . Now we observe the connection between  $\chi_{ssc}(G)$  and  $k$ .

**Lemma 5 (\*).**  $\chi_{ssc}(G) \leq |X| + 1$ .

**Theorem 8.** *The  $q$ -SUBSET SQUARE COLORING is FPT when parameterized by distance to cluster graph.*

*Proof.* If  $q \geq |X| + 1$ , then by Lemma 5 it is trivially true that a subset square coloring exists. Therefore, we assume that  $q \leq k$ .

If two or more vertices in a clique in  $G[V \setminus X]$  have the same neighborhood in  $X$ , delete all but one of those vertices. This does not affect the solution as the closed neighborhood of the vertices are the same. Therefore every clique in  $G[V \setminus X]$  has at most  $2^k$  vertices.

Now we bound the number of cliques in  $G[V \setminus X]$ . Let  $\{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_{2^k}\}$  be the family of subsets of  $X$ . For two cliques  $C_a$  and  $C_b$  in  $G[V \setminus X]$ , we say  $C_a$  and  $C_b$  have the same *type* if for all  $1 \leq i \leq 2^k$ , either both  $C_a$  and  $C_b$  each have a vertex whose neighborhood in  $X$  is  $\mathcal{X}_i$  or neither of them has such a vertex. Note that there can be at most  $2^{2^k}$  distinct types of cliques. Now we use the following reduction exhaustively to get a reduced graph  $G'$ . If there exists more than  $q2^k + 1$  cliques of the same type, delete all but  $q2^k + 1$  of them. Thus there are at most  $k + 2^{2^k} \cdot (q2^k + 1)$  vertices in  $G'$ .

We claim that  $G$  has a  $q$ -subset square coloring if and only if  $G'$  has a  $q$ -subset square coloring. Assume  $G'$  admits a  $q$ -subset square coloring. Let  $C_i$  be a clique in  $G \setminus G'$ . Therefore  $G'$  has  $(q2^k + 1)$  cliques of the same type as  $C_i$ , let them be  $C'_1, C'_2, \dots, C'_{q2^k + 1}$ . Similarly, for  $v \in V(C_i)$ , there exists  $v_j \in V(C'_j)$  for all  $1 \leq j \leq q2^k + 1$  such that  $N(v) \cap X = N(v_j) \cap X$ . We show that there exists at least one  $C_j$  such that one of the conditions is true.

- $C_j$  contains a vertex  $v_j$  such that  $N(v_j) \cap X = \emptyset$  and  $v_j$  is colored.
- none of the vertices in  $C_j$  is colored.

If the first condition is true, then we can dominate all vertices in  $C_i$  by coloring the corresponding vertex  $v$  using the same color as  $v_j$ . Now, assume that the first condition is not true. Consider the vertices  $v_j \in V(C'_j)$  for all  $1 \leq j \leq q2^k + 1$ . Since all of them have common neighbors in  $X$ , we can color at most  $q$  such vertices. Since there are at most  $2^k$  vertices in a cliques, there can be at most  $q2^k$  cliques with colored vertices. All the uncolored vertices in a clique is dominated by vertices in  $X$ . The same set of vertices can dominate the vertices in  $C_j$  in  $G$ . The other direction is easy to see.

Since the size of the reduced instance is bounded by a function of  $k$ , it follows that the  $q$ -SUBSET SQUARE COLORING is FPT when parameterized by distance to cluster graph.

#### 4.6 Parameterized by the Size of Twin Cover

**Definition 4.** [11] For a graph  $G(V, E)$  a subset  $X$  of vertices is a twin cover if for every edge  $e = uv \in E(G)$  either (a)  $u \in X$  or  $v \in X$ , or (b)  $u$  and  $v$  are true twins.

Two vertices  $u$  and  $v$  are true twins if every other vertex is either adjacent to both  $u$  and  $v$  or neither of them and  $u$  and  $v$  has an edge between them. It follows from the definition that if  $X$  is a twin cover, then  $G[V \setminus X]$  is a disjoint union of cliques and for every vertex  $v \in X$  and every clique in  $G[V \setminus X]$ ,  $v$  is

either adjacent to every vertex in the clique or  $v$  is not adjacent to any vertex in the clique. Thus for every graph  $G$ , distance to cluster graph of  $G \leq$  size of twin cover  $\leq$  size of vertex cover. Thus it follows from Theorem 8 that the  $q$ -SUBSET SQUARE COLORING problem parameterized by twin cover is FPT. Here, we give an algorithm with better running time.

**Theorem 9.** *The  $q$ -SUBSET SQUARE COLORING problem parameterized by the size of twin cover is FPT.*

*Proof.* Let  $X \subset V$  be a twin cover of the graph  $G(V, E)$  and  $|X| = k$ . It can be seen that  $\chi_{ssc}(G) \leq |X|$ . If  $q \geq k$ , then return *TRUE*. Now, we assume that  $q < k$ .

Consider all possible  $(q + 1)^k$  colorings of  $X$  using  $\{c_0, c_1, \dots, c_q\}$ . Now similar to the proof of Theorem 6, we can try to extend each of these colorings to get a valid subset square coloring. Since all the vertices in the clique in  $G[V \setminus X]$  are true twins, we can delete all but one vertex from every clique to get an equivalent reduced instance. Now the reduced instance is very similar to that in the proof of Theorem 6, and the same algorithm applies.

Therefore the running time of the  $q$ -SUBSET SQUARE COLORING problem parameterized by the size of twin cover is  $\mathcal{O}(k^{2^k})$ .  $\square$

## 5 Bounds on the Number of Colors for the $q$ -SUBSET SQUARE COLORING

In this section, we discuss bounds on the minimum number of colors needed for subset square coloring some graph classes.

### 5.1 Trees

We show that  $\chi_{ssc}(G)$  can be bounded by maximum degree, when  $G$  is a tree. As a corollary, we also show there exists a lower bound on  $\chi_{ssc}(G)$  as a function of  $n$ .

**Lemma 6 (\*).** *Let  $\Delta$  be the maximum degree of a tree. Then  $\Delta$  colors are sufficient to subset square color the tree. Moreover,  $\Delta - 1$  colors are sometimes necessary.*

**Corollary 1.** *There exist trees that require  $\Omega(\sqrt{n})$  colors to be subset square colored.*

Now, we know that Trees are a subclass of Bipartite graphs. Therefore the lower bound applies to the class of Bipartite graphs too. In the next result, we show a sub-class of Bipartite graphs can be subset square colored using constant number of colors.

### 5.2 Bipartite Permutation Graphs

In this section we discuss bounds on  $\chi_{ssc}$  for bipartite permutation graphs.

A graph is a bipartite permutation graph if it is both bipartite and permutation graph. Let  $G(A \uplus B, E)$  be a connected bipartite permutation graph, then it admits the strong ordering, adjacency and enclosure properties, as defined below [20].

- (1) An ordering of the vertices  $A$  in a bipartite graph  $G(A \uplus B, E)$  has the *adjacency property* if for each vertex  $v \in B$ , the vertices in  $N(v)$  are consecutive in the ordering of  $A$ .
- (2) An ordering of the vertices  $A$  in a bipartite graph  $G(A \uplus B, E)$  has the *enclosure property* if for every pair of vertices  $v, u \in B$  such that  $N(v)$  is a subset of  $N(u)$ , vertices in  $N(u) - N(v)$  occur consecutively in the ordering of  $A$ .
- (3) A *strong ordering* of the vertices of a bipartite graph  $G(A \uplus B, E)$  consists of an ordering of  $A$  and an ordering of  $B$  such that for all  $(a, b')$ ,  $(a', b)$  in  $E$ , where  $a, a'$  are in  $A$  and  $b, b'$  are in  $B$ ,  $a < a'$  and  $b < b'$  imply  $(a, b)$  and  $(a', b')$  are in  $E$ .

**Lemma 7.** *If  $G$  is a connected bipartite permutation graph then  $\chi_{ssc}(G) \leq 4$ .*

Let  $A = \{a_1, a_2, \dots, a_n\}$  and  $B = \{b_1, b_2, \dots, b_m\}$  have the strong ordering property. For  $a_j \in A$ , let  $s(a_j) = \min\{i | b_i \in N(a_j)\}$  and  $l(a_j) = \max\{i | b_i \in N(a_j)\}$  be the smallest and largest vertex adjacent to  $a_j$  respectively. (Symmetrically defined for  $B$ ).

Now we color a set of vertices from  $A$ , such that all vertices in  $B$  are dominated. In the first step, we color the first vertex from  $A$ ,  $a_1$ , using color one.

In the  $k$ th step we consider the smallest  $j$  such that  $b_j \in B$  is not dominated. Then we color  $a_i \in A$  such that  $i$  is the largest integer such that  $N(a_i)$  contains  $b_j$ , using color one, if  $k$  is odd, or otherwise, using color two. Repeat this till every vertex in  $B$  is dominated. Now, if  $a_i, a_j, a_k \in A$  are colored in consecutive steps, then  $N(a_i)$  and  $N(a_k)$  are disjoint. For contradiction, assume that  $N(a_i) \cap N(a_k) \neq \emptyset$ . Let  $j' = l(a_i)$ . Then the vertex  $b_{j'+1}$  is dominated by both  $a_j$  and  $a_k$ . This contradicts that  $a_j$  was colored by the algorithm to dominate  $b_{j'+1}$ . Thus no vertex in  $B$  has repeating colors in its neighborhood.

Similarly we can dominate all vertices in  $A$  by coloring vertices in  $B$  using colors three and four. This proves the result.  $\square$

Further, we show that the class of Caterpillar graphs which is a subclass of Bipartite Permutation graphs are 3-subset square colorable.

**Definition 5.** [19] *A Caterpillar graph is a tree such that every vertex is at distance at most one from a central path.*

**Lemma 8.**  $\chi_{ssc}(G) \leq 3$  when  $G$  is a caterpillar graph.

*Proof.* Let  $P$  be the central path of  $G$  with vertices  $v_1, v_2, \dots, v_{n'}$ . Now coloring vertices in  $P$  such that  $v_i$ , for  $1 \leq i \leq n'$ , is given color  $(i \bmod 3 + 1)$  is a valid 3-subset square coloring. The result follows.

### 5.3 Threshold Graph

**Definition 6.** [14] *A graph is a threshold graph if it can be constructed from the empty graph by repeatedly adding either an isolated vertex or a dominating vertex.*

**Lemma 9.** *If  $G$  is a threshold graph then  $\chi_{ssc}(G) = 1$ .*

*Proof.* If isolated vertices are present, color them using the same color. By coloring the last introduced dominating vertex  $v$ , we satisfy subset square coloring  $G$  as each vertex in  $G$  has only one colored vertex  $v$  in its closed neighborhood.

We know that the family of threshold graphs lie in the intersection of split graphs and cographs. We consider these graph classes in subsequent sections.

## 5.4 Split Graph

**Definition 7.** [2] A graph  $G$  is a split graph if  $V(G)$  can be partitioned into two sets  $A$  and  $B$  such that  $A$  induces a clique and  $B$  induces an independent set.

**Theorem 10.** There exist split graphs with  $n$  vertices that require  $\Omega(\sqrt{n})$  colors to be subset square colored.

*Proof.* We will construct a split graph  $G = (A \uplus B, E)$  as follows. Here  $A$  induces a clique and  $B$  induces an independent set. Let  $A = \{v_1, v_2, \dots, v_n\}$  and  $B = \{v_{i,j} | 1 \leq i, j \leq n\}$ . Further we add edges from  $v_{i,j} \in B$  to  $v_i \in A$  and  $v_j \in A$ , for all  $1 \leq i, j \leq n$ . Note that  $G$  has  $n^2 + n$  vertices.

All vertices  $v_{i,i} \in B$  are of degree 1. To dominate  $v_{i,i} \in B$ , either we need to color  $v_i$  or  $v_{i,i}$ , for all  $1 \leq i \leq n$ . If  $v_i$  is colored for all  $i$ ,  $1 \leq i \leq n$ , then we need  $n$  colors since all these vertices are adjacent to each other. Otherwise, assume there exists an  $i$  such that  $v_i$  is not colored and  $v_{i,i}$  is colored. Now the  $n - 1$  vertices  $v_{i,j} \in B$ , where  $i \neq j$  are dominated either by themselves or by their other neighbor  $v_j \in A$ . Note that here every vertex is dominated by a distinct vertex. Thus  $O(n)$  vertices are colored from at least one of the sets,  $\{v_{i,j} | i \neq j\}$  and  $\{v_j | j \neq i\}$ . Since any two vertices from one of these sets are at distance at most 2,  $O(n)$  colors are to be used.

## 5.5 Cographs

We start by showing the connection between  $\chi_{ssc}(G)$  and the modular width of the graph. We first define the modular width of a graph. The modular width of graph  $G$  is computed by virtue of four operations, namely creation of isolated vertex, disjoint union, complete join and substitution. More precisely, the modular width of  $G$  equals the maximum number of operands used by any occurrence of substitution operation. These four operations that are involved in modular decomposition of graph  $G$  are described in [10]. For the sake of completeness, we mention the four operations here.

**Definition 8.** [10] Algebraic operations involved to compute modular width of graph  $G$ .

- Create an isolated vertex;
- The disjoint union of two graphs, i.e., the disjoint union of two graph  $G_1$  and  $G_2$ , denoted by  $G_1 \otimes G_2$ , is the graph with vertex set  $V(G_1) \cup V(G_2)$  and edge set  $E(G_1) \cup E(G_2)$ ;



- The complete join of two graphs, i.e., the complete join of two graphs  $G_1$  and  $G_2$ , denoted by  $G_1 \oplus G_2$ , is the graph with vertex set  $V(G_1) \cup V(G_2)$  and edge set  $E(G_1) \cup E(G_2) \cup \{\{v, u\} : v \in V(G_1) \text{ and } u \in V(G_2)\}$ .
- The substitution operation with respect to some graph  $G$  with vertices  $v_1, \dots, v_n$ , i.e., for graphs  $G_1, \dots, G_n$  the substitution of the vertices of  $G$  by the graphs  $G_1, \dots, G_n$ , denoted by  $G(G_1, \dots, G_n)$ , is the graph with vertex set  $\bigcup_{1 \leq i \leq n} V(G_i)$  and edge set  $\bigcup_{1 \leq i \leq n} E(G_i) \cup \{\{u, v\} : u \in V(G_i) \text{ and } v \in V(G_j), v_i, v_j \in E(G) \text{ and } i \neq j\}$ . Hence,  $G(G_1, \dots, G_n)$  is obtained from  $G$  by substituting every vertex  $v_i \in V(G)$  with the graph  $G_i$  and adding all edges between the vertices of a graph  $G_i$  and the vertices of a graph  $G_j$  whenever  $\{v_i, v_j\} \in E(G)$ .

**Definition 9.** [10] Let  $A$  be an algebraic expression that uses only the four operation as mentioned in Definition 8. We define the width of  $A$  as the maximum number of operands used by any occurrence of the substitution operation in  $A$ . Modular width of graph  $G$ , denoted as  $w(G)$ , is the least integer  $m$  such that  $G$  can be obtained from such an algebraic expression of width at most  $m$ .

Cographs are the graphs that can be constructed from operations—creation of an isolated vertex, disjoint union of two graphs and complete join of two graphs. By definition, the modular width of cographs is two [7].

Now we state our result.

**Lemma 10.** The maximum number of colors required to subset square color a graph  $G$  equals the modular width of  $G$ , that is,  $\chi_{ssc}(G) \leq w(G)$ .

*Proof.* Now we discuss the maximum number of colors required to subset square color  $G$  while we perform those operations. On introducing an isolated vertex, we color it by using one color. To dominate the vertices created as a result of disjoint union of two graphs, we can use maximum number of colors that were used in subset square coloring each subgraphs in the disjoint union operation. Now we consider the complete join operation on the subgraph  $G_c$ . Let  $G_c = G_{c1} \otimes G_{c2}$ . If a vertex  $v$  in  $V(G_{ci})$ ,  $i = \{1, 2\}$  has degree  $G_i(V) - 1$ , then  $\chi_{ssc}(G_c) = 1$ . Otherwise we can color an arbitrary vertex from  $G_{c1}$  and  $G_{c2}$  using two distinct colors. Therefore the value of  $\chi_{ssc}$  is  $\leq 2$ . Finally, we examine the substitution operation. We color an arbitrary vertex  $v$  in each  $G_i$  using distinct colors. Besides the presence of coloured neighbour(s) in  $G_i$ , a vertex may possibly be adjacent to another coloured vertex in  $G_j$ . Therefore the value of  $\chi_{ssc}$  can be at most  $w$ .

**Corollary 2.** If  $G$  is a cograph, then  $\chi_{ssc}(G) \leq 2$ .

## References

1. Bange, D.W.: Efficient dominating sets in graphs. Appl. Discrete Math. **189**, 189–199 (1988)
2. Bertossi, A.A.: Dominating sets for split and bipartite graphs. Inf. Process. Lett. **19**(1), 37–40 (1984)

3. Bodlaender, H.L., Drange, P.G., Dregi, M.S., Fomin, F.V., Lokshtanov, D., Pilipczuk, M.: A  $c \wedge kn$  5-approximation algorithm for treewidth. *SIAM J. Comput.* **45**(2), 317–378 (2016)
4. Yuehua, B., Zhu, X.: An optimal square coloring of planar graphs. *J. Comb. Optim.* **24**(4), 580–592 (2012)
5. Calamoneri, T.: The  $l(h, k)$ -labelling problem: a survey and annotated bibliography. *Comput. J.* **49**(5), 585–608 (2006)
6. Chartrand, G., Zhang, P.: *Chromatic Graph Theory*. Chapman and Hall/CRC, London (2008)
7. Coudert, D., Ducoffe, G., Popa, A.: Fully polynomial FPT algorithms for some classes of bounded clique-width graphs. *ACM Trans. Algorithms (TALG)* **15**(3), 1–57 (2019)
8. Cygan, M., et al.: *Parameterized Algorithms*, vol. 5. Springer, Cham (2015). <https://doi.org/10.1007/978-3-319-21275-3>
9. Erickson, L., LaValle, S.M.: A chromatic art gallery problem. Technical report (2010)
10. Gajarský, J., Lampis, M., Ordyniak, S.: Parameterized algorithms for modular-width. In: Gutin, G., Szeider, S. (eds.) *IPEC 2013*. LNCS, vol. 8246, pp. 163–176. Springer, Cham (2013). [https://doi.org/10.1007/978-3-319-03898-8\\_15](https://doi.org/10.1007/978-3-319-03898-8_15)
11. Ganian, R.: Improving vertex cover as a graph parameter. *Discrete Math. Theor. Comput. Sci.* **17** (2015)
12. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Series of Books in the Mathematical Sciences), 1st edn. W. H. Freeman, New York (1979)
13. Gargano, L., Rescigno, A.A.: Complexity of conflict-free colorings of graphs. *Theoret. Comput. Sci.* **566**, 39–49 (2015)
14. Golumbic, M.C.: *Algorithmic graph theory and perfect graphs* (2004)
15. Griggs, J., Yeh, R.: Labelling graphs with a condition at distance 2. *SIAM J. Discrete Math.* **5**, 586–595 (1992)
16. Hopcroft, J., Krishnamoorthy, M.: On the harmonious coloring of graphs. *SIAM J. Algebraic Discrete Methods* **4**, 306–311 (1983)
17. Lokshtanov, D., Misra, N., Philip, G., Ramanujan, M.S., Saurabh, S.: Hardness of  $r$ -dominating set on graphs of diameter  $(r+1)$ . In: Gutin, G., Szeider, S. (eds.) *IPEC 2013*. LNCS, vol. 8246, pp. 255–267. Springer, Cham (2013). [https://doi.org/10.1007/978-3-319-03898-8\\_22](https://doi.org/10.1007/978-3-319-03898-8_22)
18. MacGillivray, G., Seyffarth, K.: Domination numbers of planar graphs. *J. Graph Theory* **22**(3), 213–229 (1996)
19. Ortiz, C., Villanueva, M.: Maximal independent sets in caterpillar graphs. *Discret. Appl. Math.* **160**(3), 259–266 (2012)
20. Spinrad, J., Brandstädt, A., Stewart, L.: Bipartite permutation graphs. *Discret. Appl. Math.* **18**(3), 279–292 (1987)
21. van den Heuvel, J., McGuinness, S.: Coloring the square of a planar graph. *J. Graph Theory* **42**(2), 110–124 (2003)
22. Wang, Y.L., Lin, T.W., Wang, L.: The local harmonious chromatic problem. In: *Proceedings of the 27th Workshop on Combinatorial Mathematics and Computation Theory*, Taichung, Taiwan (2010)