Template driven rainbow coloring of graphs

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

Master of Technology

in

Computer Science

by

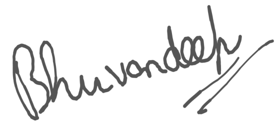
Bhuvandeep
(Roll no. CS2210)

INDIAN STATISTICAL INSTITUTE
KOLKATA - 700108

June, 2024

Declaration

I, **Bhuvandeep (Roll No: CS2210)**, hereby affirm that this report titled **"Template Driven Rainbow Coloring of Graphs"**, submitted to the Indian Statistical Institute, Kolkata, as part of the requirements for the **Master of Technology in Computer Science**, is my original work. This research was conducted under the guidance of **Dr. Mathew C. Francis** and has not been previously submitted for the award of any degree or diploma at any other institution or university. I have diligently followed academic ethics and integrity throughout this project. All external sources of information, statements, or results have been appropriately cited and acknowledged.



Bhuvandeep

Kolkata - 700 108
June 12, 2024

Certificate

This is to certify that the project report titled *"Template Driven Rainbow Coloring of Graphs"*, submitted by Bhuvandeep (Roll No: CS2210) to the Indian Statistical Institute, Kolkata, as a partial fulfillment of the requirements for the Master of Technology degree in Computer Science, has been conducted under my supervision. This work is original and has not been previously submitted for any degree or diploma at any other institution or university.

Dr. Mathew C. Francis

Chennai - 600029
June 12, 2024
Project Supervisor

Contents

Acknowledgement

I want to thank my supervisor, Dr. Mathew Francis, for teaching me to navigate my own path through this thesis and in life. I could not have wished for a friendlier supervisor. I want to express my gratitude to Prof. Mandar Mitra and Dr. Sumana Ghosh for always believing in me even when I was not able to do so. I also want to thank Dr. Debrup Chakraborty and Prof. Sourav Chakraborty for their enjoyable and inspiring classes.

1

Introduction

Graph theory is a branch of mathematics that deals with the study of graphs, which are mathematical structures used to represent relationships between objects. In simple terms, a graph is a collection of nodes, also known as vertices, connected by edges. These nodes and edges can represent a wide range of entities and connections, such as cities and roads, computers and networks, or even individuals and their friendships.

We now define some basic graph-theoretic notation and terminology that we shall need. For a graph G, we denote by $V(G)$ the set of vertices of G, and by $E(G)$ the set of edges of G. The graphs that we study shall be *undirected*, and so the edges of the graph are represented by unordered pairs of vertices; for example for $u, v \in V(G)$, we denote the fact that there is an edge between u and v in G by writing $uv \in E(G)$. The graphs that we study shall also be *simple* and *loopless*, meaning that for every edge $uv \in E(G)$, we have $u \neq v$, and there is at most one edge between a given pair of vertices. When $uv \in E(G)$, we say that u is a neighbour of v (and vice versa), or that u and v are adjacent. The edge uv is said to be incident on both u and v, which are called the endpoints of the edge. Two edges that have a common endpoint are said to be adjacent edges. The set of neighbours of a vertex u, also called the *neighbourhood* of u, is denoted by $N(u)$; i.e. $N(u) = \{v \in V(G) \colon uv \in E(G)\}$. The degree of a vertex is $deg(u) = |N(u)|$. The maximum degree of G, denoted by $\Delta(G)$, is the maximum among the degrees of the vertices of G; i.e. $\Delta(G) = \max\{deg(u) \colon u \in V(G)\}$. A set $S \subseteq V(G)$ is said to be an independent set in G if no pair of vertices in S are adjacent in G. On the other hand, if every pair of vertices in S are adjacent, then S is called a clique in G. A graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ is said to be a subgraph of G. A graph H such that $V(H) \subseteq V(G)$ and $E(H) = \{uv \in E(G) \colon u, v \in V(H)\}$ is said to be an induced subgraph of G. The subgraph induced in G by a set $S \subseteq V(G)$ is the induced subgraph of G having vertex set S. We denote by $G - u$, where $u \in V(G)$, the graph obtained from G by removing the vertex u; i.e. it is the subgraph induced in G by $V(G) \setminus \{u\}$. By $G - uv$, where $uv \in E(G)$, we denote the subgraph of G obtained by removing the edge uv from G. Two graphs G and H are said to be isomorphic to each other if there exists a bijection f from $V(G)$ to $V(H)$ such that for $u, v \in V(G)$, $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$.

We now define some common kinds of graphs. A *complete graph* is a graph in which every pair of vertices is adjacent. A complete graph on n vertices is denoted by K_n. A *path graph* on n vertices (or simply a *path* on n vertices), denoted by P_n, is any graph isomorphic to the graph having vertex set $\{v_1, v_2, \ldots, v_n\}$ and edge set $\{v_1v_2, v_2v_3, \ldots, v_{n-1}v_n\}$. The "length" of a path is the number of edges in it; i.e. the length of a P_n is equal to $n-1$. A *cycle graph* on n vertices (or simply a *cycle* on n vertices), denoted by C_n, is the graph obtained by adding an edge between the two vertices having degree 1 in P_n. The "length" of a cycle is the number of edges in it, which is the same as the number of vertices in it, and therefore the length of a C_n is equal to n. Given a graph G, a subgraph of G that is isomorphic to a path graph is said to be a "path in G", and a subgraph of G that is isomorphic to a cycle graph is said to be a "cycle in G". Graphs that do not contain cycles are called *acyclic graphs*, or *forests*. A graph is said to be *connected* if for every pair of vertices in the graph, there is a path in the graph that contains both of them. A connected acyclic graph is called a *tree*. A maximal connected subgraph of a graph G is called a *connected component* (or simply *component*) of G.

1.1 Graph coloring

Graph coloring is a pivotal concept in graph theory, with numerous practical applications. It involves assigning colors to different elements of a graph while adhering to specific constraints. The most commonly studied types of graph colorings can be broadly categorized into vertex colorings, where the vertices of the graph are assigned colors, and edge colorings, in which it is the edges that are assigned colors. In most kinds of colorings, the quantity of interest is the minimum number of colors required so that a mapping from the vertices or edges to these colors exists that also satisfies the required constraints.

Many variants of graph colorings appear in the literature. In a *proper vertex coloring*, colors are assigned to the vertices of a graph such that no two adjacent vertices share the same color. Thus, the vertices that get the same color will form an independent set in the graph. The minimum number of colors required in such a coloring of a graph is called the *chromatic number* of that graph (see Section 2.2 for more details). Proper vertex coloring is one of the classical problems that is almost as old as graph theory itself. The famous Four Color Theorem, which was proved in 1977 by Appel and Haken [2, 3], after remaining an open problem for over a century, states that every "planar graph" has a proper vertex coloring using at most four colors (see Chapter 2 for the definition of planar graphs). Observe that in a proper vertex coloring, the vertices that get a common color form an independent set in the graph; or in other words, the "color classes" (sets of vertices having the same color) of a proper vertex coloring are independent sets. Thus the chromatic number of a graph is the minimum number of independents sets into which the vertex set of a graph can be partitioned.

Proper edge coloring, on the other hand, involves assigning colors to the edges of a graph, ensuring that edges incident on the same vertex are assigned different colors. The minimum number of colors required in a proper edge coloring of a graph is called its *chromatic index*. A classical theorem of Vizing states that the chromatic index of any graph G having maximum degree Δ is either Δ or $\Delta+1$ [23]. Clearly, the edges that get one color in a proper edge coloring form a matching in the graph, and hence the chromatic index of a graph is the minimum number of matchings into which the edge set of the graph can be partitioned.

There are other variants of coloring problems in which the constraint to be satisfied is not defined as a property that has to be satisfied by a single color class, but rather is defined as a property that has to be satisfied by several color classes taken together. For example, in acyclic vertex coloring [1], we require that the union of any two color classes induces a subgraph that is acyclic, in addition to the property that each color class is an independent set; or in other words, it is a proper vertex coloring with the additional restriction that there must not be a cycle in the graph that contains only vertices of two different colors. Other variants of coloring focus on how many colors are required when the graph to be colored is not known fully in advance; i.e. parts of the graph get revealed only after some colors are already assigned to some vertices (and these cannot be changed later). An example of this kind of coloring problem is the problem of *online coloring* [22], where the vertices to be colored arrive one by one and a color has to be assigned to each vertex as they arrive. There is also a variant of coloring called *total coloring* [8], in which both the vertices and edges are assigned colors. The requirement here is that no adjacent vertices, adjacent edges or incident edges and vertices get the same color. Please refer [14] for a survey of various kinds of graph coloring problems.

The main algorithmic question related to each coloring problem is the computation of a coloring of the required type using the minimum number of colors for a given input graph. Unfortunately, this problem turns out to be NP-hard for most variants of graph colorings. In fact, the problem of deciding whether an input graph has a proper vertex coloring using at most 3 colors itself is NP-complete [10]. Moreover, even for those coloring problems for which tight upper bounds are known for the minimum number of colors required, the problem of computing the exact value can be NP-hard. For example, deciding whether an input planar graph has a coloring using at most 3 colors is NP-complete [10]. Similarly, deciding which one of Δ or $\Delta+1$ is the chromatic index of a graph having maximum degree Δ is NP-complete [15].

In this thesis, our primary focus shall be on a variant of vertex coloring called *template-driven rainbow coloring* and the algorithmic question of computing such a coloring using the minimum possible number of colors. A "template" here is just a fixed graph. The goal in template-driven rainbow coloring, given a template T (or a set of templates), is to color the vertices of an input graph G with the minimum number of colors so that every subgraph H of G that is isomorphic to T (or some graph in the given set of templates) is "rainbow-colored": meaning that no two vertices in H have the same color. We give a full formal definition of this coloring and related notation in Section 2.3.

Since the template-driven rainbow coloring problem (for arbitrary templates) is also NP-hard for general graphs (see Proposition 2), we focus on "special classes" of graphs. This is a common way of dealing with the intractability of graph problems. Many problems that we would like to have solved in practice may be NP-hard for general graphs; i.e., the existence of a polynomial-time algorithm that can solve the problem on *all* input graphs would imply that $P = NP$ [11], and therefore such an algorithm is unlikely to exist. But many a time, the graphs that arise from a specific use-case have some special structure that can be exploited to solve the problem in polynomial time. That is, there can be polynomial time algorithms to solve the problem that are guaranteed to produce the correct answer only when the input graph belongs to some special graph class (a class of graphs that satisfies some special property). We discuss this notion in detail in Section 2.1.

1.2 Motivation

Efficient and scalable implementation of parallel algorithms on multiprocessor architectures with multiple memory banks depends on how fast the items in the underlying data structure can be accessed in parallel. To simultaneously access the data items required for a computation by multiple processors, the pertinent data must reside in different memory banks [7]. This problem of conflict-free access to parallel memory systems can be formulated as rainbow coloring of templates corresponding to the items to be accessed in a host graph representing the data structure.

1.3 Notations

The notations used in this thesis are summarized in Table 1.1.

G	A simple graph
$V(G)$	Vertex set of the graph G
$E(G)$	Edge set of the graph G
$deg(v)$	Degree of a vertex v
$\chi(G)$	Chromatic number of the graph G
C_k	Cycle of length k
$\chi_T^R(G)$	Minimum number of colors in a TR-coloring with template T in G
$\chi_T^{RB}(G)$	Minimum number of colors in a TRB-coloring with template T in G
K_n	Complete graph on n vertices
$\Delta(G)$	Maximum degree in G
$G - uv$	Graph G with edge uv removed
$G - u$	Graph G with the vertex u removed

Table 1.1: Notations

1.4 Our contributions

We prove that template driven rainbow coloring with a cycle on 3 vertices as template is NP-complete even for planar graphs. We show that the problem is however polynomial time solvable for chordal graphs. We then show that if the template is a cycle on 4 vertices, then the problem is NP-hard for chordal graphs, in fact, even for split graphs. Finally, we show that if the template is a cycle on 5 or 6 vertices, the problem is in fact polynomial-time solvable for split graphs.

1.5 Thesis Outline

We start with formally defining template driven rainbow coloring in Chapter 2. In Chapter 3, we give some known results on template driven rainbow coloring of certain graph classes and a small literature survey. In Chapter 4, we state and prove our original results. We conclude by identifying some open problems in Chapter 5.

2

Preliminaries

2.1 Special classes of graphs

For many practical applications in which problems on graph need to be solved algorithmically, the graphs that arise have some specific structural properties. A set of graphs that satisfy some particular property is called a "graph class". As mentioned before, many problems that are NP-hard for graphs in general admit polynomial time algorithms when the input graphs are restricted to be from special graph classes. For example, the class of *planar graphs* consists of those graphs that can be drawn on the plane, with points representing the vertices, and curves between two points representing an edge between them, in such a way that no two edges cross. It follows trivially from the Four Color Theorem that the problem of deciding if an input graph is k-colorable is polynomial time solvable for planar graphs for every fixed value of $k \neq 3$.

We shall now take a look at some graph classes that will be relevant to this study. We do not give individual citations for most of the details in the subsequent sections; the interested reader may please refer [13] for more information regarding the graph classes mentioned.

2.1.1 Chordal graphs

Chordal graphs are the graphs in which every cycle of length greater than 3 has a chord, i.e. an edge joining non-consecutive vertices of the cycle. Chordal graphs are a very important family of graphs due to their diverse mathematical properties and various applications. Every chordal graph contains a *simplicial vertex* — a vertex whose neighbourhood is a clique. Since every induced subgraph of a chordal is also chordal, this means that the vertices of a chordal graph can be removed sequentially, so as to finally eliminate all vertices, in such a way that at every step the vertex being removed is a simplicial vertex. An ordering of vertices that corresponds to such a sequence is called a *perfect elimination ordering*. The problem of deciding whether an input graph is chordal, and if yes, generating a perfect elimination ordering of it, is polynomial-time solvable. Several problems that are NP-hard on general graphs become polynomial-time solvable when restricted to chordal graphs. For example, the problem of finding a *maximum cardinality*

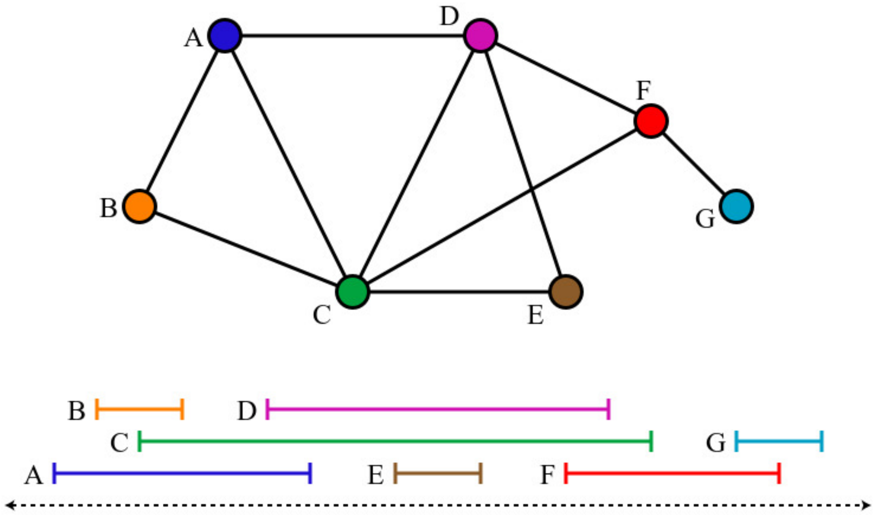


Figure 2.1: Interval graph (Image courtesy - Wikipedia)

clique is NP-hard on general graphs but can be solved in polynomial time on chordal graphs using the perfect elimination ordering. Determining the *chromatic number* (the minimum number of colors needed to color the vertices such that no two adjacent vertices share the same color) is also NP-hard on general graphs but can be found in polynomial time on chordal graphs, as the chromatic number of a chordal graph is equal to its maximum clique size.

We state below a folklore property of chordal graphs that we shall need.

Definition 2.1.1. An edge uv of a graph G is called a *bridge* if u and v are in different connected components of $G - uv$.

It is easy to see that an edge is a bridge if and only if it does not lie on any cycle. The following proposition gives a formal proof for this statement (a similar proof also appears in [24]).

Proposition 1. *Every edge of a chordal graph is a bridge or is part of a triangle.*

Proof. Consider a connected graph G which is also chordal. Let $e = (u, v) \in E(G)$. If e lies on no cycle, then the only path between u and v in G is the path consisting of the single edge (u, v). So, removing the edge (u, v) will make the graph G disconnected. Hence, (u, v) is a bridge. It remains to show that if (u, v) is part of a cycle then it is part of a triangle. Suppose that the edge (u, v) lies on a cycle. Consider a shortest cycle C that contains the edge (u, v). Let $C = v_0v_1 \ldots v_{k-1}v_0$, where $v_0 = u$ and $v_{k-1} = v$. If $k = 3$, then C is a triangle, and we are done. So we assume that $k > 3$. Then C is a cycle of length at least 4, and since G is chordal the cycle C has some chord, say v_iv_j, where $i, j \in \{0, 1, \ldots, k-1\}$. We assume without loss of generality that $i < j$. Then $v_0v_1 \ldots v_iv_jv_{j+1}v_{k-1}v_0$ is a cycle containing the edge (u, v) that is shorter than C. This contradicts the choice of C. This completes the proof. $\square$

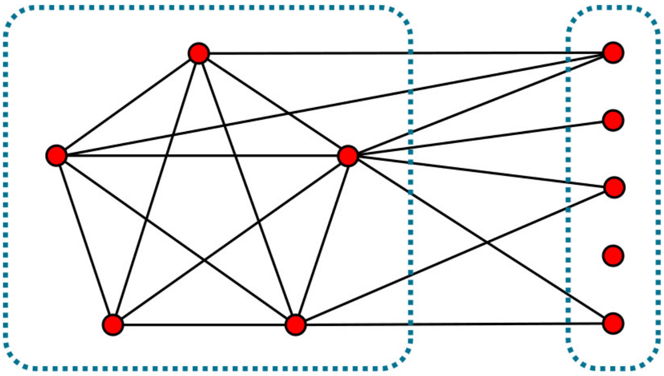


Figure 2.2: Split graph (Image courtesy - Wikipedia)

2.1.2 Interval graphs

A *closed interval* on the real line, denoted as $[a, b]$, where $a, b \in \mathbb{R}$ and $a \leq b$, is the set $\{x \in \mathbb{R} \colon a \leq x \leq b\}$. If $I = [a, b]$, we say that a is the left end point of I and b is the right end point of I. A graph G with vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$ is called an interval graph if there exists intervals $I_1, I_2, ..., I_n$ such that for $v_i, v_j \in V(G)$, where $1 \leq i < j \leq n$, $v_iv_j \in E(G)$ if and only if the intervals I_i and I_j intersect [12]. When this condition is satisfied, the collection of intervals $I_1, I_2, \ldots, I_n$ is said to be an *interval representation* of the graph G.

These graphs are used in scheduling problems in which a subset of tasks to be performed at non-overlapping times have to be selected. Graphs that have an interval representation in which each interval is of unit length are called unit interval graphs. Unit interval graphs are also called *proper interval graphs*. It is easy to see that every induced subgraph of an interval graph is also an interval graph, and similarly, every induced subgraph of a proper interval graph is a proper interval graph. Since the graph C_k is not an interval graph for any $k \geq 4$, all interval graphs are chordal graphs; or in other words, the class of interval graphs forms a subclass of the class of chordal graphs. This means that all the problems that are polynomial time solvable on chordal graphs are also polynomial time solvable on interval graphs. Moreover, the problem of determining, given an input graph G, whether G is an interval graph, and generating an interval representation for it in case it is, is also polynomial time solvable. A *left end ordering* of an interval graph G is an ordering of vertices of G according to the left end points of the corresponding intervals in an interval representation of G (assuming that the endpoints of the intervals in the representation are distinct; this is an assumption that can always be made, since every interval graph has such an interval representation).

2.1.3 Split graphs

Split graphs are graphs whose vertex set can be partitioned into a clique and an independent set. A chordal graph G is a split graph if and only if its complement, $\overline{G}$ is also chordal [9]. Thus split graphs form a subclass of chordal graphs, just like interval graphs do. But the classes of split graphs and interval graphs are incomparable — there are split graphs that are not interval graphs, and interval graphs that are not split graphs.

2.1.4 Perfect graphs

A graph G is called perfect if every induced subgraph of G has same clique number and chromatic number. It is known that the minimum number of colors required to color a perfect graph, and so also the size of a maximum cardinality clique, can be found in polynomial time. The famous Strong Perfect Graph Theorem of Chudnovsky, Robertson, Seymour and Thomas [6], which was earlier a conjecture of Berge, states that a graph G is perfect if and only if neither G nor its complement (the complement $\overline{G}$ of G is the graph having $V(\overline{G}) = V(G)$ and $E(\bar{(G)}) = \{uv \colon u, v \in V(G), u \neq v$ and $uv \notin E(G)\}$) contains an induced subgraph isomorphic to an odd cycle of length at least 5. This implies that the complement of a perfect graph is also perfect — in fact, this is the Weak Perfect Graph Theorem, that was proven in by Lovász [18, 17] (see Figure 2.3 for an example of a perfect graph). The class of chordal graphs is a subclass of the class of perfect graphs. Many natural problems on graphs like the problem of computing a maximum cardinality clique, maximum cardinality independent set, and proper vertex coloring using minimum number of colors are polynomial time solvable on perfect graphs [4].

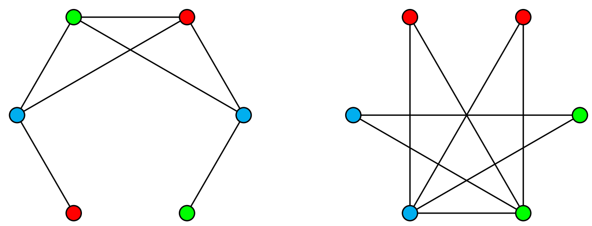


Figure 2.3: A perfect graph and its complement. Optimal proper vertex colorings are also shown.

2.2 Graph coloring

Given a set C, called the set of "colors", a function which assigns a value in C to each vertex of a graph G is called a *vertex-coloring*, or *coloring* for short, of G using colors from C. A k-*coloring* of a graph G is a map $f : V(G) \to S$ where $|S| \leq k$. As mentioned before, the elements of S are called colors, and the vertices of one color form a color class — the color class corresponding to a color $s \in S$ is $f^{-1}(s)$. A *proper* k-coloring of a graph G is a k-coloring of G with the property that $f(u) \neq f(v)$ whenever uv is an edge of G. A graph that has a proper k-coloring is said to be k-*colorable*. The chromatic number of G, denoted $\chi(G)$, is the smallest integer k such that G has a proper k-coloring. Note that any proper $k-1$ coloring is also a proper k-coloring. A graph G is k-*chromatic* if it is k-colorable but not $k-1$ colorable, or in other words, if $\chi(G) = k$.

2.2.1 Some facts about chromatic number of graphs

- A graph G is 1-chromatic if and only if $E(G) = \emptyset$; i.e. if G is an empty graph.
- The cycle C_n is 3-chromatic if n is odd and 2-chromatic otherwise.
- The complete graph K_n is n-chromatic.
- For any graph G, $\chi(G) \leq |V(G)|$. In fact, for any graph G, $\chi(G) \leq \Delta(G)+1$, where $\Delta(G)$ is the maximum degree of G. Further, Brooks' Theorem [20] states that if a connected graph G is neither an odd cycle nor a complete graph, Then $\chi(G) \leq \Delta(G)$.
- If H is any subgraph of G, then we have $\chi(H) \leq \chi(G)$. Thus, $\chi(G) \geq \omega(G)$, where $\omega(G)$ is the size of a clique of maximum cardinality in G. There exists graphs G such that $\chi(G)$ is much larger than $\omega(G)$. The Mycielski construction [19], for example, shows how, given any integer k, one can construct a k-chromatic graph with no clique of size more than 2.

2.2.2 Algorithms for coloring

The algorithmic problem of deciding whether a graph G is k-colorable is defined as given below.

COLORABILITY Given a graph G and an integer k, decide if G has a proper k-coloring.

When the integer k is part of the input, the corresponding problem is called k-COLORABILITY, defined as given below.

k-COLORABILITY Given a graph G, decide if G has a proper k-coloring.

The problem COLORABILITY is polynomial time solvable for special classes of graphs like chordal graphs, interval graphs, and in general for the class of perfect graphs. The problem 2-COLORABILITY is polynomial-time solvable for the class of all graphs (since it amounts to checking whether the input graph is bipartite [21]), but 3-COLORABILITY is NP-complete even for planar graphs having maximum degree at most 4 [10].

2.3 Template driven rainbow coloring

A graph G with a coloring c is said to be *rainbow colored* if all vertices receive different colors; i.e. $c(u) \neq c(v)$ for any two distinct vertices $u, v \in V(G)$. Given a graph G with a coloring c, a subgraph H of G is said to be rainbow colored if H is rainbow colored with respect to c; i.e. no two vertices in H have the same color. A *template-driven rainbow coloring* of G with respect to a graph H, called the "template", is a coloring of G such that all subgraphs of G isomorphic to H are rainbow-colored. We shall refer to a template-driven rainbow coloring of a graph G with respect to a template H as a *TR-H coloring* of G. The minimum number of colors required in any TR-H coloring of a graph G is denoted by $\chi^R_H(G)$.

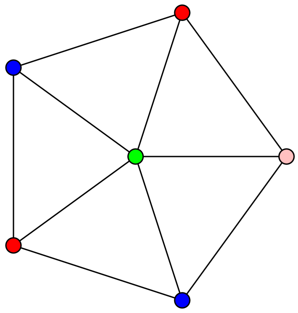


Figure 2.4: TR coloring with C_3 as template

TR-COLORABILITY
Given a graph G, a template H and an integer k, decide if G has a TR-H coloring using at most k colors.

When the template H is part of the problem definition (and not part of the input), the problem is called TR-H COLORABILITY.

TR-H COLORABILITY
Given a graph G and an integer k, decide if G has a TR-H coloring using at most k colors.

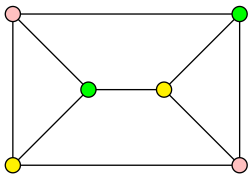


Figure 2.5: Proper coloring (minimum 3 colors required)

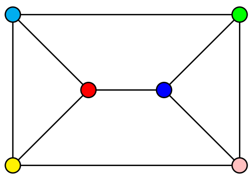


Figure 2.6: TR coloring with C_4 as template (minimum 6 colors required)

An ideal TR-H coloring should be optimal, balanced, and explicit. Optimal coloring uses the fewest colors possible, ensuring efficiency. Balanced coloring ensures that vertices are evenly distributed among color classes, preventing any memory bank (corresponding to a color) from being overloaded. This balance can be maintained by ensuring the difference between the largest and smallest color classes is minimal. Explicit coloring is crucial for practical applications, allowing the color of any vertex to be computed quickly using a simple formula. This enables efficient data management by ensuring the color assigned to a vertex can be determined in constant time. In summary, effective TR-coloring algorithms aim to achieve optimality, balance, and explicitness, thus enhancing performance in parallel computing environments.

3

Literature review and related work

Chandran et al. [5] gave an algorithm for optimal TR-coloring of proper interval graphs using cycle templates. Given any interval graph, they take its left end ordering, say $\sigma = (v_1, v_2, \ldots, v_n)$ and then define $pivot(v_i)$ as the vertex v_j with highest index j for which there is a cycle of length k in G containing both v_i and v_j. They define the maximum distance from any v_i to its pivot as k-span of G, denoted as $p_k(G)$. Their algorithm works in the following manner. Given a proper interval graph H, it finds the 2-connected components of G of H and works on each component separately. It finds the left end ordering of each 2-connected proper interval graph separately. It then computes $pivot(v)$ for all v in G. Subsequently, it finds $p_k(G)$ and then compares the k-span of all components to find the maximum one, say x. It then colors a vertex v_i with color $i \mod x$. They prove that this is a valid TR-C_k coloring of the input unit interval graph G. Their algorithm for computing a TR-C_k coloring of unit interval graphs has a time complexity of $O(k \cdot |V| + |E|)$. Their algorithm gives an optimal, balanced and explicit TR-C_k coloring of the input graph. For fixed k, their algorithm runs in $O(|V|)$ time for sparse graphs.

Hong Yang et al [25] explored template driven rainbow coloring with a path of length 3 as a template. They proved that in a graph G with n vertices and m edges, if there is a vertex v with degree $n-1$, then minimum number of colors required for template driven rainbow coloring of G with respect to template P_3 is same as chromatic number of G. They have found optimal coloring number for different classes of graphs including but not restricted to ladder graphs, star graphs, wheel graphs, cliques and prism graphs.

Chandran et al. [5] remark that a polynomial time algorithm for TR-C_k COLORABILITY for interval graphs is likely to considerably more complicated, if at all one exists.

We study the complexity of TR-C_k COLORABILITY for the class of chordal graphs, which contains the class of interval graphs.

4

Results

The following observation is a direct consequence of the fact that template-driven rainbow coloring with template P_2 is the same as proper coloring.

Proposition 2. *TR-P_2 COLORABILITY is NP-complete for general graphs.*

Proof. We show a reduction from k-COLORABILITY for general graphs to TR-P_2 COLORABILITY for general graphs. Consider any graph $G(V, E)$. Since a TR-P_2-coloring of G using k colors is the same as a proper k-coloring of G, we can decide if G has a proper k-coloring by checking if G has a TR-P_2-coloring using at most k colors. As k-COLORABILITY is NP-hard, it follows that TR-P_2 COLORABILITY is NP-hard for general graphs. It is easy to see that TR-P_2 COLORABILITY is in NP, and therefore the problem is NP-complete. □

It follows from the above proposition that TR COLORABILITY is also NP-complete for general graphs. Note that any proper coloring of a graph will give different colors to the vertices contained in a triangle, hence it is definitely a template driven rainbow coloring with C_3 as template. Thus, any proper coloring of G is also a TR-C_3 coloring of G but it may not be an optimal one. That is, there may exist a template driven rainbow coloring of G using less colors less the chromatic number of G.

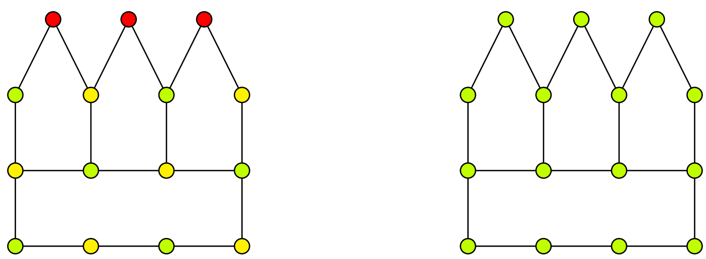


Figure 4.1: Proper coloring of a graph and a TR coloring with C_3 as template

Theorem 3. *TR-C_3 $\textsc{Colorability}$ is NP-complete for planar graphs.*

Proof. Since the it is clear that TR-C_3 $\textsc{Colorability}$ is in NP, we shall only show the proof for NP-hardness of the problem. We show a reduction from $\textsc{Colorability}$, which, as we noted before, is known to be NP-complete for planar graphs. it is easy to see that the problem $\textsc{Colorability}$ is NP-hard even for planar graphs having chromatic number at least 3. [16] Let $G(V, E)$ be a planar graph for which $\chi(G) \geq 3$. Construct a new graph G' as follows: Take the graph G and for every edge e of G, add a new vertex v_e which is adjacent only to the endpoints of e. Note that $V(G') = V(G) \cup \{v_e \colon e \in E(G)\}$. Clearly, G' is a planar graph (since we can construct a planar embedding of G' by taking a planar embedding of G and placing, for each edge e of G, the point corresponding to the vertex v_e very close to the curve representing the edge e).

We claim that for any positive integer k, $\chi^R_{C_3}(G') \leq k$ if and only if $\chi(G) \leq k$. Suppose that $\chi(G) \leq k$. Note that $k \geq 3$ since we assumed that $\chi(G) \geq 3$. Let c be a k-coloring of G. We construct a coloring c' of G' using the same set of k colors as c, as follows. For each $u \in V(G)$, we let $c'(u) = c(u)$. For each vertex $v_e \in V(G')$, where $e = uv \in E(G)$, we let $c'(v_e)$ be a color from the color set of c that is different from $c'(u)$ and $c'(v)$. Note that we can always do this since $k \geq 3$. We claim that c' is a TR-C_3 coloring of G'. Consider any C_3 having vertices u, v, w in G'. If $x, y, z \in V(G)$, then each has a distinct color in c, and therefore also in c'. If one of them, say x, is in $V(G') \setminus V(G)$, then it is must be the case that $y, z \in V(G)$, $yz \in E(G)$ and $x = v_{yz}$. Then we have by the construction of c' that $c'(x)$, $c'(y)$, and $c'(z)$ are all distinct. Thus c' is a TR-C_3 coloring of G', which implies that $\chi^R_{C_3}(G') \leq k$. Conversely, suppose that $\chi^R_{C_3}(G') \leq k$. Then we have a TR-C_3 coloring of G', say c', with at most k colors. Let c be the coloring of G obtained by restricting c' to $V(G)$; i.e. we define c to be the coloring of G using the same set of colors as c' where $c(u) = c'(u)$ for all $u \in V(G)$. We claim that c is a proper k-coloring of G. Consider any edge $xy \in E(G)$. We have by the construction of G' that x, y, v_{xy} form a C_3 in G'. As c' is a TR-C_3 coloring of G, we know that $c'(x)$, $c'(y)$, and $c'(v_{xy})$ are all distinct. Thus $c'(x) \neq c'(y)$, and therefore $c(x) \neq c(y)$. This shows that c is a proper k-coloring of G. We thus have that $\chi(G) \leq k$.

It now follows that it can be decided in polynomial time whether G is k-colorable if it can be decided in polynomial time whether G' has a TR-C_3 coloring using at most k colors. Since it is easy to see that G' can be constructed from G in polynomial time, we have a polynomial-time reduction from k-$\textsc{colorability}$ on planar graphs to TR-C_3 $\textsc{colorability}$ on planar graphs. This completes the proof. $\square$

TR Coloring for Chordal Graphs

Theorem 4. *For any connected chordal graph G, $\chi^R_{C_3}(G)$ is 1 if G does not contain a triangle, and is $\chi(G)$ otherwise.*

Proof. If G does not contain any triangle, then clearly $\chi^R_{C_3}(G) = 1$. Let us assume that G contains at least one triangle. Then $\chi(G) \geq \chi^R_{C_3}(G)$ because any proper coloring of G is also a

TR-C_3 coloring of G but it may not be an optimal one. Given a TR-C_3 coloring c of G, we say that a bridge (u, v) of G is "bad" in c if $c(u) = c(v)$.

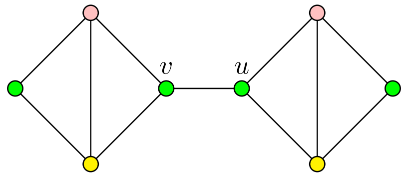


Figure 4.2: A TR-C_3 coloring of a chordal graph

Let c be an optimal TR-C_3 coloring of G having least number of bad bridges. Suppose that c is not a proper coloring of G. Then there is an edge uv in G such that its end vertices u and v receive the same color. Then clearly uv cannot be part of a triangle because then u and v cannot get same color in a TR-C_3 coloring of G. Hence, we have from Proposition 1 that uv is a bridge. Remove the edge uv from G to obtain two connected components, say G_u and G_v containing u and v respectively. Since G contains at least one triangle, we know that at least one of G_u or G_v contains a C_3. Without loss of generality, let G_u be that component. Let us assume without loss of generality that the colors used by c in $V(G_u)$ are $\{0, 1, \ldots, k-1\}$. Since G_u contains a triangle, it is clear that $k \geq 3$. The idea now is to "shuffle" the colors in G_u so that u and v get different colors.

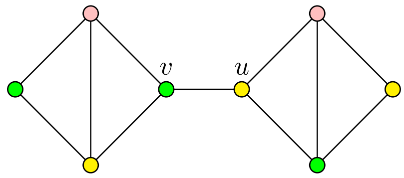


Figure 4.3: Proper coloring after shuffling the colors

Formally, we define a new TR-C_3 coloring c' of G as follows. Let $c'(x) = c(x)$ for all $x \in V(G_v)$ and let $c'(x) = c(x)+1 \mod k$ for all $x \in V(G_u)$. Note that c' restricted to G_u and c' restricted to G_v are both TR-C_3 colorings (since the color classes of c and c' are the same in both G_u and G_v). Since any triangle has to lie entirely in G_u or entirely in G_v, we now have that c' is a TR-C_3 coloring of G. Since $c(u) \neq c'(u)$ and $c(u) = c(v) = c'(v)$, the bridge uv is not bad in c'. Consider any bridge ab of G such that $c(a) \neq c(b)$. Clearly, $ab \neq uv$ and therefore both a and b have to lie in either G_u or G_v. As noted before, the color classes of c and c' are the same when restricted to either G_u or G_v. This means that $c'(a) \neq c'(b)$. Thus c' is a TR-C_3 coloring of G having lesser number of bad bridges than c, which contradicts our choice of c. We can thus conclude that c is a proper coloring of G. This implies that $\chi^R_{C_3}(G) \geq \chi(G)$, which further implies that $\chi^R_{C_3}(G) = \chi(G)$. $\square$

From Proposition 1, it is obvious that a connected chordal graph contains a triangle if and only if it is not a tree. We thus have the following corollary.

Corollary 5. *For any connected chordal graph G, $\chi^R_{C_3}(G) = 1$ if G is a tree, and $\chi_{C_3}(G) = \chi(G)$ otherwise.*

Note that if the input chordal graph G is disconnected, then each connected component can be considered separately, and so we have $\chi^R_{C_3}(G) = \max\{\chi^R_{C_3}(H)\colon H$ is a connected component of $G\}$. Since given a connected chordal graph G on n vertices and m edges, detecting whether it is a tree and computing is chromatic number can both be done in $O(n+m)$ time [13], we have the following.

Corollary 6. *TR-C_3 COLORABILITY is linear-time solvable for chordal graphs.*

Theorem 7. *TR-C_4 COLORABILITY is NP-Hard for split graphs.*

Proof. We shall show a polynomial-time reduction from k-COLORABILITY for general graphs to TR-C_4 COLORABILITY for split graphs. Let G be any graph with n vertices and m edges where m is at least 2. We shall assume that G contains no isolated vertices as k-COLORABILITY can be easily seen to be NP-hard even for graphs with no isolated vertices. Construct a new split graph $H(K, I)$ (where K is the clique side of H and I is the independent set side) in the following manner. Let $I = \{u'\colon u \in V(G)\}$ and let $K = \bigcup_{e=(u,v)\in E(G)}\{x_e, y_e\}$. Thus $|K| = 2m$ and $|I| = n$. Define $E(H) = \bigcup_{e=(u,v)\in E(G)}\{(x_e, u'),\ (x_e, v'), (y_e, u'), (y_e, v')\}$. We claim that $\chi(G) \leq k$ if and only if $\chi^R_{C_4}(H) \leq k + 2m$.

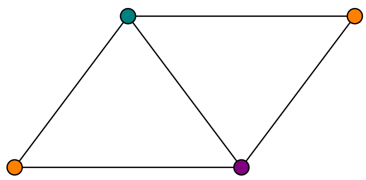


Figure 4.4: Original graph

Suppose that $\chi(G) \leq k$. Then there exists a proper coloring c of G using the colors $\{1, 2, \ldots, k\}$. Define $c_H : V(H) \to \{1, 2, \ldots, k + 2m\}$ as follows. Recall that $V(H) = K \cup I$. For each $u' \in I$, we define $c_H(u') = c(u)$. For each vertex z in K, we choose as $c_H(z)$ a unique color in $\{k+1, k+2, \ldots, k+2m\}$. We claim that c_H is a TR-C_4 coloring of H. Consider any cycle of length 4 in H. If all vertices of the cycle are from K, then in c_H, each of them has a unique color in $\{k+1, k+2, \ldots, k+2m\}$, and hence we are done. If exactly one vertex from the cycle is in I, then that vertex has a color in $\{1, 2, \ldots, k\}$ and the remaining three vertices have unique colors from $\{k+1, k+2, \ldots, k+2m\}$, and again we are done. Finally, let us consider the case when there are exactly two vertices from I in the cycle (note that there cannot be three or more vertices from I in the cycle), say u' and v'. Let the cycle be $u'wv'zu'$ where $w, z \in K$. By the way we constructed H, it is clear that $w = x_e$ for some $e = (a, b) \in E(G)$ and that $N_H(w) \cap I = \{a', b'\}$. Since $u'wv'zu'$ is a cycle, we now have that $\{u', v'\} = \{a', b'\}$. Thus we can conclude that $uv \in E(G)$. Then we have $c(u) \neq c(v)$, and therefore $c_H(u') \neq c_H(v')$. Since $c_H(u'), c_H(v') \in \{1, 2, \ldots, k\}$ and $c_H(w)$ and $c_H(z)$ are two distinct colors from $\{k+1, k+2, \ldots, k+2m\}$, we again have that each vertex of the

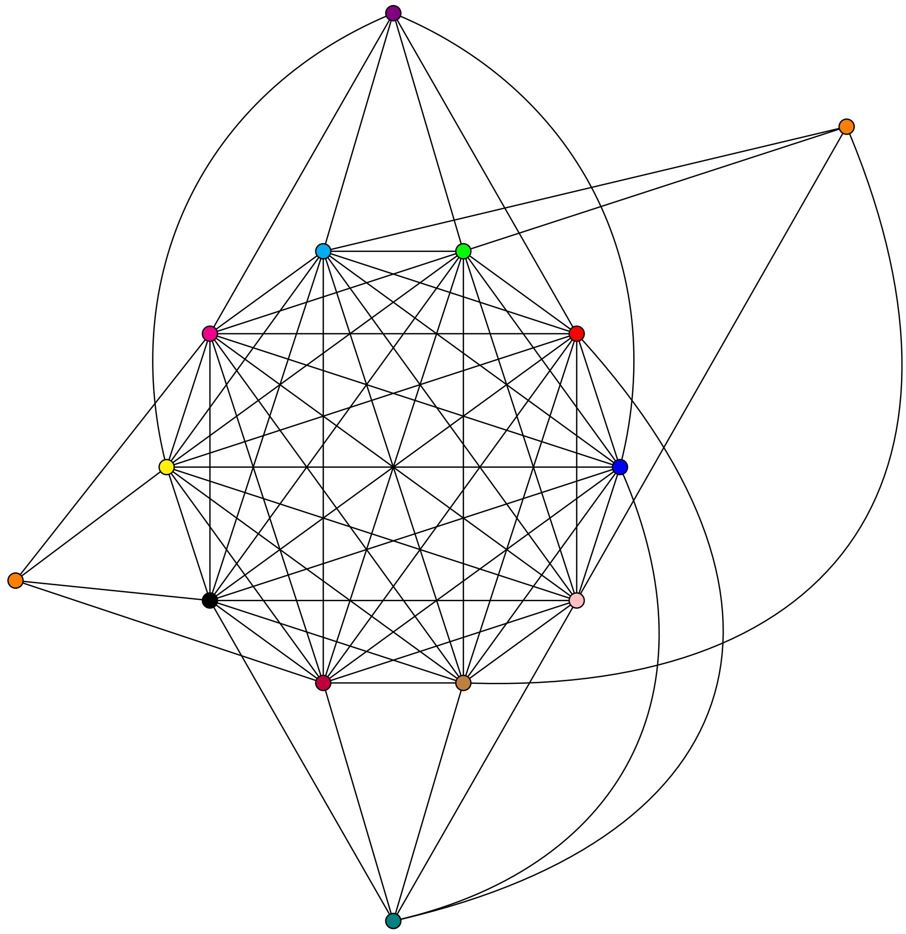


Figure 4.5: Constructing split graph from the original graph

cycle has a different color. This shows that c_H is a TR-C_4 coloring of H. Next, we shall show if $\chi^R_{C_4}(H) \leq k + 2m$, then $\chi(G) \leq k$. Suppose that there exists a TR-C_4 coloring c_H of H using colors $\{1, 2, \ldots, k + 2m\}$. Clearly, $|K| \geq 4$, since $m \geq 2$. Since K is a clique, any four vertices of K lie on a cycle of length four, and therefore each of the $2m$ vertices of K must have gotten a different color in c_H. We assume without loss of generality that these are the colors $\{k+1, k+2, \ldots, k+2m\}$. We claim for each vertex $u' \in I$, $c_H(u) \notin \{k+1, k+2, \ldots, k+2m\}$. Suppose that there is a vertex $u' \in I$ such that $c_H(u') \in \{1, 2, \ldots, 2m\}$. Let w be the vertex in K such that $c_H(w) = c_H(u')$. Since G contains no isolated vertices, there is some edge $e \in E(G)$ incident on u in G; i.e. $e = (u, v)$ for some $v \in V(G)$. Then in H, we have the edges (u', x_e) and (u', y_e). If $w \notin \{x_e, y_e\}$ then $u'x_e w y_e u'$ is a cycle on four vertices and therefore we have $c_H(u') \neq c_H(w)$, which is a contradiction. On the other hand, $w \in \{x_e, y_e\}$, we can assume without loss of generality that $w = x_e$, and then we have that $u'x_e z y_e u'$ is a cycle on four vertices, where z is any vertex in $K \setminus \{x_e, y_e\}$. Again, this means that $c_H(u')$ is different from $c_H(x_e) = c_H(w)$, which is a contradiction. Thus we have that $c_H(u') \notin \{k+1, k+2, \ldots, k+2m\}$

for any $u' \in I$, or in other words, $c_H(u') \in \{1, 2, \ldots, k\}$ for each $u' \in I$. We shall now define $c : V(G) \to \{1, 2, \ldots, k\}$ as follows: we simply let $c(u) = c_H(u')$ for each $u \in V(G)$. From our observations above, it is clear that $c(u) \in \{1, 2, \ldots, k\}$ for each $u \in V(G)$. We claim that c is a proper coloring of G. Suppose for the sake of contradiction that there exists an edge $e = (u, v) \in E(G)$ such that $c(u) = c(v)$. From the construction of c, we now have that $c_H(u') = c_H(v')$. Also, from the construction of H, it is clear that $x_e u' y_e v' x_e$ is a cycle of length 4 in H. But this contradicts the fact that c_H is a TR-C_4 coloring of H. Thus we can conclude that c is a proper coloring of G. This shows that $\chi(G) \leq k$, and we are done. Since it is easy to see that the graph H can be constructed from G in polynomial-time, this completes the proof. □

Theorem 8. *For each $k \geq 5$, TR-C_k COLORABILITY can be solved in polynomial time for split graphs that have at least $k - 2$ vertices on the clique side.*

Proof. Let $G(K, I)$ be a split graph where K is the clique side of G and I is the independent set side of G. We assume that $|K| \geq k - 2$. Since a vertex of degree one cannot be contained in any cycle, if G contains a degree one vertex u (then clearly, $u \in I$), then we can construct a TR-C_k coloring of $G - u$, and then color u with any color so as to get a TR-C_k coloring of G using the same number of colors. Since any TR-C_k coloring of G, when restricted to $G - u$, gives a TR-C_k coloring of $G - u$, a TR-C_k coloring of G that we construct from an optimal TR-C_k coloring of $G - u$ will also be optimal. So we can reduce our problem to the problem of computing an optimal TR-C_k coloring for split graphs having at least $k - 2$ vertices on the clique side, and containing no degree one vertices.

Next, if there exists $u, v \in I$ such that u and v have degree 2 and $N(u) = N(v)$ then clearly, no cycle of length k can contain both u and v, since $k \geq 5$. Thus, we can construct a TR-C_k coloring c of $G - u$ and then color u with the color $c(v)$ to get a coloring c' of G using the same number of colors as c. If there is a cycle of length k that is not rainbow in c', then clearly that cycle contains u and not v, but then replacing u with v gives a cycle of length k in $G - u$ that is not rainbow in c, which is a contradiction. Thus c' is a TR-C_k coloring of G that uses the same number of colors as c. Again, for the same reason as before, if c is an optimal TR-C_k coloring of $G - u$, then c is an optimal TR-C_k coloring of G. This means that we can reduce our problem to the problem of computing a TR-C_k coloring for split graphs having at least $k - 2$ vertices on the clique side, and containing no degree one vertices, and containing no pair of degree 2 vertices having the same neighbourhood.

From the above discussion, we can assume that G contains no degree one vertices and no pair of degree two vertices having the same neighbourhood.

Now if $|V(G)| < k$, then there cannot be a cycle of length k in G. So we assume that G has at least k vertices. We are given that $|K| \geq k - 2$.

For any vertex $u \in I$ and $v \in K$, it can be seen that u, any two vertices in $N(u)$, and any $k - 3$ vertices from $K - N(u)$ form a C_k in G (recall that each vertex in I has degree at least 2). Thus there is always a cycle of length k that contains both u and v. Thus the vertices in I all receive colors different from the colors of the vertices in K in any TR-C_k coloring of G.

We now claim that any two vertices in I are contained in a C_k in G. Since all vertices have degree at least 2 and no two degree 2 vertices have the same neighbourhood, we have that for any pair of vertices $u, v \in I$ having degree 2, one of the following happens: either $|N(u) \cap N(v)| = 1$ (shown in Figure 4.6) or $|N(u) \cap N(v)| = 0$ (shown in Figure 4.7). It is not difficult to see that there are cycles of length k containing u and v in both the cases. Even if u and/or v have degree more than 2, at least one of the configurations shown in Figure 4.6 or 4.7 can always be reached by deleting some edges in the graph. So there is a cycle of length k containing u and v in this case too. Thus any two vertices in I have to receive different colors in any TR-C_k coloring of G.

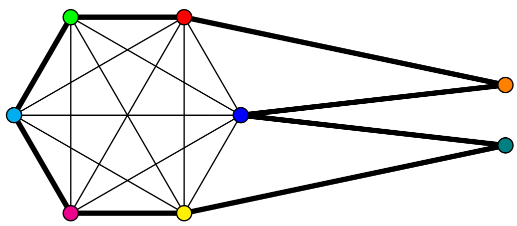


Figure 4.6: $|K| = k - 2$, $N(u) \cap N(v) \neq \emptyset$ (here $k = 8$)

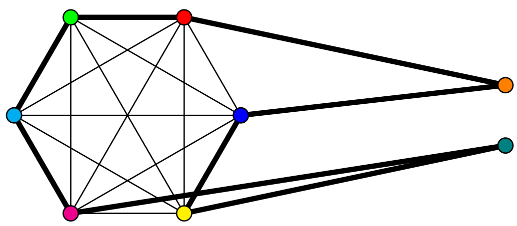


Figure 4.7: $|K| = k - 2$, when $N(u) \cap N(v) = \emptyset$ (here $k = 8$)

We now show that no two vertices in K can receive the same color in any TR-C_k coloring of G. Let us first consider the case when $|K| = k - 2$. Since we have assumed that G contains a C_k, we know that $|I| \geq 2$. Now consider any distinct $u, v \in I$. From the observation in the previous paragraph, we have that there is a C_k containing u, v and all the vertices in K. This means that if $|K| = k-2$, then each vertex of K receives a different color in any TR-C_k coloring of G. Next consider the case when $|K| = k - 1$. Then since G contains a C_k, we know that $|I| \geq 1$. Then as observed previously, any vertex in I together with the vertices in K forms a C_k in G. Thus in this case too, each vertex of K has to receive a different color in any TR-C_k coloring of G. Finally, if $|K| \geq k$, then any two vertices $u, v \in K$ together with $k - 2$ other vertices of K lie on a cycle of length k in G. Thus no two vertices in K receive the same color in any TR-C_k coloring of G in this case too.

We have thus proved that every vertex of G gets a different color in any TR-C_k coloring of G if G contains a cycle of length k. Clearly, giving every vertex a different color always results in a TR-C_k coloring of G, and by our arguments above, such a coloring is optimal when there is a cycle of length k. Since it can be checked in polynomial time whether the graph G contains a C_k, and in case it does not, giving every vertex the same color is an optimal TR-C_k coloring

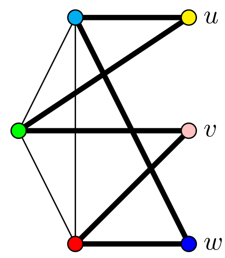


Figure 4.8: TR coloring with C_6 as template

of G, we can conclude that TR-C_k COLORABILITY is polynomial time solvable for split graphs that contain at least $k-2$ vertices on the clique side. □

Corollary 9. *TR-C_5* COLORABILITY *is polynomial time solvable for split graphs.*

Proof. Let G be a split graph with K denoting its clique side and I denoting its independent set side. Any cycle on 5 vertices in G can have at most 2 vertices from I. Thus every C_5 in G contains at least 3 vertices from K. Thus, if $|K| < 3$, then there cannot exist any C_5 in G, hence TR-C_5 coloring of G can be done optimally with just a single color. So we assume that $|K| \geq 3$. Then we have $|K| \geq k-2$ (here we let $k=5$), and therefore we can use the previous theorem to find an optimal TR-C_5 coloring of G in polynomial time. □

Corollary 10. *TR-C_6* COLORABILITY *is polynomial time solvable for split graphs.*

Proof. Let G be a split graph whose clique side is K and independent set side is I. As in the proof of the previous theorem, we can assume that G contains no vertices of degree 1, and that no two vertices I having degree 2 have the same neighbourhood. If $|K| \geq 4$, then we can apply the previous theorem setting $k=6$, to conclude that an optimal TR-C_6 coloring of G can be computed in polynomial time. Again if $|K| < 3$, then there cannot exist any C_6 in G, and therefore we can generate an optimal TR-C_6 coloring of G by giving every vertex a single color. So let us assume that $|K| = 3$. If $|I| < 3$, then $|V(G)| < 6$, and then again there is no C_6 in G, and we can produce an optimal TR-C_6 coloring of G as before. So we can assume that $|K| = 3$ and $|I| \geq 3$. Let u, v, w be any three vertices in I. Then since no two degree 2 vertices in I have the same neighbourhood, we can reach the configuration shown in Figure 4.8, by removal of some edges, if necessary. Then clearly, there is a C_6 that contains all the vertices in K and each of u, v, w. This means that any pair of vertices in G lies on a C_6, which implies that every vertex of G has to get a different color in any TR-C_6 coloring of G. Thus we can generate an optimal TR-C_6 coloring of G by just giving every vertex a different color. This completes the proof. □

5 Future work and conclusion

It would be interesting to try to construct a polynomial time algorithm for template driven rainbow coloring for split graphs with a cycle C_k as a template, where k is a fixed integer greater than 6. We feel that by exploring the structure of pairs of vertices in I that do not together lie on any C_k, some progress can be achieved on this problem. One could also try solving this problem for split graphs for templates other than cycles, but the problem seems to be quite difficult. Considering that the problem is NP-hard for the template C_4 even in split graphs, it seems plausible that the problem could be NP-hard to solve for interval graphs with C_k as a template.

Bibliography

[1] Noga Alon, Colin Mcdiarmid, and Bruce Reed. Acyclic coloring of graphs. *Random Structures & Algorithms*, 2(3):277–288, 1991.

[2] Kenneth Appel and Wolfgang Haken. Every planar map is four colorable. Part I: Discharging. *Illinois Journal of Mathematics*, 21:429–490, 1977.

[3] Kenneth Appel, Wolfgang Haken, and John Koch. Every planar map is four colorable. Part II: Reducibility. *Illinois Journal of Mathematics*, 21:491–567, 1977.

[4] Lowell W. Beineke and Robin J. Wilson, editors. *Topics in Chromatic Graph Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2015.

[5] L.S. Chandran, S.K. Das, P. Hell, S. Padinhatteeri, and R.R. Pillai. Template-driven rainbow coloring of proper interval graphs. *Discrete Applied Mathematics*, 328:97–107, 2023.

[6] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164:51–229, 2006.

[7] Sajal K. Das, Irene Finocchi, and Rossella Petreschi. Conflict-free star-access in parallel memory systems. *Journal of Parallel and Distributed Computing*, 66(11):1431–1441, 2006.

[8] C.M.H. de Figueiredo. Total colouring. In Lowell W. Beineke, Martin C. Golumbic, and Robin J. Wilson, editors, *Topics in Algorithmic Graph Theory*, Encyclopedia of Mathematics and its Applications, chapter 3. Cambridge University Press, 2021.

[9] Stephane Foldes and Peter L. Hammer. Split graphs. Proc. 8th southeast. Conf. on Combinatorics, graph theory, and computing; Baton Rouge 1977, 311–315, 1977.

[10] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

[11] Oded Goldreich. *P, NP, and NP-Completeness: The Basics of Computational Complexity*. Cambridge University Press, 2010.

[12] Martin Charles Golumbic. Interval graphs and related topics. *Discrete Mathematics*, 55(2):113–121, 1985.

[13] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Elsevier Science, 2004.

[14] T.R. Jensen and B. Toft. *Graph Coloring Problems*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011.

[15] Daniel Leven and Zvi Galil. NP completeness of finding the chromatic index of regular graphs. *Journal of Algorithms*, 4(1):35–44, 1983.

[16] R.M. R. Lewis. *A Guide to Graph Colouring: Algorithms and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2015.

[17] László Lovász. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B*, 13(2):95–98, 1972.

[18] László Lovász. Normal hypergraphs and the perfect graph conjecture. *Discrete Mathematics*, 2(3):253–267, 1972.

[19] Jan Mycielski. Sur le coloriage des graphs. *Colloquium Mathematicae*, 3(2):161–162, 1955.

[20] Michael Stiebitz and Bjarne Toft. Brooks's theorem. In Lowell W. Beineke and Robin J. Wilson, editors, *Topics in Chromatic Graph Theory*, Encyclopedia of Mathematics and its Applications, pages 36–55. Cambridge University Press, 2015.

[21] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[22] S. Vishwanathan. Randomized online graph coloring. In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, pages 464–469 vol.2, 1990.

[23] V. G. Vizing. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz*, 3:25–30, 1964.

[24] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 2 edition, 2000.

[25] Hong Yang, Muhammad Naeem, and Shahid Qaisar. On the P_3 coloring of graphs. *Symmetry*, 15(2), 2023.