Building a larger class of graphs for efficient reconfiguration of vertex colouring

by

Owen Merkel

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2020

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Owen Merkel is the sole author of Chapters 1, 2, 3, and 6 which were written under the supervision of Prof. Therese Biedl and Prof. Anna Lubiw and were not written for publication.

Exceptions to sole authorship are Chapters 4 and 5, which is joint work with Prof. Therese Biedl and Prof. Anna Lubiw and consists in part of a manuscript written for publication.

Abstract

A k-colouring of a graph G is an assignment of at most k colours to the vertices of G so that adjacent vertices are assigned different colours. The reconfiguration graph of the k-colourings, $\mathcal{R}_k(G)$, is the graph whose vertices are the k-colourings of G and two colourings are joined by an edge in $\mathcal{R}_k(G)$ if they differ in colour on exactly one vertex. For a k-colourable graph G, we investigate the connectivity and diameter of $\mathcal{R}_{k+1}(G)$. It is known that not all weakly chordal graphs have the property that $\mathcal{R}_{k+1}(G)$ is connected. On the other hand, $\mathcal{R}_{k+1}(G)$ is connected and of diameter $O(n^2)$ for several subclasses of weakly chordal graphs such as chordal, chordal bipartite, and P_4-free graphs.

We introduce a new class of graphs called OAT graphs that extends the latter classes and in fact extends outside the class of weakly chordal graphs. OAT graphs are built from four simple operations, disjoint union, join, and the addition of a clique or comparable vertex. We prove that if G is a k-colourable OAT graph, then $\mathcal{R}_{k+1}(G)$ is connected with diameter $O(n^2)$. Furthermore, we give polynomial time algorithms to recognize OAT graphs and to find a path between any two colourings in $\mathcal{R}_{k+1}(G)$.

Feghali and Fiala defined a subclass of weakly chordal graphs, called compact graphs, and proved that for every k-colourable compact graph G, $\mathcal{R}_{k+1}(G)$ is connected with diameter $O(n^2)$. We prove that the class of OAT graphs properly contains the class of compact graphs. Feghali and Fiala also asked if for a k-colourable (P_5, co-P_5, C_5)-free graph G, $\mathcal{R}_{k+1}(G)$ is connected with diameter $O(n^2)$. We answer this question in the positive for the subclass of P_4-sparse graphs, which are the (P_5, co-P_5, C_5, P, co-P, fork, co-fork)-free graphs.

Acknowledgements

I would like to thank my supervisors Prof. Therese Biedl and Prof. Anna Lubiw for their helpful advice, support, and guidance. I would also like to thank Prof. Naomi Nishimura and Prof. Lap Chi Lau for agreeing to be readers for the thesis and providing useful comments and feedback.

I also thank Prof. Kathie Cameron for her guidance and mentorship and for encouraging me to pursue graduate studies.

I am very grateful for the support of my family. Without the help of my parents, this would not have been possible.

Table of Contents

List of Figures

Chapter 1

Introduction

A reconfiguration framework consists of states and transitions between states. The states represent feasible solutions to a source problem and there is a transition between states if the corresponding feasible solutions satisfy a predefined adjacency relationship. Reconfiguration has been considered with many different source problems (see e.g. [51]) and these reconfiguration problems have a wide range of applications. In this thesis, we focus on the reconfiguration of vertex colouring.

The reconfiguration graph of the k-colourings, $\mathcal{R}_k(G)$, is the graph whose vertices are the k-colourings of G and two colourings are joined by an edge in $\mathcal{R}_k(G)$ if they differ in colour on exactly one vertex. See Figure 1 for an example of a 2-colourable graph G and a small portion of $\mathcal{R}_3(G)$.

Reconfiguration of vertex colouring can be used to model real-world problems, for example in the reassignment of frequencies in a wireless network. Cereceda also noticed that reconfiguration of vertex colouring was used in the study of statistical physics, in particular, in the study of the Glauber dynamics of an anti-ferromagnetic Potts model (see [14]). These

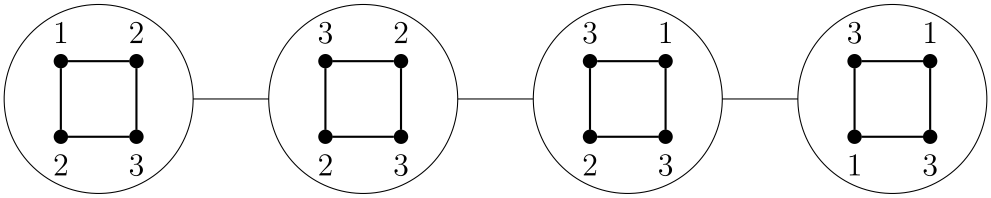


Figure 1.1: An example of a graph G and a subgraph of $\mathcal{R}_3(G)$.

applications have motivated many researchers to study colouring reconfiguration, leading to new and interesting proof techniques to solve these problems.

Two such problems are concerned with the connectivity and diameter of $\mathcal{R}_k(G)$. The first problem is to determine if for any two colourings, it is always possible to reconfigure one colouring into the other by recolouring a single vertex at a time while ensuring that all intermediate colourings remain proper. The second problem is to determine an upper bound on the number of recolourings needed to reconfigure one colouring to another.

In this thesis, we only consider reconfiguration of vertex colouring. We note that several variants of graph colouring have been considered for reconfiguration including list colouring, edge colouring, acyclic colouring, and equitable colouring. The vertex colouring variant has also been considered with a different reconfiguration step. Instead of recolouring a single vertex at each step, the colours of a Kempe chain are interchanged (see Section 3.1.2).

We introduce a class of graphs which we call *OAT* graphs that can be constructed from four simple operations. We will prove that OAT graphs have nice properties for colouring reconfiguration.

Definition 1. *A graph G is an OAT graph if it can be constructed from single vertex graphs with a finite sequence of the following four operations. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be vertex-disjoint OAT graphs.*

1. *Taking the* disjoint union *of G_1 and G_2, defined as $(V_1 \cup V_2, E_1 \cup E_2)$.*
2. *Taking the* join *of G_1 and G_2, defined as $(V_1 \cup V_2, E_1 \cup E_2 \cup \{xy \mid x \in V_1, y \in V_2\})$.*
3. *Adding a vertex $u \notin V_1$* comparable *to vertex $v \in V_1$, defined as $(V_1 \cup \{u\}, E_1 \cup \{ux \mid x \in X\})$, where $X \subseteq N(v)$.*
4. *Attaching a complete graph $Q = (V_Q, E_Q)$ to a vertex v of G_1, defined as $(V_1 \cup V_Q, E_1 \cup E_Q \cup \{qv \mid q \in V_Q\})$.*

These operations are summarized in Figure 1.2. The class of OAT graphs also includes graphs that are not perfect as illustrated in Figure 1.3. However, for any graph G in this class, the chromatic number of G is equal to the clique number of G.

Observation 1. *If G is an OAT graph, then $\chi(G) = \omega(G)$.*

Specifically, the chromatic number and clique number of an OAT graph changes with each operation as follows.

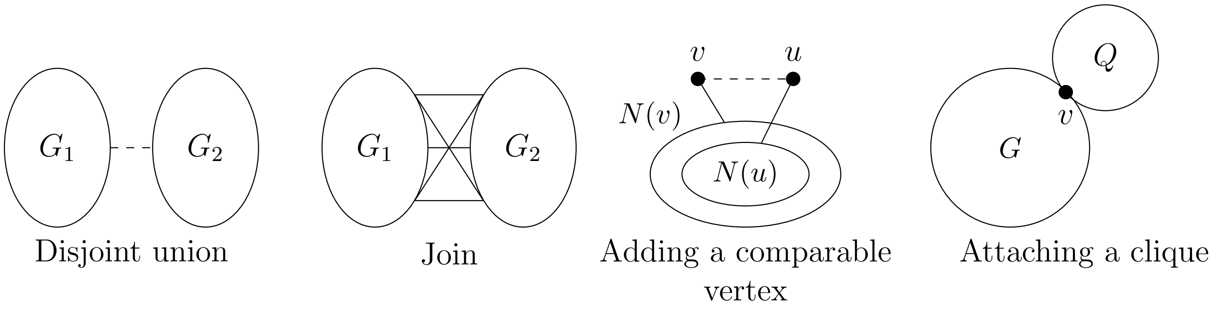


Figure 1.2: The four operations that construct OAT graphs. A dashed line indicates no edge.

1. If G is the disjoint union of the graphs G_1 and G_2, then $\chi(G) = \max\{\chi(G_1), \chi(G_2)\}$ and $\omega(G) = \max\{\omega(G_1), \omega(G_2)\}$.
2. If G is the join of the graphs G_1 and G_2, then $\chi(G) = \chi(G_1) + \chi(G_2)$ and $\omega(G) = \omega(G_1) + \omega(G_2)$.
3. If G is obtained by adding a comparable vertex to a graph H then $\chi(G) = \chi(H)$ and $\omega(G) = \omega(H)$.
4. If G is obtained by attaching a complete graph Q to a graph H, then $\chi(G) = \max\{\chi(H), |Q| + 1\}$ and $\omega(G) = \max\{\omega(H), |Q| + 1\}$.

Up to the knowledge of the author, this is the first time that results on the diameter of $\mathcal{R}_{k+1}(G)$ have been proven for a class that includes graphs which are not perfect. Next, we give several observations to help the reader gain intuition on how to construct simple graph classes using the operations given in Definition 1.

Observation 2. *The complete graph K_n is an OAT graph for every $n \geq 1$.*

Proof. The graph K_n can be constructed by repeatedly applying the join operation with single vertices. □

Observation 3. *The complete bipartite graph $K_{p,q}$ is an OAT graph for every $p, q \geq 1$.*

Proof. To construct $K_{p,q}$, first construct each bipartite set using the disjoint union operation. Then, apply the join operation between the bipartite sets. □

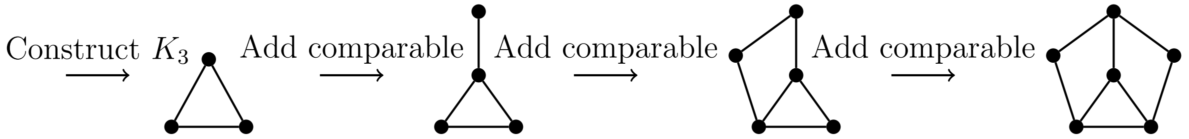


Figure 1.3: Constructing an OAT graph that is not perfect.

Observation 4. *The path P_n is an OAT graph for every $n \geq 1$.*

Proof. The graph P_n can be constructed from a single edge by repeatedly adding a pendant vertex, which is also a comparable vertex. □

Observation 5. *The cycle C_n is not an OAT graph for $n \geq 5$ but there exist OAT graphs that contain C_n as an induced subgraph for every $n \geq 3$.*

Proof. First observe that no operation could have been used to construct the graph C_n for $n \geq 5$. Since C_n is connected, it could not have been constructed from the disjoint union operation. Since co-C_n is connected for $n \geq 5$, C_n could not have been constructed from the join operation. Since the neighbourhood of each vertex of C_n is exactly two and no two vertices share exactly the same neighbourhood, C_n could not have been constructed from adding a comparable vertex. Finally, since C_n is biconnected, it could not have been constructed from attaching a clique.

To construct an OAT graph that contains C_{2k+1} for $k \geq 2$, add $k-1$ pendant vertices to the same vertex v of a clique with three vertices. Next, add vertices of degree two that are comparable to v to construct the cycle (see also Figure 1.3). The graph C_{2k+1} can then be obtained by deleting the vertex v. To construct an OAT graph that contains C_{2k} for $k \geq 2$, begin by constructing the star graph $K_{1,k}$ (see Observation 3) with center v. Next, add $k-1$ vertices of degree two comparable to v to construct the cycle. The graph C_{2k} can then be obtained by deleting the vertex v. □

Next, we give motivation for studying OAT graphs and in particular, the reconfiguration graph of OAT graphs. Bonamy et al. [8] asked the following question. Given a k-colourable perfect graph G, is $\mathcal{R}_{k+1}(G)$ connected with diameter $O(n^2)$? One cannot hope for a smaller diameter since for P_n, the path on n vertices, $\mathcal{R}_3(P_n)$ has diameter $\Omega(n^2)$ [8]. Bonamy and Bousquet [4] answered this question in the negative, using an example of Cereceda, van den Heuvel, and Johnson [15], who showed that there exists a bipartite graph G where $\mathcal{R}_{k+1}(G)$ has an isolated vertex and where k can be arbitrarily large (we will see it in Figure 3.6). Feghali and Fiala [33] also investigated this question and found an infinite family of weakly chordal graphs G where $\mathcal{R}_{k+1}(G)$ has an isolated vertex (we will see it in Figure 3.14). In the same paper, Feghali and Fiala introduced a subclass of weakly chordal graphs called *compact* graphs (definitions and details in Section 3.5.7). They proved that for a k-colourable compact graph G, $\mathcal{R}_{k+1}(G)$ is connected with diameter $O(n^2)$.

The following theorem is the main result of this thesis.

Theorem 1. *Let G be an OAT graph and let $k \geq \chi(G)$. Then $\mathcal{R}_{k+1}(G)$ is connected with diameter at most $4n^2$.*

In addition, we prove that the class of OAT graphs contains the class of P_4-free graphs, chordal bipartite graphs, and compact graphs, unifying several results in the literature. We also prove that the class of OAT graphs contains the class of P_4-sparse graphs, for which colouring reconfiguration results were not previously known.

Another result of this thesis is an algorithm that recognizes OAT graphs in $O(n^3)$ time.

Theorem 2. *There exists an $O(n^3)$ time algorithm to test whether a given graph G on n vertices is an OAT graph.*

1.1 Organization

The rest of this thesis is organized as follows.

In Chapter 2, we provide the graph-theoretic definitions and notation used in this thesis. We review concepts used in graph colouring and we define the main graph classes that are discussed throughout the thesis.

In Chapter 3 we briefly review reconfiguration frameworks and define several problems relating to reconfiguration of vertex colouring. We survey results regarding the reconfiguration graph of vertex colourings with the number of colours depending on several different graph invariants. These invariants include the maximum degree, maximum average degree, degeneracy, treewidth, and Grundy number. Following this, we survey results on the complexity of k-Colour Path and k-Colour Bounded Path. Finally, we survey results regarding the k-recolouring diameter of a graph G where $k \geq \chi(G) + 1$ and where G belongs to one of several graph classes. We also discuss the relationship between OAT graphs and these graphs classes.

In Chapter 4, we prove our main result: If G is a k-colourable OAT graph, then $\mathcal{R}_{k+1}(G)$ is connected with diameter $O(n^2)$.

In Chapter 5, we prove several lemmas that together imply that OAT graphs can be recognized in polynomial time. We give an $O(n^3)$ time algorithm to recognize OAT graphs and provide pseudocode for the algorithm.

Finally, in Chapter 6, we summarize the results of this thesis and end with several open problems.

Chapter 2

Preliminaries

This chapter is dedicated to providing the notation and definitions used throughout the thesis. In Section 2.1, we give standard graph-theoretic notation and terminology. In Section 2.2, we define graph colouring and discuss the greedy colouring algorithm. In Section 2.3, we introduce several classes of graphs that will be referred to in this thesis and discuss the relationship between these classes.

2.1 Graphs

For definitions not given here, refer to the book by Diestel [24]. Throughout this thesis, all graphs considered are finite, simple (loops and multiple edges are not permitted), and undirected, unless otherwise stated. Let $G = (V, E)$ be a graph. The vertex-set of G is denoted by $V(G)$ and the edge-set of G is denoted by $E(G)$. We typically use $n = |V(G)|$ to denote the number of vertices of G, and we typically use $m = |E(G)|$ to denote the number of edges of G. Two vertices $u, v \in V(G)$ are called *adjacent* in G if there is an edge between u and v. The *complement* of a graph G, denoted co-G, is the graph with vertex-set $V(G)$ where two vertices $u, v \in V(G)$ are adjacent in co-G if and only if they are not adjacent in G. The *neighbourhood* of a vertex $v \in V(G)$, denoted by $N_G(v)$ (or simply $N(v)$ if the context is clear), is the set of vertices adjacent to v in G. The *closed neighbourhood* of v, denoted $N[v]$, is $N(v) \cup \{v\}$. For $u \in N(v)$, we say u is a *neighbour* of v.

For an edge $e = uv \in E(G)$, u and v are called the *endpoints* of e. A *matching* in a graph G is a set $M \subseteq E(G)$ such that no two edges in M share a common endpoint. A vertex is *matched* if it is the endpoint of some edge in a matching. A matching M of G is called *perfect* if every vertex of G is matched in M.

The *degree* of a vertex v, denoted by $d(v)$, is equal to $|N(v)|$. A graph is said to be d-regular if every vertex has degree d. The *maximum degree* of G, denoted by $\Delta(G)$, is the maximum degree of a vertex of G. A graph is d-degenerate if for all subgraphs H of G, H has a vertex with degree at most d.

For $U \subseteq V(G)$, the subgraph of G *induced* by U, denoted by $G[U]$, is the graph with vertex-set U where vertices $u, v \in U$ are adjacent in $G[U]$ if and only if u and v are adjacent in G. For a graph H, we say that G is *H-free* if G does not contain an induced subgraph isomorphic to H. For a set of graphs $\mathcal{H}$, we say that G is $\mathcal{H}$-free if G is H-free for every $H \in \mathcal{H}$. For convenience and when the context is clear, we consider a set of vertices $U \subseteq V(G)$ as a subgraph, namely, the subgraph of G induced by U. We use the notation $G \setminus v$ to denote the graph obtained from G by deleting v and all edges incident to v. We use the notation $G \setminus U$ to denote the graph obtained from G by deleting all vertices of U and all edges incident to the vertices of U. A vertex $v \in V(G)$ is a *cut vertex* if $G \setminus v$ has more connected components than G. A *separator* of a graph G is a set $S \subset V(G)$, such that $G \setminus S$ has more connected components than G.

We use P_n to denote the path on n vertices and C_n to denote the cycle on n vertices. A *hole* is a cycle on at least five vertices and an *anti-hole* is the complement of a hole. A hole is *even* or *odd* if it has an even or odd number of vertices, respectively. The *girth* of a graph is the minimum number of vertices in an induced cycle. A *tree* is a connected graph with no induced cycles.

The *length* of a path between two vertices is the number of edges in the path. The *distance* between two vertices $u, v \in V(G)$, denoted by $d(u, v)$, is the minimum length of a path between u and v. If there is no path between u and v, then $d(u, v) = \infty$. The *diameter* of a graph is the maximum distance between two vertices in the graph. For non-adjacent vertices $u, v \in V(G)$, we say that u is *comparable* to v if $N(u) \subseteq N(v)$. For vertices $x, y \in V(G)$, x and y are called *true twins* if $N(x) = N(y)$ and x and y are adjacent. The vertices x and y are called *false twins* if $N(x) = N(y)$ and x and y are not adjacent.

A *stable set* in a graph is a set of vertices with no edges between them. A graph is *bipartite* if its vertex-set can be partitioned into two stable sets. The complete bipartite graph $K_{p,q}$ is the bipartite graph with p and q vertices in each bipartition, such that there are all possible edges between them. For $q \geq 3$, the graph $K_{1,q}$ is called a *star graph* and the vertex in the bipartite set with exactly one vertex is called the *center* of the graph.

A *clique* in a graph is a set of vertices with all possible edges between them. The *clique number* of G, denoted by $\omega(G)$, is the size of a largest clique in G. A graph is called *complete* if it has all possible edges. The complete graph on n vertices is denoted by K_n. We use the term clique and complete graph interchangeably even though one refers to a set of vertices and the other refers to a graph. For $X, Y \subseteq V(G)$, we say that X is *joined* to Y if every vertex in X is adjacent to every vertex in Y.

2.2 Graph Colouring

For a positive integer k, a *k-colouring* of a graph G is a function $c : V(G) \to \mathcal{S}$ for some finite set $\mathcal{S}$ with $|\mathcal{S}| \leq k$. A colouring c is called *proper* if $c(u) \neq c(v)$ whenever u and v are adjacent in G. Throughout this thesis, we only consider proper colourings and omit the use of the word "proper". Unless otherwise stated, we take $\mathcal{S} = \{1, 2, \ldots, k\}$. Note that a k-colouring of a graph G induces a partition of the vertices of G into k (possibly empty) stable sets. We refer to these stable sets as *colour classes*.

A graph is *k-colourable* if it admits a k-colouring. The *chromatic number* of a graph G, denoted by $\chi(G)$, is the minimum number k for which G is k-colourable. A graph G is *perfect* if for all induced subgraphs H of G, $\chi(H) = \omega(H)$.

A naive method used to colour a graph is the *greedy colouring algorithm*. The greedy colouring algorithm is a procedure that takes as input a graph G and an ordering of the vertices of G and computes a colouring of G as follows. For the ordering $\{v_1, v_2, \ldots, v_n\}$ of G and for each $1 \leq i \leq n$, colour the vertex v_i with the smallest colour that has not been used to colour a neighbour of v_i. Clearly, the greedy colouring algorithm gives a proper colouring of the vertices. However, the number of colours used in a greedy colouring can be much larger than the chromatic number, depending on which ordering of the vertices is used.

It is easy to see that there exists an ordering of the vertices of G such that the greedy colouring algorithm uses the minimum number of colours. Take any $\chi(G)$-colouring α of G and order the vertices as follows. Start by taking all the vertices of G coloured 1, followed by all the vertices coloured 2, and so on. Another ordering (if it exists) where the greedy colouring algorithm will use the minimum number of colours is a *perfect elimination ordering*. A *perfect elimination ordering* of G is an ordering $\{v_1, v_2, \ldots v_n\}$ of the vertices of G such that for $1 \leq i \leq n$, the neighbourhood of v_i in the subgraph of G induced by $\{v_1, v_2, \ldots, v_{i-1}\}$ is a clique. It is well known that the graphs that have perfect elimination orderings are exactly the chordal graphs (we will define chordal graphs in the next section). The greedy colouring algorithm will colour a graph with the minimum number of colours if a perfect elimination ordering is given since whenever a vertex is coloured, its neighbourhood forms a clique.

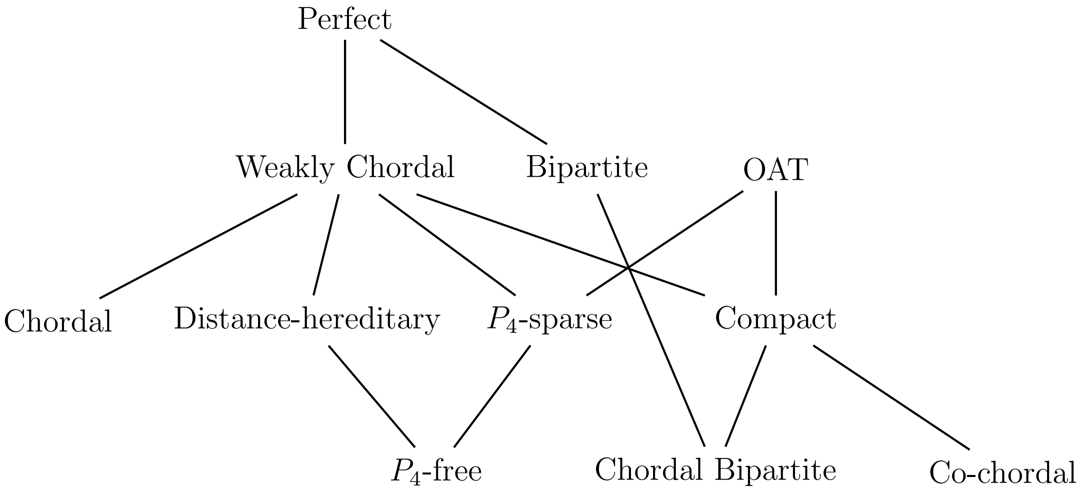


Figure 2.1: The relationship between graph classes.

2.3 Graph Classes

In this section we define several classes of graphs that have been considered for reconfiguration of vertex colouring. We will review results for these classes in Section 3.5. Figure 2.1 illustrates the relationship between these graph classes. A class $\mathcal{C}$ is a subclass of another class $\mathcal{D}$ if and only if $\mathcal{C}$ is a descendent of $\mathcal{D}$.

Recall that a graph is *bipartite* if its vertex-set can be partitioned into two stable sets. It is well known that bipartite graphs are exactly the graphs with no odd cycles. A graph is *chordal* if it does not contain a cycle with four or more vertices as an induced subgraph. It is well known that a graph is *chordal* if and only if it has a perfect elimination ordering [37]. A graph is *co-chordal* if it is the complement of a chordal graph.

Recall that a graph is H-free if it does not contain an induced subgraph isomorphic to H. For example, P_4-free graphs do not contain the graph P_4 as an induced subgraph. The class of OAT graphs generalizes the class of P_4-free graphs, also called *cographs*, because they are exactly the graphs that can be constructed from single vertex graphs with the join and disjoint union operation [22]. The following two classes generalize P_4-free graphs.

Definition 2 ([41]). *A graph is P_4-*reducible *if each vertex of the graph is in at most one induced P_4.*

Definition 3 ([39]). *A graph is P_4-*sparse *if for every set of 5 vertices, there is at most one induced P_4.*

Another class of graphs that generalizes the class of P_4-free graphs is the class of distance-hereditary graphs [38]. A graph G is *distance-hereditary* if for all connected induced subgraphs H of G, and any two vertices x, y of H, the distance between x and y in H is the same as the distance between x and y in G. The class of distance-hereditary graphs is exactly the class of graphs that can be constructed from single vertex graphs by adding a pendant vertex, a true twin, or a false twin [38].

A graph is *weakly chordal* if it does not contain a hole or an anti-hole as an induced subgraph. A graph is *chordal bipartite* if it is both weakly chordal and bipartite. Recall that a graph G is *perfect* if for all induced subgraphs H of G, $\chi(H) = \omega(H)$. The strong perfect graph theorem [20] states that a graph is perfect if and only if it does not contain an induced subgraph isomorphic to an odd hole or an odd anti-hole. It is clear from this characterization that every weakly chordal graph is perfect. In fact, many classes of graphs discussed in this thesis are subclasses of perfect graphs. For example, bipartite, P_4-free, P_4-sparse, distance-hereditary, chordal, co-chordal, compact, and weakly chordal graphs are all classes of perfect graphs.

A graph is *planar* if it can be drawn in the plane with points for vertices and curves for edges such that edges only meet at vertices. A *plane* graph is a planar graph drawn in such a way. The faces of a plane graph are the regions of the plane bounded by the edges of the graph, including the unbounded outer region. Euler's formula states that for a connected plane graph with n vertices, m edges, and f faces, $n - m + f = 2$. The famous four-colour theorem states that every planar graph is 4-colourable [2].

Chapter 3

Literature Review

In this chapter, we survey results on reconfiguration of vertex colouring with various restrictions on the number of colours and for various graph classes. In Section 3.1 we introduce the general notion of reconfiguration and discuss reconfiguration of graph colouring. In Section 3.2, we survey results concerning the connectivity and diameter of $\mathcal{R}_k(G)$ where the parameter k is defined in terms of several graph invariants, including the maximum degree, maximum average degree, and the Grundy number. In Section 3.3, we survey results on Cereceda's conjecture and the cases for which it has been proven. In Section 3.4, we survey results on the complexity of several reconfiguration problems. In Section 3.5 we survey results on the connectivity and diameter of $\mathcal{R}_k(G)$ where the parameter k is in terms of the chromatic number.

3.1 Reconfiguration

In this section, we give definitions and notation for reconfiguration problems that are discussed throughout the thesis. Reconfiguration frameworks consist of states and transitions between states usually modelled by a *reconfiguration graph*. We saw these briefly in Figure 1.1 but give the detailed definition here. The nodes of the reconfiguration graph represent feasible solutions to a source problem and there is an edge between two nodes in the reconfiguration graph if the corresponding feasible solutions satisfy a predefined adjacency relationship. This adjacency relationship is called a *reconfiguration step* and is a transformation rule that transforms one feasible solution into another. A *reconfiguration sequence* is a path between two nodes in the reconfiguration graph.

Reconfiguration problems have various applications, for example modelling dynamic situations for which a feasible solution is known but for which a more desirable solution is wanted. These applications lead to studying the following problems on the reconfiguration graph. The simplest problem is about the existence of a path between two nodes in the reconfiguration graph. Given two nodes x and y in the reconfiguration graph, is there a reconfiguration sequence between x and y? The second problem is about the connectivity of the reconfiguration graph. For all pairs of nodes x and y, is there a reconfiguration sequence between x and y? The third problem is about the diameter of the reconfiguration graph. That is, what is the maximum length of a shortest reconfiguration sequence between any two nodes in the reconfiguration graph? And finally, the most difficult problem is about finding a shortest path between two nodes. Given two nodes x and y in the reconfiguration graph, what is a shortest reconfiguration sequence between them? Refer to [51] for a survey on reconfiguration.

3.1.1 Colouring Reconfiguration

In this thesis, we focus on the reconfiguration graph of vertex colourings, defined as follows.

Definition 4. *Given a k-colourable graph G, the* reconfiguration graph *$\mathcal{R}_k(G)$ is the graph whose nodes are the proper k-colourings of G and two nodes are joined by an edge in $\mathcal{R}_k(G)$ if the corresponding colourings differ in colour on exactly one vertex.*

A graph G is *k-mixing* if $\mathcal{R}_k(G)$ is connected. The *mixing number* of a graph G is the minimum k for which G is k-mixing. The *k-recolouring diameter* of G is the diameter of $\mathcal{R}_k(G)$. A k-colouring of G is called *frozen* if it is an isolated vertex in the reconfiguration graph $\mathcal{R}_k(G)$. That is, a k-colouring for which all k colours appear on the closed neighbourhood of every vertex. A simple way to show that a graph G is not k-mixing is to exhibit a frozen k-colouring of G.

There are many problems that have been studied in the area of reconfiguration of vertex colouring. We define several of these problems here.

k-Mixing
Instance: A k-colourable graph G.
Question: Is G k-mixing (i.e. is $\mathcal{R}_k(G)$ connected)?

k-Colour Path
Instance: A graph G and two k-colourings of G, α and β.
Question: Is there a path between α and β in $\mathcal{R}_k(G)$?

k-Colour Bounded Path
Instance: A graph G, two k-colourings of G, α and β, and a positive integer l.
Question: Is there a path of length at most l between α and β in $\mathcal{R}_k(G)$?

We survey results regarding the complexity of these problems in Section 3.4.

3.1.2 Reconfiguration with Kempe changes

Reconfiguration of vertex colouring has also been considered with a different reconfiguration step, called a *Kempe change*. Let α be a colouring of G and let c and d be two distinct colours of α. Let $G(c, d)$ denote the subgraph of G induced by the vertices coloured c and d in α. A *Kempe chain* is a connected component of $G(c, d)$. The *Kempe change* reconfiguration step interchanges the colours of some Kempe chain of G. Two colourings are *Kempe equivalent* if each can be obtained from the other by a sequence of Kempe changes. Kempe changes were introduced by Kempe in his failed attempt at proving the four colour theorem. Next we survey some results on Kempe equivalence but note that the rest of this thesis will not discuss reconfiguration using Kempe changes.

Meyniel [49] proved that all 5-colourings of a planar graph are Kempe equivalent. This result was generalized by Las Vergnas and Meyniel [47] who proved that all 5-colourings of a K_5-minor free graph are Kempe equivalent. Mohar [50] proved that all $(k+1)$-colourings of a k-colourable planar graph are Kempe equivalent. In the same paper, Mohar conjectured the following.

Conjecture 1 ([50])**.** *For $k \geq 3$, all k-colourings of a k-regular connected graph that is not a complete graph are Kempe equivalent.*

Conjecture 1 was disproven for the case $k = 3$ by Feghali, Johnson and Paulusma [34], who showed that all 3-colourings of a connected 3-regular graph G are Kempe equivalent unless G is isomorphic to K_4 or the triangular prism. For $k \geq 4$, Conjecture 1 was proven by Bonamy, Bousquet, Feghali, and Johnson [6].

Theorem 3 ([6])**.** *For $k \geq 4$, all k-colourings of a k-regular connected graph that is not a complete graph are Kempe equivalent.*

3.2 Recolouring in terms of several graph invariants

In this section, we survey results on the connectivity and diameter of $\mathcal{R}_k(G)$ where the parameter k is in terms of a graph invariant of G, namely the maximum degree, maximum

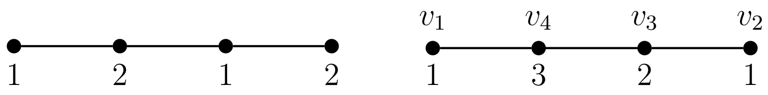


Figure 3.1: A 2-colouring of P_4 and a greedy 3-colouring.

average degree, and Grundy number.

3.2.1 Maximum degree

Recall that Brooks' Theorem states that every connected graph G is $\Delta(G)$-colourable unless G is isomorphic to a complete graph or an odd cycle, in which case G is $(\Delta(G)+1)$-colourable. Jerrum [45] proved the following, relating the mixing number and the maximum degree of a graph.

Theorem 4 ([45])**.** *For any graph G and $k \geq \Delta(G)+2$, G is k-mixing.*

Feghali, Johnson, and Paulusma proved the following analogue to Brooks' Theorem in terms of the reconfiguration graph.

Theorem 5 ([35])**.** *Let G be a connected graph that is not isomorphic to an odd cycle or a complete graph, and let $k \geq \Delta(G)+1$. For a k-colouring α of G, if α is not frozen, then there exists a $\Delta(G)$-colouring γ such that the distance between α and γ in $\mathcal{R}_k(G)$ is $O(n^2)$.*

Furthermore, for $\Delta(G) \geq 3$, the authors showed that the reconfiguration graph $\mathcal{R}_{\Delta(G)+1}(G)$ consists of isolated vertices and at most one non-trivial component that has diameter $O(n^2)$. Bonamy, Bousquet, and Perarnau [7] investigated the proportions of frozen colourings to the total number of colourings. In particular, they showed that the number of frozen colourings is exponentially smaller than the total number of colourings. They also showed that frozen colourings may exist even for graphs of arbitrarily large girth.

3.2.2 Grundy number

Recall the greedy colouring algorithm discussed in Section 2.2. Note that the greedy colouring algorithm may use more colours than the minimum (see Figure 3.1). The *Grundy number* of G, denoted by $\chi_g(G)$, is the maximum k over all vertex orderings for which the greedy colouring algorithm will output a k-colouring of G.

Bonamy and Bousquet proved the following theorem relating the mixing number of a graph and the Grundy number.

Theorem 6 ([4])**.** *For any graph G and $k \geq \chi_g(G)+1$, G is k-mixing and the k-recolouring diameter of G is at most $4 \cdot \chi(G) \cdot n$.*

This result improves the bound given by Jerrum (Theorem 4) due to the following. One can easily see that $\chi_g(G) \leq \Delta(G) + 1$ since each vertex of G has at most $\Delta(G)$ neighbours and so a greedy colouring will not use more than $\Delta(G) + 1$ colours. Note that there exist graphs where this bound is tight. For example, let G be the complete bipartite graph $K_{p,p}$ minus a perfect matching (see Figure 3.6 on page 24). Then $\chi(G) = 2$ but $\chi_g(G) = p = \Delta(G) + 1$ (order the vertices in Figure 3.6 from left to right and top to bottom). Also note that there exist graphs G for which $\chi_g(G)$ is arbitrarily smaller than $\Delta(G)$, for example the star graphs $K_{1,p}$.

3.2.3 Maximum average degree

The *maximum average degree* of a graph G, denoted $\text{mad}(G)$, is the maximum average degree of a non-empty induced subgraph H of G. Formally, $\text{mad}(G)$ is defined as follows.

$$\text{mad}(G) = \max_{\emptyset \neq H \subseteq G} \frac{2|E(H)|}{|V(H)|}$$

It can be shown using Euler's formula that planar graphs, planar graphs of girth 5, and triangle-free planar graphs have maximum average degree strictly less than 6, 7/2, and 4, respectively.

Bousquet and Perarnau proved the following theorem relating the maximum average degree of a graph with its reconfiguration graph.

Theorem 7 ([13])**.** *Let d and k be such that $k \geq d+1$. For every $\epsilon > 0$ and every graph G with mad$(G) = d - \epsilon$, there exists a constant $c := c(d, \epsilon)$ such that the k-recolouring diameter of G is $O(n^c)$.*

This was improved by Feghali [31] who proved a bound of $O(n(\log n)^d)$. This theorem has several implications on Cereceda's conjecture for classes of planar graphs. We will review this now.

3.3 Cereceda's Conjecture

In this this section, we survey results on the reconfiguration graph $\mathcal{R}_k(G)$ where the parameter k depends on the degeneracy of G.

Recall that a graph is d-degenerate if for all subgraphs H of G, H has a vertex with degree at most d. If a graph is d-degenerate, then there exists an ordering of the vertices of G, $v_1, v_2, \ldots, v_n$, such that v_i has at most d neighbours in $v_1, v_2 \ldots, v_{i-1}$ for $1 \leq i \leq n$. We call this ordering a *d-degenerate ordering* of the vertices of G. Note that for a d-degenerate graph G, the greedy colouring algorithm will output a $(d+1)$-colouring of G if a d-degenerate ordering is given.

Cereceda [14] proved that for a d-degenerate graph G and $k \geq 2d+1$, the k-recolouring diameter of G is $O(n^2)$. This was improved by Bousquet and Heinrich [12] who showed that if $k \geq \frac{3}{2}(d+1)$ then the k-recolouring diameter of G is $O(n^2)$. Bousquet and Perarnau [13] showed that for $k \geq 2d+2$ the k-recolouring diameter of G is at most $(d+1) \cdot n$. Bousquet and Bartier [11] proved that for any d-degenerate chordal graph G and $k \geq d+4$, the k-recolour diameter of G is at most $f(\Delta(G)) \cdot n$. It was shown by Dyer [27] and independently by Cereceda, van den Heuvel, and Johnson [15] that for a d-degenerate graph G and $k \geq d+2$, G is k-mixing. However, the k-recolouring diameter exhibited by these constructive proofs is $O(c^n)$ for some constant c. Cereceda [14] conjectured that the exponential upper bound could be improved to a quadratic bound.

Conjecture 2 (Cereceda's Conjecture [14])**.** *For a d-degenerate graph G and $k \geq d+2$, the k-recolouring diameter of G is $O(n^2)$.*

This is an open problem that has become known as Cereceda's Conjecture. If Cereceda's conjecture is proven, this upper bound would be tight due to the following. Bonamy et al. [8] proved a lower bound on the diameter of $\mathcal{R}_{k+1}(G)$ for the path P_n. They gave two 3-colourings of P_n that are at distance $\Omega(n^2)$ in $\mathcal{R}_3(G)$. From this, it is not hard to construct a 2-degenerate chordal graph G and two 4-colourings of G that are at distance $\Omega(n^2)$ (see Figure 3.2). One cannot hope for fewer than $d+2$ colours since, for example, the complete graph K_n is $(n-1)$-degenerate and every n-colouring of K_n is frozen.

Cereceda's conjecture has been proven for 1-degenerate graphs (trees) [8], and for 2-degenerate graphs with maximum degree at most 3 [35]. Cereceda's conjecture is still open for $d \geq 2$ but has been solved for several graph classes, including chordal graphs, graphs of bounded treewidth, and distance-hereditary graphs. Note that chordal graphs are $(\chi(G)-1)$-degenerate since every chordal graph has a perfect elimination ordering which is also a $(\chi(G)-1)$-degenerate ordering. Bonamy et al. [8] proved that for a chordal graph G and $k \geq \chi(G)+1$, G is k-mixing and has k-recolouring diameter $O(n^2)$, proving Cereceda's conjecture for this class (we will return to this in Section 3.5.6).

A *chordal completion* of a graph G is a chordal graph H obtained from G by adding edges. The *treewidth* of a graph G is equal to the minimum of $\omega(H)-1$ over all chordal completions H of G. Note that a graph G with treewidth at most d is d-degenerate since it is the subgraph of a chordal graph with clique number at most $d+1$. Thus, a perfect elimination ordering of H is also a d-degenerate ordering of G. Bonamy and Bousquet [4] proved the following theorem, thereby confirming Cereceda's conjecture for graphs of bounded treewidth.

Theorem 8 ([4]). *For a graph G and $k \geq tw(G)+2$, G is k-mixing and has k-recolouring diameter $O(n^2)$.*

Feghali [29] gave a short proof of this theorem which can be transformed into an algorithm that finds a path between two $tw(G)+2$ colourings of G in quadratic time. Graph classes with bounded treewidth include outerplanar graphs and Apollonian networks, which we define next. A graph is *outerplanar* if it is isomorphic to a plane graph O such that every vertex of O is on the outer face. A graph is an *Apollonian network* if it is isomorphic to a plane graph that can be constructed from a triangle drawn in the plane by repeatedly adding vertices of degree 3 to some triangular face. Note that there exist graphs with bounded degeneracy and unbounded treewidth, for example a $\sqrt{n} \times \sqrt{n}$ grid. We note that a $\sqrt{n} \times \sqrt{n}$ grid is an OAT graph since it can be constructed from an edge by repeatedly adding comparable vertices.

Cereceda's conjecture is open even if the "$O(n^2)$ recolouring diameter" is relaxed to any polynomial recolouring diameter. For many years, the best known bound on the k-recolouring diameter of a d-degenerate graph, for $k \geq d+2$, was $O(k^n)$ as given by Dyer et al. [27]. Recently, Bousquet and Heinrich greatly improved this bound.

Theorem 9 ([12]). *For a d-degenerate graph G and $k \geq d+2$, the k-recolouring diameter of G is $O(n^{d+1})$.*

This implies that for a fixed constant d and $k \geq d+2$, the k-recolouring diameter of a d-degenerate graph is polynomial in n. In particular for 2-degenerate graphs, the 4-recolouring diameter is $O(n^3)$.

3.3.1 Planar graphs

In this section, we discuss results on Cereceda's conjecture relating to planar graphs. It can be shown using Euler's formula that every planar graph is 5-degenerate and every triangle-free planar graph is 3-degenerate. As observed by Bonamy and Bousquet [4], there exist planar graphs that are not 5-mixing and not 6-mixing which is shown by the

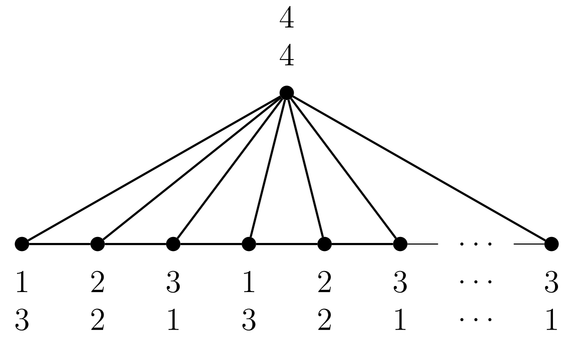


Figure 3.2: A 2-degenerate chordal outerplanar graph and two 4-colourings which are at distance $\Omega(n^2)$ [4].

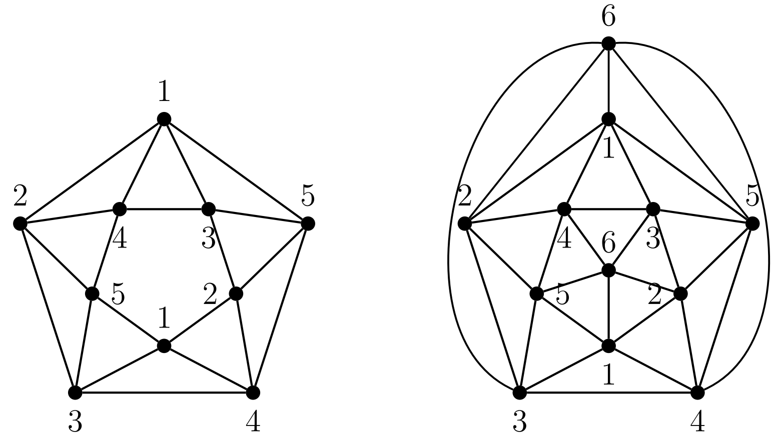


Figure 3.3: A frozen 5-colouring and frozen 6-colouring of a planar graph [14].

frozen colourings in Figure 3.3. Cereceda's conjecture would imply that the 7-recolouring diameter of a planar graph is $O(n^2)$.

The result of Bousquet and Perarnau [13] relating the maximum average degree of a graph and the recolouring diameter of a graph (see Section 3.2) implies that the 8-recolouring diameter of a planar graph is polynomial in n. This bound was improved by Feghali [31] who showed that the 8-recolouring diameter of a planar graph is $O(n(\log n)^7)$. Eiben and Feghali [28] showed that the 7-recolouring diameter of a planar graph is at most $2^{O(\sqrt{n})}$. Feghali [32] showed that the 10-recolouring diameter of a planar graph is at most n^2. This was improved by Dvořák and Feghali [26] who showed that the 10-recolouring diameter of a planar graph is at most $8n$. The results of Bousquet and Heinrich [12] imply that the 7-recolouring diameter of a planar graph is $O(n^6)$ and for $k \geq 9$, the k-recolouring diameter of a planar graph is $O(n^2)$.

As mentioned above, it can be shown using Euler's formula that every triangle-free planar graph (and thus every planar bipartite graph) is 3-degenerate. Bousquet and Heinrich [12] proved that for a planar bipartite graph G, $\mathcal{R}_5(G)$ has diameter $O(n^2)$, proving Cereceda's conjecture for this class of graphs. Bousquet and Perarnau [13] proved that the 6-recolour diameter of a triangle-free planar graph is polynomial in the number of vertices. This was improved by Feghali [31] (see Section 3.2) who proved a bound of $O(n(\log n)^5)$. The same theorem of Feghali proves Cereceda's conjecture for planar graphs of girth 5 since these graphs are 3-degenerate and have maximum average degree at most 7/2.

3.4 Complexity of Reconfiguration

In this section, we survey results regarding the complexity of the k-Colour Path and k-Colour Bounded Path problems (see Section 3.1 for the problem statements).

There is a general pattern that is followed between the complexity of a source problem and the complexity of finding a path between two solutions in the reconfiguration graph. In particular, if a source problem is NP-complete then the problem of finding a path between two solutions in the reconfiguration graph is PSPACE-complete. A known exception that breaks this pattern is the 3-colouring problem. That is the decision problem of determining whether a given graph G is 3-colourable. We will see that the corresponding problem on the reconfiguration graph 3-Colour Path was shown to be solvable in polynomial time.

3.4.1 Complexity of k-Colour Path

Recall that k-Colour Path is the problem, given a k-colourable graph G and two k-colourings α and β of G, determine whether there is a reconfiguration sequence between α and β in $\mathcal{R}_k(G)$.

The complexity class PSPACE is the class of decision problems that can be solved by a deterministic Turing machine using an amount of space that is polynomial in the size of the input. Similarly, the class NPSPACE is the class of decision problems that can be solved by a non-deterministic Turing machine using an amount of space that is polynomial in the size of the input (see for example [57]). Cereceda [14] showed that k-Colour Path and k-Mixing are in fact in PSPACE. It is actually shown that k-Colour Path is in NPSPACE, and by Savitch's Theorem [56], which states that PSPACE=NPSPACE, k-Colour Path is in PSPACE.

For $k = 2$, the k-Colour Path problem is trivial (see Section 3.5.1). Cereceda, van den Heuvel, and Johnson [17] examined the 3-Colour Path problem. The authors proved that the decision problem 3-Colour Path is solvable in polynomial time. The proof characterizes the instances for which a reconfiguration sequence exists and either exhibits a reconfiguration sequence or exhibits a structure for which no reconfiguration sequence can exist. The authors also prove that the diameter of every component of $\mathcal{R}_3(G)$ is $O(n^2)$, and there exist 3-colourable graphs G where a component of $\mathcal{R}_3(G)$ has diameter $\Omega(n^2)$.

Bonsma and Cereceda [9] showed that for all $k \geq 4$, k-Colour Path is PSPACE-complete. The authors also showed that k-Colour Path remains PSPACE-complete for bipartite graphs, for planar graphs and $4 \leq k \leq 6$, and for planar bipartite graphs and $k = 4$.

They also defined a class of graphs $\{G_{N,k} \mid k \geq 4, N \in \mathbb{N}\}$ such that $G_{N,k}$ has size $O(N^2)$ and $G_{N,k}$ has two k-colourings α and β in the same component of $\mathcal{R}_k(G_{N,k})$ such that the distance between α and β is $\Omega(2^N)$. The fact that there exist colourings at superpolynomial distances is not surprising since the decision problem k-Colour Path is PSPACE-complete.

3.4.2 Complexity of k-Colour Bounded Path

Recall that k-Colour Bounded Path is the problem, given a graph G, two k-colourings of G, α and β, and a positive integer l, determine whether there is a path of length at most l between α and β in $\mathcal{R}_k(G)$. Johnson, Kratsch, Kratsch, Patel, and Paulusma [46] showed that 3-Colour Bounded Path can be solved in $O(n+m)$ time, and their algorithm finds a shortest path between two 3-colourings, generalizing the work of Cereceda et al. [17] on the 3-Colour Path problem. A polynomial time algorithm cannot be expected for k-Colour Bounded Path and $k \geq 4$, since using a reduction from k-Colour Path, it can be observed that in general, k-Colour Bounded Path is PSPACE-hard. Note that there are at most k^n distinct k-colourings of a graph, so a path between two k-colourings exists if and only if a path exists of length at most k^n. This only establishes weak PSPACE-hardness since a chosen value of $l = k^n$ is exponential in the input size. Bonsma, Mouawad, Nishimura, and Raman [10] showed that k-Colour Bounded Path is NP-complete when l is encoded in unary, and thus is strongly NP-hard.

A *parameterized problem* is a decision problem in which every problem instance I has an associated integer parameter p. A parameterized problem is *fixed parameter tractable* (FPT) if every instance I can be solved in $O(f(p)|I|^c)$ time where f is a computable function that depends only on p and where c is a constant independent of p (see for example [25] [36]). It was shown independently by Bonsma et al. [10] and Johnson et al. [46] that k-Colour Bounded Path is FPTwhen parameterized by $k + l$, although the algorithms given in these two papers are quite different. Bonsma et al. [10] proved that k-Colour Bounded Path can be solved in $O((k \cdot l)^{l^2+l} \cdot ln^2)$ and Johnson et al. [46] proved that k-Colour Bounded Path can be solved in time $O(2^{k(l+1)} \cdot l^l \cdot \text{poly}(n))$.

3.5 Recolouring and the chromatic number

Now we turn to reconfiguration results where the parameter k depends on the chromatic number $\chi(G)$. As we will see in Section 3.5.1, the case $k = \chi(G)$ is well-understood. We therefore focus on the case $k \geq \chi(G) + 1$. There are many results here, and a major motivation of this thesis was to find a graph class that unifies and generalizes many of these results. We therefore not only survey the results but also study how the respective graph classes relate to OAT graphs. Figure 3.4 gives an overview of the graph classes.

3.5.1 Recolouring with the minimum number of colours

Here we discuss the reconfiguration graph $\mathcal{R}_k(G)$ for $k = \chi(G)$. We first consider the case when $k = \chi(G) = 2$, so G is bipartite. The following observations are due to Cereceda, van den Heuvel, and Johnson [15]. For any graph G with $\chi(G) = 2$, $\mathcal{R}_2(G)$ is disconnected. If G is connected, then $\mathcal{R}_2(G)$ consists of two frozen colourings. If G is not connected, then

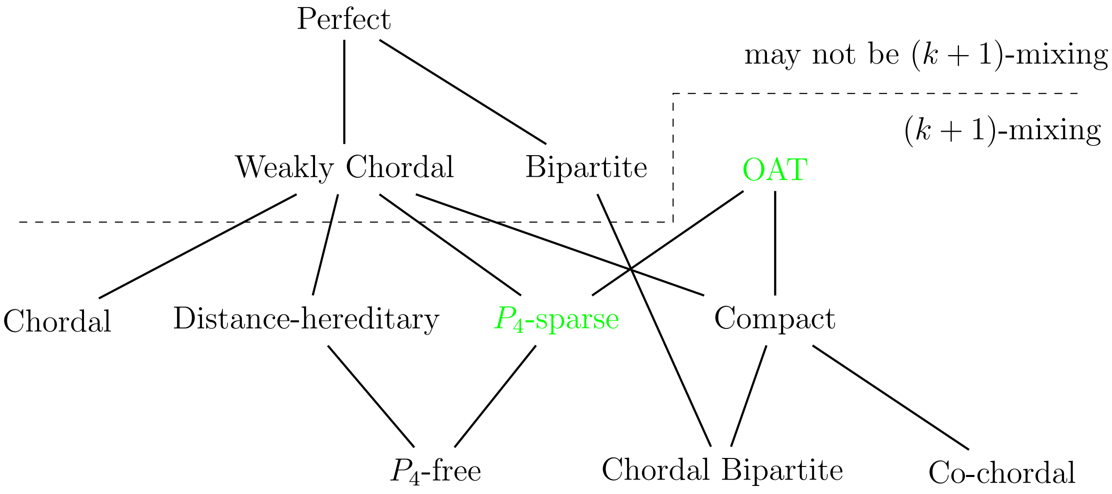


Figure 3.4: Classes of graphs for which it is known whether a k-colourable graph in the class is $(k+1)$-mixing. Reconfiguration results for OAT graphs and P_4-sparse graphs are new results of this thesis.

there is a path between two colourings α and β in $\mathcal{R}_2(G)$ if and only if for every non-trivial connected component H of G, α and β agree on colour for every vertex of H. Furthermore, suppose G has p isolated vertices and q non-trivial connected components. Then $\mathcal{R}_2(G)$ has 2^q connected components, each of which is a p-dimensional hypercube.

Next, consider the case when $k = \chi(G) = 3$. For any graph G with $\chi(G) = 3$, $\mathcal{R}_3(G)$ is disconnected [15]. For every $k \geq 4$, there exist k-chromatic graphs G for which $\mathcal{R}_k(G)$ is connected and for which $\mathcal{R}_k(G)$ is disconnected. It is easy to see that for $n \geq 2$, every n-colouring of K_n is a frozen colouring. Cereceda, van den Heuvel, and Johnson [15] give a family of graphs $\{H_k \mid k \in \mathbb{N}\}$ such that H_k is k-chromatic and k-mixing (see Figure 3.5).

Many classes of graphs discussed in this thesis are subclasses of perfect graphs. As shown in the next section, for a k-colourable perfect graph G, it is not necessarily true that $\mathcal{R}_{k+1}(G)$ is connected. This began an investigation into which subclasses of perfect graphs have this special property. The rest of Section 3.5 is dedicated to surveying the classes of perfect graphs for which this property has been proved or disproved.

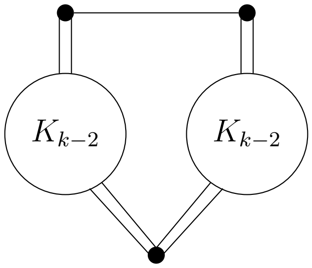


Figure 3.5: A k-chromatic graph that is k-mixing for $k \geq 4$ [14].

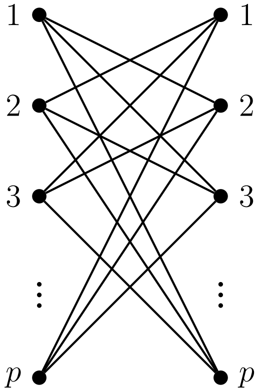


Figure 3.6: A frozen p-colouring of L_p, a complete bipartite graph minus a matching. [15]

3.5.2 Bipartite graphs

Cereceda, van den Heuvel, and Johnson [15] gave a family of bipartite graphs $\{L_p \mid p \geq 3\}$ where L_p is obtained from the complete bipartite graph $K_{p,p}$ by deleting the edges of a perfect matching. They prove that for all $k \neq p$, L_p is k-mixing, and for $k = p$, L_p is not k-mixing. See Figure 3.6 for a frozen p-colouring of L_p. We note that L_p is not an OAT graph for any p, but every bipartite graph is an induced subgraph of an OAT graph.

Observation 6. *For $p \geq 3$, L_p is not an OAT graph.*

Proof. We show that there is no operation that could have been used to construct L_p from smaller graphs. Since L_p is connected, it could not have been constructed from the disjoint union operation. Note that co-L_P consists of two disjoint cliques with the edges of a matching between them. Since co-L_p is connected, L_p could not have been constructed from the join operation. Since every vertex of L_p has a distinct set of $p - 1$ neighbours,

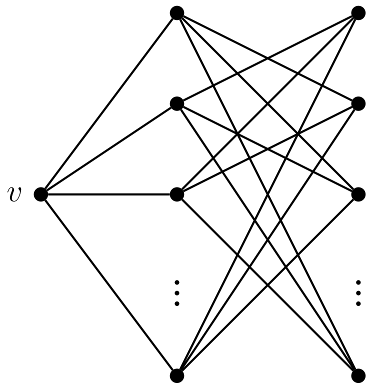


Figure 3.7: An OAT graph L_p^+ with L_p as an induced subgraph.

L_p could not have been constructed by adding a comparable vertex. Finally, since L_p is biconnected, it could not have been constructed from attaching a clique. □

Observation 7. *Every bipartite graph is an induced subgraph of a bipartite OAT graph.*

Proof. Let B be a bipartite graph. Suppose B has p vertices in one bipartite set and q vertices in the other bipartite set. To construct a bipartite OAT graph B^+ that contains B as an induced subgraph, first construct the star graph $K_{1,p}$ with center v. The p vertices of $K_{1,p} \setminus v$ will correspond to one bipartite set of B. Since v is adjacent to all other vertices, the q vertices in the other bipartite set of B can be added to $K_{1,p}$ as comparable vertices. See Figure 3.7 for a bipartite OAT graph L_p^+ that contains L_p as an induced subgraph. □

The reader may wonder how it is possible that for $p \geq 3$, L_p has a frozen p-colouring, while L_p^+ contains L_p as an induced subgraph and is p-mixing since it is a 2-colourable OAT graph. The reason is that not all p-colourings of L_p can be extended to p-colourings of L_p^+. Note that in the frozen colouring of L_p, each bipartite set uses all p colours on its vertices. Since u is adjacent to every vertex in one of the bipartite sets, the frozen colouring of L_p can no longer extend to a p-colouring of L_p^+ since each of the p colours would appear on the neighbourhood of u.

Cereceda, van den Heuvel, and Johnson [16] examined the 3-Mixing problem for bipartite graphs, since for a 3-chromatic graph G, $\mathcal{R}_3(G)$ is not connected. The authors characterize the bipartite graphs for which $\mathcal{R}_3(G)$ is connected and show that the decision problem 3-Mixing is coNP-complete. The authors also show that when the input graph is restricted to be planar bipartite, the 3-Mixing problem is solvable in polynomial time.

3.5.3 P_4-free and P_5-free graphs

Bonamy and Bousquet [4] proved that for a graph G, if k is larger than the Grundy number of G, then G is k-mixing and the k-recolour diameter of G is at most $4 \cdot \chi(G) \cdot n$ (see Section 3.2). It is well known that the greedy colouring algorithm will output a $\chi(G)$ colouring of a P_4-free graph G for any vertex ordering [18]. Thus, for any P_4-free graph, the Grundy number is equal to the chromatic number. Bonamy and Bousquet [4] then proved the following result with an improved bound on the recolouring diameter for P_4-free graphs.

Theorem 10 ([4])**.** *If G is a k-colourable P_4-free graph, then $\mathcal{R}_{k+1}(G)$ is connected with diameter at most $\chi(G) \cdot n$.*

Note that a similar result does not hold for P_t-free graphs for any $t \geq 6$. For all $p \geq 3$, the graphs L_p introduced by Cereceda, van den Heuvel, and Johnson [15] are P_6-free, are 2-colourable, and are not p-mixing.

In the case of P_5-free graphs, Bonamy and Bousquet [4] mistakenly thought[1] to have given a 4-colourable P_5-free graph G and a frozen 5-colouring of G. This graph is in fact not P_5-free as illustrated in Figure 3.8. In addition, Bonamy and Bousquet mistakenly thought[1] to have given a family of P_5-free graphs $\{G_k \mid k \geq 3\}$ where G_k is $(k+1)$-colourable and has a frozen $2k$-colouring. The graph G_k also contains an induced P_5 for every $k \geq 3$ (see Figure 3.8 for an induced P_5 in G_3). It is again an open question whether a k-colourable P_5-free graph is $(k+1)$-mixing. We note that this question is open for $k \geq 3$ since every 2-colourable P_5-free graph is chordal bipartite.

3.5.4 P_4-reducible and P_4-sparse graphs

Here we recall two classes of graphs that generalize the class of P_4-free graphs. Jamison and Olario [41] introduced the class of *P_4-reducible* graphs, which are the graphs where each vertex is in at most one induced P_4. Hoàng [39] further generalized this class to the *P_4-sparse* graphs, which are the graphs where for every set of 5 vertices, there is at most one induced P_4.

Jamison and Olario [43] prove that P_4-sparse graphs are exactly the class of graphs that can be constructed from single vertex graphs with the join operation, the disjoint union operation and a third operation defined as follows (note that we only use the following operation in the proof of Lemma 11).

[1] confirmed in private communication.

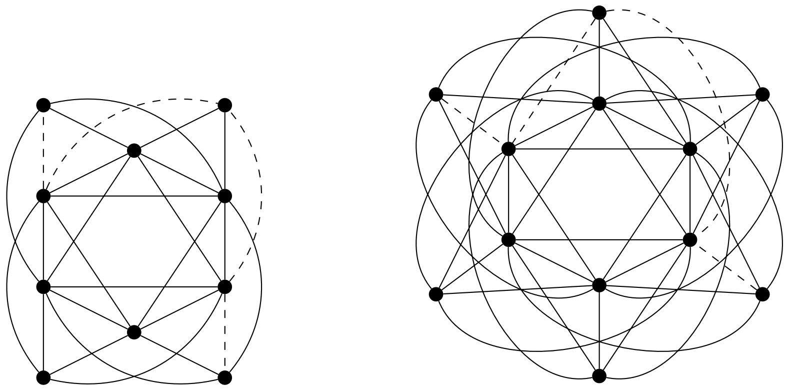


Figure 3.8: Graphs mistaken to be P_5-free by Bonamy and Bousquet. [4]. An induced P_5 is marked with dashed edges.

Let $G_1 = (V_1, \emptyset)$ and $G_2 = (V_2, E_2)$ be vertex-disjoint P_4-sparse graphs with $V_2 = \{v\} \cup K \cup R$ such that:

- $|K| = |V_1| + 1 \geq 2$.
- K is a clique.
- R is joined to K and every vertex in R is non-adjacent to v.
- There exists a vertex $v' \in K$ such that $N_{G_2}(v) = \{v'\}$ or $N_{G_2}(v) = K \setminus \{v'\}$.

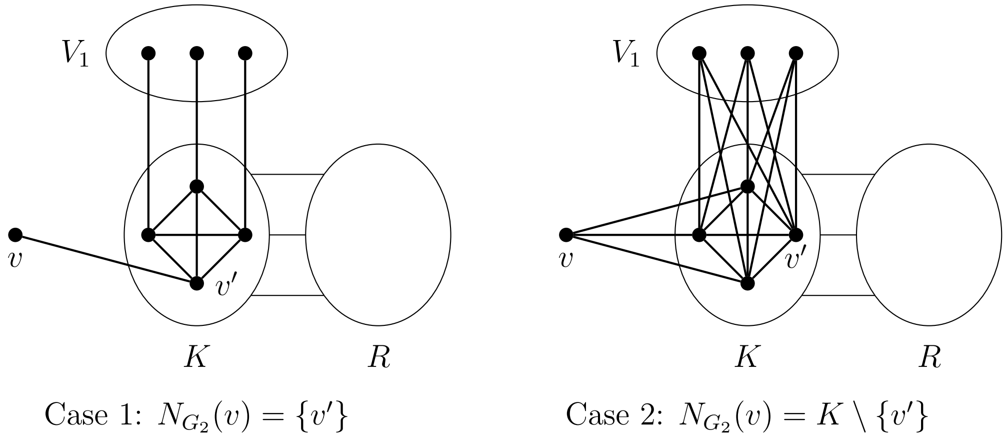
Choose a bijection $f : V_1 \to K \setminus \{v'\}$. Define the third operation on G_1 and G_2 to be the graph $(V_1 \cup V_2, E_2 \cup E')$ where

$$E' = \begin{cases} \{xf(x) \mid x \in V_1\} & \text{if } N_{G_2}(v) = \{v'\} \\ \{xz \mid x \in V_1, z \in K \setminus \{f(x)\}\} & \text{if } N_{G_2}(v) = K \setminus \{v'\} \end{cases}$$

See Figure 3.9 for an illustration of this operation. We note that every P_4-sparse graph (and thus every P_4-reducible graph) is an OAT graph.

Theorem 11. *Every P_4-sparse graph is an OAT graph.*

Proof. Let G be a P_4-sparse graph. The proof is by induction on the number of vertices of G. The claim clearly holds for single vertex graphs. Suppose G was constructed by the join



Case 1: $N_{G_2}(v) = \{v'\}$ Case 2: $N_{G_2}(v) = K \setminus \{v'\}$

Figure 3.9: The third operation defining P_4-sparse graphs.

or disjoint union of the P_4-sparse graphs G_1 and G_2. Then by the induction hypothesis, G_1 and G_2 are OAT graphs and it follows that G is also an OAT graph.

Now suppose G was constructed by the third operation of P_4-sparse graphs on G_1 and G_2. By the induction hypothesis, G_2 is an OAT graph. There are two cases to consider depending on whether $N_{G_2}(v) = \{v'\}$ or $N_{G_2}(v) = K \setminus \{v'\}$. First suppose $N_{G_2}(v) = \{v'\}$. Then each vertex of V_1 is a pendant vertex in G and is comparable to v'. Therefore, G is an OAT graph since it can be constructed from G_2 by adding each vertex of G_1 as a comparable vertex. Now suppose $N_{G_2}(v) = K \setminus \{v'\}$. Then each vertex $x \in V_1$ is comparable to $f(x)$ in G since K is a clique. Once again, G is an OAT graphs since it can be obtained from G_2 by adding each vertex of G_1 as a comparable vertex. □

Jamison and Olario [42] gave the complete forbidden induced subgraph characterization for P_4-sparse graphs. The class of (P_5, co-P_5, C_5, P, co-P, fork, co-fork)-free graphs (see Figure 3.10) is equivalent to the class of P_4-sparse graphs. Feghali and Fiala left as an open problem whether $\mathcal{R}_{k+1}(G)$ is connected with diameter $O(n^2)$ for a k-colourable (P_5, co-P_5, C_5)-free graph G. We answer this question in the positive for the subclass of P_4-sparse graphs.

3.5.5 Distance-hereditary graphs

Here we discuss the relationship between OAT graphs and distance-hereditary graphs and survey results on the reconfiguration graph of a distance-hereditary graph. Recall that a

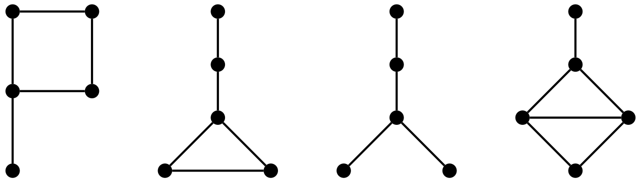


Figure 3.10: The graphs P, co-P, fork and co-fork.

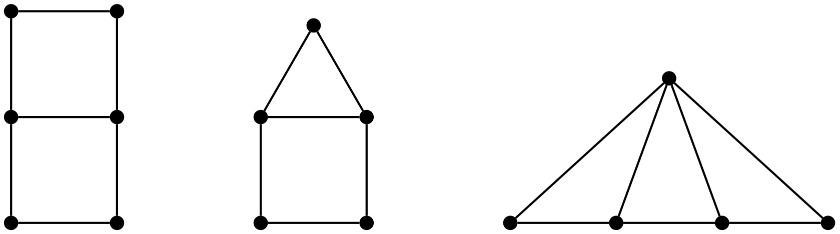


Figure 3.11: The domino, co-P_5, and gem graphs.

graph G is *distance-hereditary* if for all connected induced subgraphs H of G, and any two vertices x, y of H, the distance between x and y in H is the same as the distance between x and y in G.

Although both OAT graphs and distance-hereditary graphs extend the class of P_4-free graphs, both classes contain graphs that are not members of the other class. One can show from the definition that every distance-hereditary graph does not contain a domino, co-P_5, or gem graph as an induced subgraph [3] (see Figure 3.11). However, each of these graphs is an OAT graph since they can be constructed from a clique by adding comparable vertices. We also note that the class of OAT graphs does not extend the class of distance-hereditary graphs due to the following. The graph in Figure 3.12 is a distance-hereditary graph since it can be constructed from P_4 by adding four true twins. However, this graph is not an OAT graph since none of the four operations defining OAT graphs could have been applied to build it.

The class of distance-hereditary graphs is a known generalization of P_4-free graphs

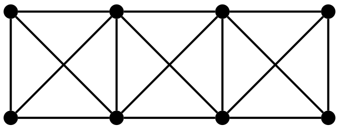


Figure 3.12: A graph that is chordal, distance-hereditary, and not OAT.

(see Section 2.3 for definition and characterization). Bonamy and Bousquet [5] proved the following result.

Theorem 12 ([5])**.** *If G is a k-colourable distance-hereditary graph, then $\mathcal{R}_{k+1}(G)$ is connected with diameter $O(k \cdot \chi(G) \cdot n^2)$.*

3.5.6 Chordal and chordal bipartite graphs

Here we survey results on the reconfiguration graph of chordal and chordal bipartite graphs. We also examine the relationship between these graph classes and OAT graphs.

We first note that the class of OAT graphs extends the class of chordal bipartite graphs due to the following. Bonamy et al. [8] proved that every chordal bipartite graph can be constructed from a set of one or more isolated vertices by adding a pendant vertex or adding a comparable vertex. Since adding a pendant vertex to any connected graph with at least one edge is a special case of adding a comparable vertex, it follows that the operations defining OAT graphs are sufficient for building such graphs.

Bonamy, Johnson, Lignos, Patel, and Paulusma [8] introduced a class of graphs called k-colour-dense graphs and proved that this class of graphs contains the class of chordal graphs and chordal bipartite graphs. The following definition of k-colour-dense is given in [8]. For a fixed integer $k \geq 1$, a k-colourable graph G is *k-colour-dense* if either

1. G is the disjoint union of complete graphs, each of which has at most k vertices, or
2. G has a separator S where $G \setminus S$ has components D and D' with vertices $u \in D$ and $v \in D'$ such that
   (a) $|D| = 1$ or $|D \cup S| \leq k$, and
   (b) $S \subseteq N(v)$, and
   (c) the graph obtained from deleting u and v, and adding a vertex x adjacent to $N(u) \cup N(v)$, is k-colour-dense.

Bonamy et al. [8] proved the following results for k-colour-dense graphs.

Theorem 13 ([8])**.** *If G is a k-colour-dense graph, then $\mathcal{R}_{k+1}(G)$ has diameter at most $2n^2$.*

The authors show that every k-colourable chordal graph is k-colour-dense and every chordal bipartite graph is 2-colour-dense. Furthermore, for each $k \geq 2$ and $n \geq k$, there exists a k-colourable chordal graph on n vertices that has $(k+1)$-recolour diameter $\Theta(n^2)$.

We note that there exist OAT graphs that are not k-colour-dense graphs. The graph in Figure 3.12 is chordal, and therefore k-colour-dense, but is not an OAT graph. We do not know whether every OAT graph is a k-colour-dense graph and leave this as an open problem.

3.5.7 Compact graphs

Recently, Feghali and Fiala [33] examined a subclass of weakly chordal graphs called *compact graphs* defined below. A *2-pair* $\{u, v\}$ is a pair of non-adjacent vertices such that every chordless path between u and v has exactly two edges. For a 2-pair $\{u, v\}$, let $S(u, v) = N(u) \cap N(v)$ and let C_v denote the component of $G \setminus S(u, v)$ that contains the vertex v.

Definition 5 ([33])**.** *A weakly chordal graph G is* compact *if every subgraph H of G either*

- *is a complete graph, or*
- *contains a 2-pair $\{x, y\}$ such that $N_H(x) \subseteq N_H(y)$, or*
- *contains a 2-pair $\{x, y\}$ such that $C_x \cup S(x, y)$ in H induces a clique on at most three vertices.*

See also Figure 3.13. Feghali and Fiala proved that every co-chordal graph and every 3-colourable (P_5, C_5, co-P_5)-free graph is compact [33]. We note that the class of OAT graphs is a strict generalization of compact graphs.

Theorem 14. *Every compact graph is an OAT graph but not every OAT graph is a compact graph.*

Proof. Let G be a compact graph. The proof is by induction on the number of vertices of G. If G is a complete graph, then clearly G is an OAT graph. If not, then suppose G contains a 2-pair $\{x, y\}$ such $N_G(x) \subseteq N_G(y)$. Then x is a vertex comparable to y. Since the conditions defining compact graphs must hold for all induced subgraphs, $G \setminus x$ is also a compact graph. By the induction hypothesis, $G \setminus x$ is an OAT graph. Then we can construct G from $G \setminus x$ by adding back the comparable vertex x and the appropriate edges.

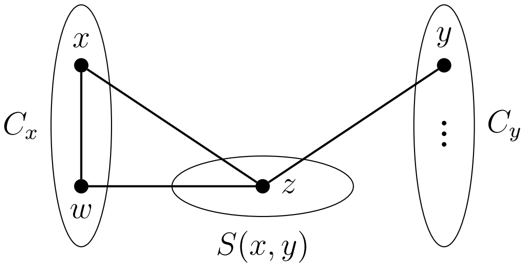


Figure 3.13: A compact graph that is not a complete graph and that does not have a comparable vertex [33].

Lastly, suppose G contains a 2-pair $\{x,y\}$ such that $C_x \cup S(x,y)$ induces a clique on at most three vertices in G. If $S(x,y)$ contains two vertices then x is comparable to y and the proof follows from the argument above. Now assume $S(x,y)$ contains exactly one vertex z and let w be the unique vertex in $N_G(x) \setminus N_G(y)$ (see Figure 3.13). Then $G \setminus \{x,w\}$ is a compact graph and by the induction hypothesis, $G \setminus \{x,w\}$ is an OAT graph. Then G is an OAT graph since it can be constructed from $G \setminus \{x,w\}$ by attaching the complete graph on two vertices $\{x,w\}$ to the vertex z. Therefore every compact graph is an OAT graph.

Next we show that there are infinitely many OAT graphs that are not compact graphs. Consider the infinite family of graphs constructed by attaching a complete graph with more than three vertices to some vertex of another complete graph with more than three vertices. By definition, every graph in this class is an OAT graph. Note that any graph in this class does not satisfy any of the three requirements in the definition of compact graphs.

We consider another example showing that the class of OAT graphs contains graphs which are not compact. Note that since weakly chordal graphs are perfect, by definition compact graphs are perfect. The graph in Figure 1.3 is not perfect but is an OAT graph since it can be constructed from adding three comparable vertices to a clique of size three. Therefore, not all OAT graphs are compact graphs. □

Feghali and Fiala [33] prove the following result on the reconfiguration graph of a compact graph.

Theorem 15 ([33])**.** *If G is a k-colourable compact graph, then $\mathcal{R}_{k+1}(G)$ is connected with diameter at most $2n^2$.*

We note that our Theorem 1 generalizes this.

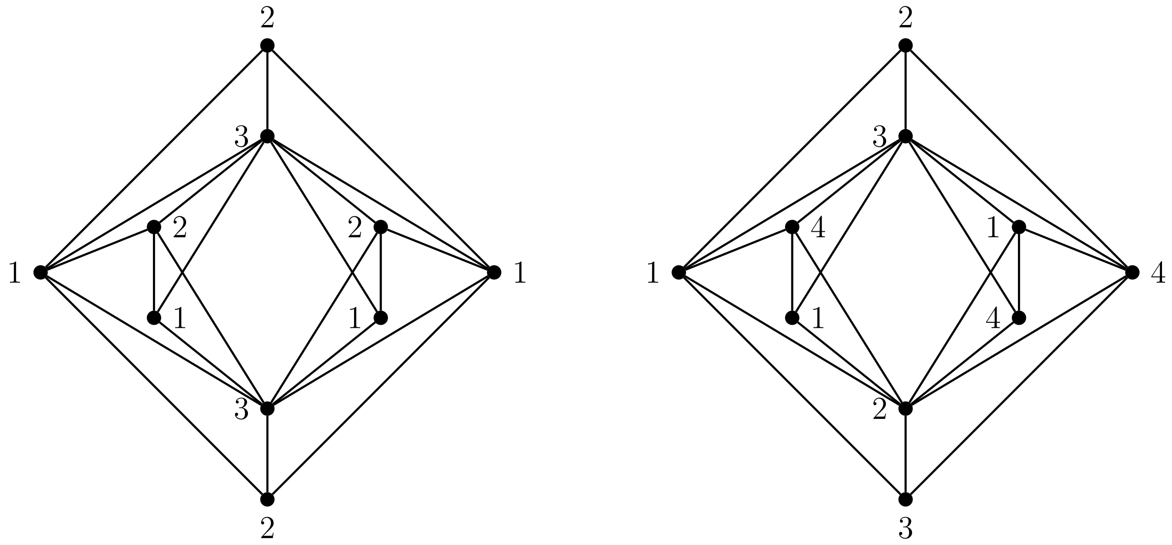


Figure 3.14: A 3-colouring of a weakly chordal graph and a frozen 4-colouring [33].

3.5.8 Weakly chordal graphs

Feghali and Fiala [33] investigated the reconfiguration graph for the class of weakly chordal graphs. They found an infinite family $\{G_k \mid k \geq 3\}$ of k-colourable weakly chordal graphs where $\mathcal{R}_{k+1}(G_k)$ has an isolated vertex. See Figure 3.14 for a 3-colouring of G_3 and a frozen 4-colouring of G_3. To see that this colouring is indeed frozen, notice that for any vertex v, every colour either appears on the neighbourhood of v or is the colour of v.

Chapter 4

Recolouring OAT Graphs

In this chapter, we show that the $(k+1)$-recolouring diameter of a k-colourable OAT graph is $O(n^2)$. Our strategy uses a *canonical* $\chi(G)$*-colouring* as a central vertex in the reconfiguration graph $\mathcal{R}_{k+1}(G)$. For any two colourings α and β in $\mathcal{R}_{k+1}(G)$, we show how to transform both into the canonical $\chi(G)$-colouring γ by recolouring each vertex at most $2n$ times. Then to transform α to β, follow the steps from α to γ and then follow the steps from β to γ in reverse.

In Section 4.1 we define the canonical χ-colouring of an OAT graph. In Section 4.2 we discuss the Renaming Lemma, a method for finding a reconfiguration sequence between two colourings of a graph, both of which induce the same partition of vertices into colour classes. In Section 4.3 we prove our main result, that the diameter of $\mathcal{R}_{k+1}(G)$ for a k-colourable OAT graph G is $O(n^2)$.

We also note a lower bound on the diameter of $\mathcal{R}_{k+1}(G)$. The path P_n for all $n \geq 1$ is an OAT graph with $\chi(P_n) = 2$. Bonamy et al. [8] proved that $\mathcal{R}_3(P_n)$ has diameter $\Omega(n^2)$. Thus, for general k-colourable OAT graphs G, the diameter of $\mathcal{R}_{k+1}(G)$ is $\Omega(n^2)$.

4.1 The canonical χ-colouring

Let $\mathcal{S}$ be a set of k colours and let $\alpha : V(G) \to \mathcal{S}$ be a k-colouring of G. The set $\mathcal{S}$ is called the *set of permissible colours* for α and we denote $\mathcal{S}$ by $\mathcal{S}(\alpha)$ when α is not clear from the context. We also call α an $\mathcal{S}$-colouring when wanting to emphasize its set of permissible colours. The reason behind using this more general notation has to do with the names of the colours and can be understood from the following example. Suppose G is constructed by the join of L and R. Suppose (to be concrete) that we have colourings α and β of G on $k = 5$ colours. To reconfigure α to β, we will recurse on L and R. Now, it may happen that α colours the vertices of L with $\{1, 4\}$ and β colours the vertices of L with $\{2, 5\}$. Both α and β are 2-colourings of L, but we use this more general notation to distinguish these.

Next we define the reconfiguration graph of the $\mathcal{S}$-colourings for some fixed set $\mathcal{S}$. Let $\mathcal{R}_{\mathcal{S}}(G)$ be the graph whose vertices are the $\mathcal{S}$-colourings of G such that two vertices of $\mathcal{R}_{\mathcal{S}}(G)$ are adjacent if and only if they differ by colour on exactly one vertex. By contrast, the definition of $\mathcal{R}_k(G)$ assumes that $\mathcal{S} = \{1, 2, \ldots, k\}$. If $|\mathcal{S}| = k$ then $\mathcal{R}_{\mathcal{S}}(G)$ is isomorphic to $\mathcal{R}_k(G)$.

Let $\mathcal{C}(\alpha)$ be the set of colours c such that $\alpha(v) = c$ for some vertex $v \in V(G)$. Thus $\mathcal{C}(\alpha) \subseteq \mathcal{S}(\alpha)$ but they need not be equal. We say that the colour c *appears* in α if $c \in \mathcal{C}(\alpha)$. We say that a colouring α of G can be *transformed* into a colouring β of G in $\mathcal{R}_{\mathcal{S}}(G)$ if there is a path from α to β in $\mathcal{R}_{\mathcal{S}}(G)$. Let H be a subgraph of G. Let n_H denote the number of vertices of H. The *projection* of α onto H is the colouring $\alpha_H : V(H) \to \mathcal{S}(\alpha_H)$ where $\alpha_H(v) = \alpha(v)$ for all $v \in V(H)$. There is usually a natural way to select the set of permissible colours $\mathcal{S}(\alpha_H)$, but it will be specified if it is not clear from the context.

One way that clearly encodes how to construct an OAT graph is by using a tree structure.

Definition 6. *Given an OAT graph G, a* build-tree *of G, denoted $BT(G)$, is a rooted tree that encodes how to construct G, and has the following properties.*

1. *The root node of $BT(G)$ is G,*
2. *the leaf nodes of $BT(G)$ are exactly the single vertices of G, and*
3. *the internal nodes of $BT(G)$ each represent an operation that takes as input the graphs corresponding to exactly two child nodes.*

We note that a build-tree of G is not unique. For example, take the OAT graph G constructed by attaching a complete graph to a vertex of another complete graph. The graph G could have also been constructed by joining a single vertex to the disjoint union of two complete graphs.

A build-tree leads to a naturally defined ordering of the vertices of G as follows. Starting from the leftmost leaf node of $BT(G)$ to the rightmost leaf node, order the vertices of G from v_1 to v_n. For a given build-tree, we fix this ordering of vertices to force uniqueness

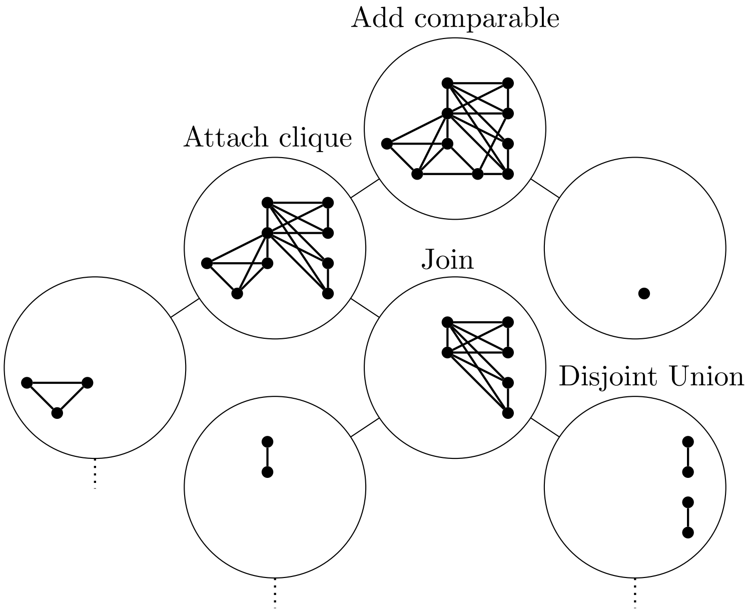


Figure 4.1: An example of an OAT graph G and a build tree of G.

on the *canonical* $\chi(G)$*-colouring of* G, defined next. In the case when a complete graph Q with t vertices is attached to a vertex of an OAT graph H, we repeatedly use the join operation to construct Q and also consider the vertices of Q to be given in some ordering $\{q_1, q_2, \dots q_t\}$. See Figure 4.1 for an example of an OAT graph G and a build tree of G.

Definition 7. *Let G be an OAT graph and let $\mathcal{C}$ be an ordered set of $\chi(G)$ colours. Fix a build-tree σ of G. The* canonical χ-colouring of G *with respect to $\mathcal{C}$ and σ is the $\chi(G)$-colouring of G constructed recursively through σ as follows.*

1. *If G is a single vertex v, then v is coloured with the first colour of $\mathcal{C}$.*

2. *If G is the disjoint union of L and R, then take a canonical χ-colouring of L with respect to the first $\chi(L)$ colours of $\mathcal{C}$ and a canonical χ-colouring of R with respect to the first $\chi(R)$ colours of $\mathcal{C}$.*

3. *If G is the join of L and R, then take a canonical χ-colouring of L with respect to the first $\chi(L)$ colours in $\mathcal{C}$ and take a canonical χ-colouring of R with respect to the next $\chi(R)$ colours in $\mathcal{C}$ (this is possible since $|\mathcal{C}| \geq \chi(L) + \chi(R)$).*

4. If G is constructed by adding a comparable vertex u to a vertex v of a graph H, then take a canonical χ-colouring of H with respect to $\mathcal{C}$ and colour u the same colour as v.

5. If G is constructed by attaching a complete graph Q to a vertex v of a graph H, then take a canonical χ-colouring of H with respect to the first $\chi(H)$ colours of $\mathcal{C}$. Consider the vertices of Q as $\{q_1, q_2, \ldots\}$. Colour the vertices $q_1, q_2, \ldots$ of Q in order with the first $|Q|$ colours of $\mathcal{C} \setminus c$ where c is the colour given to v in the canonical χ-colouring of H.

Note that the canonical χ-colouring of G with respect to $\mathcal{C}$ and σ is unique since by induction, at each step in the construction, there is no choice on which colour a vertex is assigned. For the rest of this section, we assume the build-tree σ of G is fixed.

4.2 The Renaming Lemma

Our proofs use induction to recolour the subgraphs that build up the OAT graph in a fixed construction. There are generally two steps to these proofs. The first step is to recolour the vertices so that the partition of vertices into colour classes is the same as the target colouring, namely the canonical χ-colouring. The second step is to *rename* these colours so that the correct colour appears on the correct colour class. We rely on the *Renaming Lemma* which states that once the vertices are partitioned into the desired colour classes, we can rename each colour class to the desired colour by recolouring each vertex at most twice.

The Renaming Lemma is an adaptation of an idea that is used in token swapping. It was discovered by Akers and Krishnamurthy [1], independently by Portier and Vaughan [54], and later by Pak [53]. It was also rediscovered by Bonamy and Bousquet [4] who rephrased the lemma in terms of recolouring complete graphs. Our statement is expressed more generally.

Lemma 1 (Renaming Lemma [4])**.** *If α and β are two k-colourings of G that induce the same partition of vertices into colour classes, and if $\mathcal{S}$ is a set of at least $k+1$ colours such that the permissible colours $\mathcal{S}(\alpha)$ and $\mathcal{S}(\beta)$ are each a subset of $\mathcal{S}$, then α can be transformed into β in $\mathcal{R}_{\mathcal{S}}(G)$ by recolouring each vertex at most 2 times.*

The proof of Lemma 1 uses directed graphs, and so we briefly review the notation that is used. Let G be a directed graph. Each edge of G is considered as an ordered pair

$e = (u, v)$ where e is directed from u into v. The *indegree* of a vertex $v \in V(G)$, denoted $d^-(v)$, is the number of edges of G that are directed into v. The *outdegree* of a vertex $v \in V(G)$, denoted $d^+(v)$, is the number of edges of G that are directed from v.

Proof of Lemma 1. Let $V_1, V_2, \ldots, V_k$ be the partition of the vertices of G into k colour classes induced by both α and β. Let Q be the complete graph on k vertices $\{q_1, q_2, \ldots q_k\}$ and let α_Q (resp. β_Q) be the colouring of Q where q_i is coloured the same as each vertex of V_i in α (resp. β) for all $i = 1 \ldots k$. Suppose that α_Q can be transformed into β_Q in $\mathcal{R}_{\mathcal{S}}(Q)$. Then to transform α into β in $\mathcal{R}_{\mathcal{S}}(G)$, follow the steps from α_Q to β_Q as follows. Whenever q_i is recoloured, then recolour every vertex in V_i the same colour. Therefore, it is enough to show how to transform α_Q into β_Q in $\mathcal{R}_{\mathcal{S}}(Q)$ by recolouring each vertex at most twice.

Initially, fix the colouring of Q to be α_Q. Let D be the directed graph on k vertices such that there is an arc $q_j q_i$ in D if and only if in the current colouring of Q, q_j is coloured $\beta(q_i)$. Since no two vertices of Q are coloured the same colour in any colouring, $d^-(q_i) \leq 1$ and $d^+(q_i) \leq 1$ for all $i = 1 \ldots k$. Therefore, D is the disjoint union of directed paths and directed cycles. Note that for any vertex q, if $d^-(q) = 0$ in D, then it can be immediately recoloured into $\beta_Q(q)$.

Recolour each directed path $v_1, v_2, \ldots, v_p$ as follows. Since $d^-(v_1) = 0$ recolour it $\beta_Q(v_1)$. Now we have that $d^-(v_2) = 0$ so recolour v_2 with $\beta_Q(v_2)$. Continue recolouring this way until all vertices in the path are coloured as in β_Q. Note that each vertex in the directed path was recoloured at most once.

Now assume D contains no directed paths and is the disjoint union of only directed cycles. Recolour each directed cycle $v_1, v_2, \ldots, v_p, v_1$ as follows. Since $|\mathcal{S}| > k$ while $|Q| = k$, v_p can be recoloured with some colour that does not appear in the current colouring. Now $d^-(v_1) = 0$ so the directed cycle becomes a directed path $v_1, v_2, \ldots, v_p$. Recolour each vertex as described in the case of a directed path.

Note that each vertex in a directed path was recoloured at most once. Also only one vertex in each directed cycle was recoloured at most twice, and each other vertex in the cycle was recoloured at most once. $\square$

4.3 Recolouring to the canonical χ-colouring

In this section we prove that the $(k+1)$-recolouring diameter of a k-colourable OAT graph is quadratic in the number of vertices, which follows from the following lemma.

Lemma 2. *Let G be an OAT graph. Let $\mathcal{S}$ be a set of $k+1$ colours where $k \geq \chi(G)$ and let $\mathcal{C}$ be an ordered set of $\chi(G)$ colours such that $\mathcal{C} \subseteq \mathcal{S}$. Then any colouring in $\mathcal{R}_{\mathcal{S}}(G)$ can be transformed into the canonical χ-colouring of G with respect to $\mathcal{C}$ by recolouring each vertex at most $2n$ times.*

Proof. The proof is by induction on the number of vertices n of G. Let $\alpha : V(G) \to \mathcal{S}$ be a $(k+1)$-colouring of G. We show how to transform α into the canonical χ-colouring γ of G with respect to $\mathcal{C}$ by recolouring each vertex at most $2n$ times. Clearly this holds if G is a single vertex, so assume G was constructed with one of the four operations defining OAT graphs.

Case 1: Suppose G is constructed as the disjoint union of the graphs L and R. Note that L and R can be recoloured independently since there are no edges between L and R. Let α_L be the projection of α onto L and define $\mathcal{S}(\alpha_L) = \mathcal{S}$ to be its set of permissible colours. Similarly, let α_R be the projection of α onto R and define $\mathcal{S}(\alpha_R) = \mathcal{S}$ to be its set of permissible colours. By Observation 1, $\chi(G) = \max\{\chi(L), \chi(R)\}$, and it follows that α_L is an $\mathcal{S}$-colouring of L with $|\mathcal{S}| \geq \chi(L) + 1$ and α_R is an $\mathcal{S}$-colouring of R with $|\mathcal{S}| \geq \chi(R) + 1$. Recall that for a subgraph H of G, n_H denotes the number of vertices of H. By the induction hypothesis, we can transform α_L within $\mathcal{R}_{\mathcal{S}}(L)$ into the canonical χ-colouring of L with respect to the first $\chi(L)$ colours of $\mathcal{C}$ by recolouring each vertex of L at most $2n_L$ times. Similarly, by the induction hypothesis, we can transform α_R within $\mathcal{R}_{\mathcal{S}}(R)$ into the canonical χ-colouring of R with respect to the first $\chi(R)$ colours of $\mathcal{C}$ by recolouring each vertex of R at most $2n_R$ times. Note that both of these reconfiguration sequences appear in $\mathcal{R}_{\mathcal{S}}(G)$ since there are no edges between L and R. Taking these two reconfiguration sequences consecutively gives the reconfiguration sequence within $\mathcal{R}_{\mathcal{S}}(G)$ to transform α into the canonical χ-colouring of G with respect to $\mathcal{C}$. Each vertex of L has been recoloured at most $2n_L < 2n$ times and each vertex of R has been recoloured at most $2n_R < 2n$ times.

Case 2: Suppose G is constructed as the join of the graphs L and R. Let α_L and α_R denote the projections of α onto L and R, respectively. Note that $\mathcal{C}(\alpha_L)$ is disjoint from $\mathcal{C}(\alpha_R)$ since there are all possible edges between L and R. We consider two cases depending on the number of colours appearing in α_L and α_R.

First suppose $|\mathcal{C}(\alpha_L)| = \chi(L)$ and $|\mathcal{C}(\alpha_R)| = \chi(R)$. Then there exists some colour c that does not appear in α since, by Observation 1, $\chi(G) = \chi(L) + \chi(R)$, and $|\mathcal{S}| > \chi(G)$. Define $\mathcal{S}(\alpha_L) = \mathcal{C}(\alpha_L) \cup \{c\}$ as the set of permissible colours for α_L. Then α_L is an $\mathcal{S}(\alpha_L)$-colouring of L and $|\mathcal{S}(\alpha_L)| > \chi(L)$. By the induction hypothesis, α_L can be transformed within $\mathcal{R}_{\mathcal{S}(\alpha_L)}(L)$ into the canonical χ-colouring of L with respect to the first $\chi(L)$ colours of $\mathcal{C}(\alpha_L)$ by recolouring each vertex at most $2n_L$ times. Furthermore, by our choice of $\mathcal{S}(\alpha_L)$, none of the intermediate colourings of L uses a colour from $\mathcal{C}(\alpha_R)$ so the same reconfiguration sequence appears within $\mathcal{R}_{\mathcal{S}}(G)$.

Since the canonical χ-colouring of L uses $\chi(L)$ colours, some colour c' does not appear in the current colouring of G. Define $\mathcal{S}(\alpha_R) = \mathcal{C}(\alpha_R) \cup \{c'\}$ as the set of permissible colours for α_R. Then α_R is an $\mathcal{S}(\alpha_R)$-colouring of R and $|\mathcal{S}(\alpha_R)| > \chi(R)$. By the induction hypothesis, α_R can be transformed within $\mathcal{R}_{\mathcal{S}(\alpha_R)}(R)$ into the canonical χ-colouring of R with respect the first $\chi(R)$ colours of $\mathcal{C}(\alpha_R)$ by recolouring each vertex at most $2n_R$ times. The same reconfiguration sequence appears within $\mathcal{R}_{\mathcal{S}}(G)$ since $\mathcal{S}(\alpha_R)$ is disjoint from the colours that appear on the vertices of L.

Now suppose $|\mathcal{C}(\alpha_L)| > \chi(L)$ or $|\mathcal{C}(\alpha_R)| > \chi(R)$ (suppose the former). Define $\mathcal{S}(\alpha_L) = \mathcal{C}(\alpha_L)$ as the set of permissible colours for α_L. Then α_L is an $\mathcal{S}(\alpha_L)$-colouring and $|\mathcal{S}(\alpha_L)| > \chi(L)$. By the induction hypothesis, α_L can be transformed within $\mathcal{R}_{\mathcal{S}(\alpha_L)}(L)$ into the canonical χ-colouring of L with respect the first $\chi(L)$ colours of $\mathcal{C}(\alpha_L)$ by recolouring each vertex at most $2n_L$ times. The same reconfiguration sequence appears in $\mathcal{R}_{\mathcal{S}}(G)$ since $\mathcal{S}(\alpha_L)$ and $\mathcal{C}(\alpha_R)$ are disjoint. Now some colour c^* that appeared in α_L no longer appears in the current colouring of G. Define $\mathcal{S}(\alpha_R) = \mathcal{C}(\alpha_R) \cup \{c^*\}$ as the set of permissible colours. Then α_R is a $\mathcal{S}(\alpha_R)$-colouring and $|\mathcal{S}(\alpha_R)| > \chi(R)$. By the induction hypothesis, α_R can be transformed within $\mathcal{R}_{\mathcal{S}(\alpha_R)}(R)$ into the canonical χ-colouring of R with respect the first $\chi(R)$ colours of $\mathcal{C}(\alpha_R)$ by recolouring each vertex at most $2n_R$ times. A similar argument holds if instead $|\mathcal{C}(\alpha_R)| > \chi(R)$. The same reconfiguration sequence appears in $\mathcal{R}_{\mathcal{S}}(G)$ since $\mathcal{S}(\alpha_R)$ is disjoint from the colours appearing on L.

To complete this part of the proof, we now have a colouring α' of G such that α'_L is a canonical $\chi(L)$-colouring of L and α'_R is a canonical $\chi(R)$-colouring of R. Then α' and the canonical $\chi(G)$-colouring γ of G must partition the vertices of G into the same colour classes. Then by the Renaming Lemma (Lemma 1), we can transform α' into γ by recolouring each vertex at most twice. Therefore we can transform α into γ by recolouring each vertex of G at most $2\max\{n_L, n_R\} + 2 \leq 2n$ times.

Case 3: Suppose G is constructed by adding a vertex u comparable to a vertex v of the OAT graph $H = G \setminus \{u\}$. First recolour u the same colour as v. This is possible since u and v are non-adjacent and $N(u) \subseteq N(v)$. Let α_H be the projection of α onto H and define $\mathcal{S}(\alpha_H) = \mathcal{S}$ to be its set of permissible colours. By Observation 1, $\chi(H) = \chi(G)$ and so α_H is an $\mathcal{S}$-colouring with $|\mathcal{S}| > \chi(H)$. By the induction hypothesis, α_H can be transformed within $\mathcal{R}_{\mathcal{S}}(H)$ into the canonical $\chi(H)$-colouring with respect to $\mathcal{C}$ by recolouring each vertex of H at most $2n_H$ times. To extend this reconfiguration sequence to $\mathcal{R}_{\mathcal{S}}(G)$, whenever v is recoloured, recolour u the same colour. By definition, this colouring of G is the canonical χ-colouring of G with respect to $\mathcal{C}$. Each vertex of H was recoloured at most $2n_H < 2n$ times and u was recoloured at most $2n_H + 1 < 2n$ times.

Case 4: Suppose G is constructed by attaching a complete graph Q to some vertex v of an OAT graph H. Let α_H be the projection of α onto H and define $\mathcal{S}(\alpha_H) = \mathcal{S}$ to be its set of permissible colours. By Observation 1, $\chi(H) \leq \chi(G)$ and so α_H is an $\mathcal{S}$-colouring of H with $|\mathcal{S}| > \chi(H)$. By the induction hypothesis, α_H can be transformed within $\mathcal{R}_{\mathcal{S}}(H)$ into the canonical χ-colouring γ_H of H with respect to the first $\chi(H)$ colours of $\mathcal{C}$. To extend this reconfiguration sequence to $\mathcal{R}_{\mathcal{S}}(G)$, whenever v is recoloured to some colour c, we may need to first recolour at most one vertex q of Q that is coloured c. Since $\chi(G) = \max\{\chi(H), n_Q + 1\}$ and $|\mathcal{S}| \geq \chi(G) + 1 \geq n_Q + 2$, and each vertex of Q has degree n_Q, there exists some colour c' that does not appear on the neighbourhood of q and is not the colour c. Recolour q with the colour c' and then continue by recolouring v colour c. Now H is coloured with the canonical $\chi(H)$-colouring γ_H.

Let $c^* = \gamma_H(v)$ and let α'_Q be the current colouring of Q and define $\mathcal{S}(\alpha'_Q) = \mathcal{S} \setminus \{c^*\}$ to be its set of permissible colours. Recall that the vertices of Q are ordered $\{q_1, q_2, \ldots\}$. The canonical χ-colouring of Q with respect to $\mathcal{C}$ is the colouring γ_Q such that q_i is coloured the ith colour of $\mathcal{C} \setminus \{c^*\}$. Since $|\mathcal{S}| \geq n_Q + 2$, then $|\mathcal{S} \setminus \{c^*\}| \geq n_Q + 1$. By the Renaming Lemma (Lemma 1), α'_Q can be transformed within $\mathcal{R}_{\mathcal{S}(\alpha'_Q)}(Q)$ into γ_Q by recolouring each vertex of Q at most twice. Since each vertex of Q is only adjacent to v in H and c^* was never used in this recolouring of Q, this reconfiguration sequence can extend to $\mathcal{R}_{\mathcal{S}}(G)$. Now by definition, the current colouring of G is the canonical χ-colouring of G with respect to $\mathcal{C}$. Each vertex of H was recoloured at most $2n_H$ times and each vertex of Q was recoloured at most $2n_H + 2 \leq 2n$ times.

□

We are now ready to prove Theorem 1.

Proof of Theorem 1. Fix $\mathcal{S} = \{1, 2, \ldots, k+1\}$ to be the set of permissible colours used in the colourings of $\mathcal{R}_{k+1}(G)$ and let $\mathcal{C}$ be an ordered set of $\chi(G)$ colours such that $\mathcal{C} \subseteq \mathcal{S}$. Let $\alpha, \beta : V(G) \to \mathcal{S}$ be two $(k+1)$-colourings of G. Then by Lemma 2, we can transform both α and β into the canonical χ-colouring γ of G with respect to $\mathcal{C}$ in $\mathcal{R}_{\mathcal{S}}(G)$ by recolouring each vertex at most $2n$ times. Then to transform α to β, follow the sequence from α to γ and then follow the sequence from β to γ in reverse. Therefore $\mathcal{R}_{k+1}(G)$ is connected with diameter at most $4n^2$. □

Chapter 5

Recognizing OAT Graphs

This chapter is dedicated to the problem of recognizing OAT graphs. In Section 5.1, we show that OAT graphs can be recognized by greedily deconstructing them and in Section 5.2, we give an algorithm that recognizes OAT graphs in $O(n^3)$ time.

We first survey recognition algorithms for various graph classes discussed in this thesis. The following graph classes have linear time recognition algorithms: chordal graphs [55], co-chordal graphs [40], P_4-free graphs [23], P_4-sparse graphs [43], and distance-hereditary graphs [38]. Chordal bipartite graphs can be recognized in $O(\min\{n^2, (n+m)\log n\})$ time [48, 52, 58], weakly chordal graphs can be recognized in $O(mn^2)$ time [60], and perfect graphs can be recognized in $O(n^9)$ time [19].

5.1 Deconstructing OAT graphs

In this section we prove several useful lemmas that together imply that OAT graphs can be recognized by greedily deconstructing them.

Lemma 3. *A graph G is an OAT graph if and only if every connected component of G is an OAT graph.*

Proof. Suppose every component of G is an OAT graph. Then the disjoint union operation can be used repeatedly to construct G. For the other direction, suppose G is an OAT graph. The proof is by induction on the number of vertices of G. The claim clearly holds if G has only one vertex or if G is connected, so assume not. By definition, G was constructed using the operations described in Definition 1. Then G was not constructed from the join operation since G is disconnected. The other operations preserve the connected components of G. We will show that in some construction of G, the disjoint union operation can be swapped with these operations.

Case 1: Suppose G was constructed by the disjoint union operation of L and R. By definition, L and R are OAT graphs. By the induction hypothesis, every component of L is an OAT graph and every component of R is an OAT graph. Note that the components of G are just the components of L and R. Therefore, every component of G is an OAT graph as desired.

Case 2: Suppose G was constructed by adding a vertex u comparable to a vertex v of H. By definition H is an OAT graph. By the induction hypothesis, every component of H is an OAT graph. Clearly, v only has neighbours in one component H^* of H. Then $H^* \cup u$ is an OAT graph since it was constructed from an OAT graph by adding a comparable vertex and the appropriate edges. Since all other components of G are the components of H, every component of G is an OAT graph.

Case 3: Suppose G was constructed by attaching a clique Q to a vertex z of a graph H. By definition, H is an OAT graph. By the induction hypothesis, each component of H is an OAT graph. Suppose z is a vertex of a component H^* of H. Then $G[H^* \cup Q]$ is an OAT graph since it was constructed from an OAT graph by attaching a clique. Since all other components of G are the components of H, every component of G is an OAT graph.

□

Lemma 4. *Suppose the vertices of G can be partitioned into two sets L and R such that L is joined to R. Then G is an OAT graph if and only if L is an OAT graph and R is an OAT graph.*

Proof. Suppose L and R are both OAT graphs. Then G is an OAT graph since G can be constructed from the join of L and R. For the other direction, suppose that G is an OAT graph. The proof is by induction on the number of vertices of G. By definition, G was constructed by the operations described in Definition 1. Note that G was not constructed by the disjoint union operation since G is connected.

Case 1: Suppose G is constructed from the join of L^* and R^*. Assume that $L \neq L^*$ (and $R \neq R^*$) and $L \neq R^*$ (and $R \neq L^*$) since otherwise we are done. By definition L^* and R^* are OAT graphs. Then L can be partitioned into two sets $L_1 = L^* \cap L$ and $L_2 = R^* \cap L$.

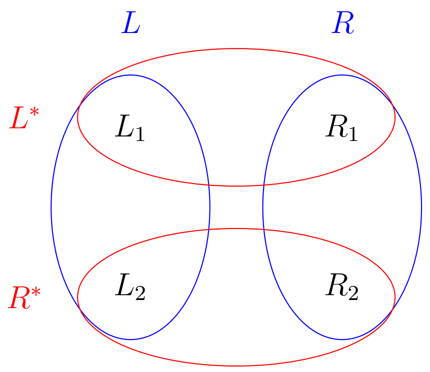


Figure 5.1: Two partitions of G into sets that are joined to each other.

Similarly, R can be partitioned into two sets $R_1 = L^* \cap R$ and $R_2 = R^* \cap R$ (see Figure 5.1). Note that at most one of L_1, L_2, R_1, and R_2 can be empty, otherwise $L = L^*$ and $R = R^*$ (or $L = R^*$ and $R = L^*$). Since L^* is joined to R^*, L_1 is joined to L_2 and R_1 is joined to R_2. Since $L^* = L_1 \cup R_1$ is an OAT graph and L_1 is joined to R_1, by the induction hypothesis, L_1 and R_1 are OAT graphs (if one is empty, then the other is an OAT graph since L^* is an OAT graph). Similarly, since $R^* = L_2 \cup R_2$ is an OAT graph and L_2 is joined to R_2, by the induction hypothesis, L_2 and R_2 are OAT graphs (if one is empty, then the other is an OAT graph since R^* is an OAT graph). If one of L_1 or L_2 is empty, then L is an OAT graph since either $L = L_1$ or $L = L_2$. If both L_1 and L_2 are not empty, then L is an OAT graph since L is the join of L_1 and L_2. Similarly, if one of R_1 or R_2 is empty, then R is an OAT graph since either $R = R_1$ or $R = R_2$. If both R_1 and R_2 are non-empty, then R is an OAT graph since R is the join of R_1 and R_2.

Case 2: Suppose G was constructed by adding a vertex u comparable to a vertex v of H. Then u and v must both be in L or both be in R since u and v are non-adjacent. Without loss of generality, suppose $u, v \in L$. By definition, H is an OAT graph. Note that H is the join of $L \setminus u$ and R. Then by the induction hypothesis $L \setminus u$ and R are OAT graphs. Then L is an OAT graph since it can be constructed from $L \setminus u$ by adding the vertex u comparable to v and the appropriate edges.

Case 3: Suppose G was constructed by attaching a clique Q to a vertex z of an OAT graph H. Since there are no edges between Q and $H \setminus z$ in G, either all of the vertices of $G \setminus z$ are in L or all of the vertices of $G \setminus z$ are in R. Without loss of generality, suppose the vertices of $G \setminus z$ are in L. Then since $R \neq \emptyset$, z is the only vertex in R. Then clearly R is an OAT graph. Note that L is composed of the graphs $H \setminus z$ and Q. By definition, H is an OAT graph. Note that since $z \in R$, H is the join of $H \setminus z$ and z. Then by the induction hypothesis, $H \setminus z$ is an OAT graph. Then L is an OAT graph since L can be constructed by the disjoint union of $H \setminus z$ and Q.

□

Lemma 5. *If G has a vertex u comparable to some vertex v of G, then G is an OAT graph if and only if $G \setminus u$ is an OAT graph.*

Proof. Suppose $G \setminus u$ is an OAT graph. Then G is an OAT graph since G can be constructed from $G \setminus u$ by adding back the comparable vertex u and all appropriate edges. For the other direction, suppose G is an OAT graph. The proof is by induction on the number of vertices of G. By definition, G was constructed by using the operations described in Definition 1.

Case 1: Suppose G was constructed from the disjoint union of the OAT graphs L and R. Without loss of generality, suppose $u \in L$. Assume that L has another vertex, since if not $G \setminus u$ is OAT by definition. If u is an isolated vertex, i.e. $N_G(u) = \emptyset$, then u is comparable to every other vertex of L. Otherwise, both u and v must be in L since there is a path connecting them through $N(u) \cap N(v)$. In either case, u is comparable to some vertex of L. By the induction hypothesis, $L \setminus u$ is an OAT graph. Then $G \setminus u$ is an OAT graph since it can be constructed from the disjoint union of $L \setminus u$ and R.

Case 2: Suppose G was constructed from the join of the OAT graphs L and R. Then both u and v are vertices of L or both u and v are vertices of R, since they are non-adjacent. Without loss of generality, suppose u and v are vertices of L. Since u and v are comparable in G, and removing the same set of vertices from $N(u)$ and $N(v)$ will not change that $N(u) \subseteq N(v)$, vertex u is comparable to v in L. By the induction hypothesis, $L \setminus u$ is an OAT graph. Then $G \setminus u$ is an OAT graph since it can be constructed from the join of $L \setminus u$ and R.

Case 3: Suppose G was constructed by adding a vertex x comparable to another vertex y of $G \setminus x$ (see Figure 5.2). Assume $u \neq x$ since otherwise $G \setminus u$ is an OAT graph by definition. Then since $G \setminus x$ is an OAT graph and since u is still comparable to v in $G \setminus x$, by the induction hypothesis $G \setminus \{x, u\}$ is an OAT graph. Next we show that x is comparable to some vertex of $G \setminus \{x, u\}$. If $u \neq y$, then y is a vertex of $G \setminus \{x, u\}$ and therefore x is comparable to some vertex of $G \setminus \{x, u\}$. If $u = y$, then v must be a vertex of $G \setminus \{x, u\}$ and $N(x) \subseteq N(y) = N(u) \subseteq N(v)$. Furthermore, x is non-adjacent to v in G since u is non-adjacent to v and $N(x) \subseteq N(u)$. In any case, x is comparable to some vertex of $G \setminus \{x, u\}$ and we can construct $G \setminus u$ from $G \setminus \{x, u\}$ by adding back x and the appropriate edges. Therefore, $G \setminus u$ is an OAT graph.

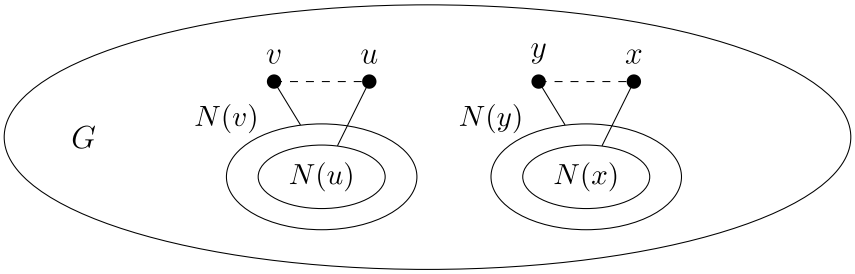


Figure 5.2: An OAT graph with two distinct comparable vertices.

Case 4: Suppose G was constructed by attaching a complete graph Q to a vertex z of an OAT graph H. If Q only contains the vertex u, then $H = G \setminus u$ is an OAT graph. Assume that there exists some vertex $x \neq u$ in Q. If $u \in Q \cup \{z\}$ then u is adjacent to x, and so v must be adjacent to x as well. Then $v \in Q \cup \{z\}$, contradicting that u and v are non-adjacent, and so it must be that $u \in H \setminus z$. If u is comparable to some vertex in H, then by the induction hypothesis $H \setminus u$ is an OAT graph, and we can construct $G \setminus u$ from $H \setminus u$ by attaching Q at z, proving that $G \setminus u$ is an OAT graph. So assume u is not comparable to any vertex in H, which implies that $v \in Q$. Then $N(u) = \{z\}$ since $u \in H \setminus z$ has no neighbours in Q while $N(v) \subseteq Q \cup \{z\}$. We also know that z has no other neighbours in H, otherwise u would be comparable to that neighbour. If $H = \{u, z\}$ then $G \setminus u$ is a clique and therefore an OAT graph. If H contains vertices other than u and z, then none of them are adjacent to v, z, or Q. Then H is the disjoint union of $H \setminus \{u, z\}$ and $\{u, z\}$. By Lemma 3, $H \setminus \{u, z\}$ is an OAT graph. Therefore $H \setminus u$ is an OAT graph since it can be constructed from $H \setminus \{u, z\}$ by using the disjoint union operation with z. Then $G \setminus u$ is an OAT graph since it can be constructed from $H \setminus u$ by attaching the clique Q at z.

□

Lemma 6. *Suppose G has a cut vertex z such that $G \setminus z$ has a component Q that is a complete graph and such that z is joined to Q. Then G is an OAT graph if and only if $G \setminus Q$ is an OAT graph.*

Proof. Let $H = G \setminus Q$ and suppose H is an OAT graph. Then G is an OAT graph since G can be constructed from H by attaching a clique Q to the vertex z. For the other direction, suppose G is an OAT graph. The proof is by induction on the number of vertices of G. By definition, G can be constructed by using the operations described in Definition 1.

Case 1: Suppose G was constructed from the disjoint union of the OAT graphs L and R. Without loss of generality, suppose $z \in L$. Then every vertex of Q is in L since z is adjacent to every vertex of Q. By the induction hypothesis, $L \setminus Q$ is an OAT graph. Then $G \setminus Q$ is an OAT graph since G can be constructed from the disjoint union of $L \setminus Q$ and R.

Case 2: Suppose G was constructed from the join of the OAT graphs L and R. Then all of the vertices of Q and H are in R or in L since there are no edges between Q and H. Without loss of generality, suppose all of the vertices of Q and H are in L. Then since R is not empty, it must be the case that $z \in R$. Since Q is a connected component of L, by Lemma 3, $L \setminus Q$ is an OAT graph. Then G is an OAT graph since G can be constructed by the join of $L \setminus Q$ and R.

Case 3: Suppose G was constructed by adding a vertex x comparable to another vertex y of an OAT graph H'. Note that $z \neq x$ since z is adjacent to every vertex of Q (so y could not be in Q) and no other vertex of $G \setminus Q$ is adjacent to a vertex of Q (so y could not be in $H \setminus z$ since x would have neighbours in Q). If $x \in Q$ then x is the only vertex of Q since x is adjacent to every other vertex of Q and z and no other vertex of $G \setminus Q$ is adjacent to a vertex in Q. Then $H = G \setminus Q$ and therefore $G \setminus Q$ is an OAT graph. Now assume $x \in G \setminus Q$ and $x \neq z$. Since H' is an OAT graph, and can be constructed from $H' \setminus Q$ by attaching a clique at z, by the induction hypothesis $H' \setminus Q$ is an OAT graph. If $y \in G \setminus Q$, then $G \setminus Q$ is an OAT graph since $G \setminus Q$ can be constructed from $H' \setminus Q$ by adding back the vertex x comparable to y. If $y \in Q$, then either x is an isolated vertex or x is only adjacent to z in G. If x is an isolated vertex, then $G \setminus Q$ is an OAT graph since it can be constructed from the disjoint union of x and $H' \setminus Q$. If x is only adjacent to z in G, then $G \setminus Q$ is an OAT graph since it can be constructed by attaching the clique x to the vertex z of $H' \setminus Q$.

Case 4: Suppose G was constructed by attaching a complete graph Q' to a vertex z' of an OAT graph H'. Assume that $Q \neq Q'$ (and therefore disjoint) since if not, we are done. Then by the induction hypothesis, $H' \setminus Q$ is an OAT graph. Then $G \setminus Q$ is an OAT graph since it can be constructed from $H \setminus Q$ by attaching the clique Q' at the vertex z'.

□

5.2 Recognition algorithm

In this section we give an algorithm called RECOGNIZE_OAT that recognizes OAT graphs in $O(n^3)$ time. The algorithm takes a graph G as input and either outputs a build-tree, certifying that G is an OAT graph or outputs the answer no. Recall that a build-tree is a tree whose root node is G, whose leaf nodes are the single vertices of G, and whose internal nodes represent the operations used to build G. As discussed previously, the build-tree of G is not unique, and the algorithm only returns one of possibly many build-trees.

Let G be the input graph. The idea of the algorithm is to check if any of the four defining operations can be used to construct G from smaller graphs. If so, then we recursively test those smaller graphs. Lemmas 3, 4, 5, and 6 justify the correctness of this approach. In particular, it does not matter in which order we check the four operations. Here are further details that (arbitrarily) use the order of operations from Definition 1. See Algorithm 1 for the pseudocode.

The algorithm first checks if G is connected. If not, then recursively check each connected component of G. If each component is an OAT graph, then G is an OAT graph by Lemma 3 and can be constructed by the disjoint union of its components. The build-tree of G is updated accordingly.

If G is connected, then the algorithm checks if the complement of G is connected. If not, then compute the connected components of co-G. The components of co-G correspond to a partition of the vertices of G into sets that have all possible edges between them. For each such component C of co-G, check whether co-C is an OAT graph. If each co-C is an OAT graph, then by Lemma 4 G is an OAT graph and can be constructed from the join operation of each co-C. The build-tree is then updated accordingly.

If both G and co-G are connected, the algorithm examines if G contains a vertex u comparable to a vertex v. If so, then recursively check if $G \setminus u$ is an OAT graph. If so, then G is an OAT graph by Lemma 5 and can be constructed from $G \setminus u$ by adding the comparable vertex u. The build-tree of G is updated accordingly.

Finally, the algorithm checks if G has any cut vertices z such that $G \setminus z$ has a component Q that is a complete graph and such that z is adjacent to all vertices in Q. If so, the algorithm recursively checks if $G \setminus Q$ is an OAT graph. If so, then G is an OAT graph by Lemma 6 and can be constructed by attaching Q to z in $G \setminus Q$. The build-tree of G is updated accordingly.

If none of the above cases apply then G is not an OAT graph as justified by Lemma 3, 4, 5, and 6. Next we give implementation details and analyze the run-time of the algorithm RECOGNIZE_OAT(G).

Let G be the input graph with n vertices and m edges. In each recursive step of the algorithm, there are four conditions that may need to be checked. We can only bound

Algorithm 1: The recognition algorithm RECOGNIZE_OAT(G).

```
Input: A graph $G$
Output: A build-tree that constructs $G$, or “false” ($G$ is not an OAT graph)

if $G$ is a single vertex then
    return a leaf node that stores the single vertex.
end

else if $G$ is disconnected then
    for each component $H$ of $G$ do
        if RECOGNIZE_OAT($H$) = false then  return false ;
    end
    $G$ is the union of its components. Update the build-tree accordingly and return
      it ;
end

else if co-$G$ is disconnected then
    for each component $C$ of co-$G$ do
        if RECOGNIZE_OAT(co-$C$) = false then  return false ;
    end
    $G$ is the join of the complements of the components of co-$G$. Update the
      build-tree accordingly and return it ;
end

else if $G$ has a vertex $u$ comparable to another vertex $v$ then
    if RECOGNIZE_OAT($G \setminus u$) = false then  return false ;
    else  $G$ is obtained from $G \setminus u$ by adding a comparable vertex $u$. Update the
      build-tree accordingly and return it ;
end

else if $G$ has a cut-vertex $z$ such that $G \setminus z$ has a component $Q$ that is a complete
  graph and such that $z$ is joined to $Q$ then
    if RECOGNIZE_OAT($G \setminus Q$) = false then  return false ;
    else $G$ is obtained from attaching the clique $Q$ to $z$. Update the build-tree
      accordingly and return it;
end
else
    return false
end
```

the run-time of checking some of these conditions by $O(n^2)$, so we will not be concerned with finding better bounds for checking other conditions. We assume that G is given by adjacency lists and compute the adjacency matrix of G in $O(n^2)$ time. Let $A(G)$ (or simply A) be the adjacency matrix of G where the rows and columns of A are indexed by the vertices of G. For $u, v \in V(G)$, the entry of A at row u and column v, denoted by $A[u, v]$, is 1 if u and v are adjacent and 0 otherwise.

Step 1: First determine the connected components of G. This can be done using depth-first search in $O(n + m)$ time.

Step 2: Next compute the adjacency matrix of co-G in $O(n^2)$ time by replacing $A[u, v]$ by $1 - A[u, v]$ for $u \neq v$. Then compute the connected components of co-G in $O(n^2)$ time using depth-first search.

Step 3: Next, search for the biconnected components and the cut vertices of G. This can be done in $O(n + m)$ time using depth-first search if G is stored with adjacency lists [61]. For each of the biconnected components Q of G, check if the vertices form a clique. This can be done in $O(n_Q^2)$ time (where $n_Q = |V(Q)|$) by checking whether Q has $\binom{n_Q}{2}$ edges. Therefore, this entire step can be done in $O(n^2)$ time.

Step 4: Finally, search for a vertex u comparable to another vertex v of G. A brute force approach for finding such a pair would take $O(n^3)$ time at each recursive step. Instead, we use the square of the adjacency matrix $A^2(G)$ and maintain it recursively (details below) to find a pair of comparable vertices at each step. It is well known that $A^r[x, y]$ gives the number of paths of length at most r from x to y in G. Then for non-adjacent and distinct vertices $x, y \in V(G)$, $N(x) \subseteq N(y)$ if and only if $A^2[x, y] = d(x)$ (the degree of x) [59]. Therefore, given A^2 we can test the existence of u in $O(n^2)$ time by scanning the entries of A^2.

We note that A^2 can be computed in $O(n^\omega)$ time where the current best known value of ω is about 2.376 [21]. However, to achieve the $O(n^3)$ running time of our algorithm, we only require that A^2 be computed in $O(n^3)$ time since, as we next show, A^2 can be updated in $O(n^2)$ time in each recursive step of RECOGNIZE_OAT(G).

Lemma 7. *Let G be an OAT graph and let $A^2(G)$ be given. Then for each subgraph H of G considered in the recursive steps of* RECOGNIZE_OAT(G)*, $A^2(H)$ can be computed in $O(n^2)$ time.*

Proof. We show how to update $A^2(G)$ in the recursive steps for each of the four operations.

Case 1: Suppose G is the disjoint union of the graphs L and R. Then $A^2(L)$ is simply the submatrix of $A^2(G)$ with rows and columns corresponding to the vertices of L, and similarly for $A^2(R)$.

Case 2: Suppose G was constructed from the join of the graphs L and R. Then $A^2(L)$ is the matrix obtained from $A^2(G)$ by deleting the rows and columns corresponding to the vertices of R and subtracting $|V(R)|$ from every entry. This holds because for any two vertices $l_1, l_2 \in L$, and any vertex $r \in R$, we had the path l_1, r, l_2 in G and this path does not exist in L. Similarly, $A^2(R)$ is the matrix obtained from $A^2(G)$ by deleting the rows and columns corresponding to the vertices of L and subtracting $|V(L)|$ from every entry.

Case 3: Suppose G was constructed by adding a vertex u comparable to a vertex of the OAT graph H. Then $A^2(H)$ can be obtained from $A^2(G)$ by deleting row u and column u and for each pair of vertices $x, y \in N(u)$ (not necessarily distinct), subtracting 1 from $A^2[x, y]$. This holds because G contained the path x, u, y which no longer exists in H.

Case 4: Suppose G was constructed from attaching the complete graph Q to the vertex v of the OAT graph H. Then $A^2(H)$ can be obtained from $A^2(G)$ by deleting the rows and columns corresponding to the vertices of Q and by subtracting $|Q|$ from the entry $A^2[v, v]$.

Clearly, for each of these cases, the desired matrix can be obtained from $A^2(G)$ in $O(n^2)$ time. □

Thus, our algorithm initially computes $A^2(G)$ which takes $O(n^3)$ time. Then at each step, we can find which operation (if any) could be used to construct G in $O(n^2)$ time. In the same time, we can also compute A^2 for the applicable subgraphs and then recurse. The algorithm may be required to recurse $\Theta(n)$ times and therefore the total run-time of the algorithm is $O(n^3)$. We note that changing the order in which the algorithm checks for each of the four conditions may improve the run-time. This proves Theorem 2.

Chapter 6

Conclusion

In this thesis, we introduce a class of graphs called OAT graphs defined by four simple operations. This class of graphs includes chordal bipartite graphs, compact graphs, P_4-sparse graphs, and some graphs which are not perfect but have equal chromatic number and clique number. We give an algorithm to recognize OAT graphs in $O(n^3)$ time and if the input graph G is an OAT graph, the algorithm will return a build-tree of G. We showed that for any k-colourable OAT graph G, the reconfiguration graph $\mathcal{R}_{k+1}(G)$ is connected with diameter $O(n^2)$. The proof of this can be converted into an algorithm that exhibits a recolouring sequence between any two $(k+1)$-colourings of G in polynomial time, assuming we have a build-tree of G.

We close with a few open problems. It was mistakenly reported by Bonamy and Bousquet [4] that there exists k-colourable P_5-free graphs G such that $\mathcal{R}_{k+1}(G)$ has an isolated vertex. We showed that the examples given by Bonamy and Bousquet in this paper are in fact not P_5-free (see Section 3.5.3). This motivates the following question.

Problem 1. *Given a k-colourable P_5-free graph G, is G $(k+1)$-mixing?*

We note that every 2-colourable P_5-free graph is chordal bipartite, so this statement holds for $k = 2$. It might be that the answer is yes for other small values of k. This leads to the following question.

Problem 2. *For which $k \geq 3$ is a k-colourable P_5-free graph $(k+1)$-mixing?*

We now examine questions related to OAT graphs and weakly chordal graphs. It would be interesting to consider if there are other simple operations that could generalize the class of OAT graphs and maintain the property that $\mathcal{R}_{k+1}(G)$ is connected.

Problem 3. *Is it possible to include the operations of adding a true twin and adding a simplicial vertex to the operations defining OAT graphs and still have* $\mathcal{R}_{k+1}(G)$ *connected and of diameter* $O(n^2)$*?*

The class of graphs built from these six operations would then include distance-hereditary graphs and chordal graphs. Or perhaps there are other simple operations that would allow for the inclusion of these graph classes?

It is known that not all k-colourable weakly chordal graphs are $(k+1)$-mixing (see Section 3.5.8). Complexity questions appear to be wide open, so we ask the following.

Problem 4. *Given a* k*-colourable weakly chordal graph* G*, what is the complexity of determining whether* G *is* $(k+1)$*-mixing?*

References

[1] S. Akers, B. Krishnamurthy. A group-theoretic model for symmetric interconnection networks. *IEEE Transactions on Computers*, 38:555–566, 1989.

[2] K. Appel, W. Haken. The solution of the four color map problem. *Scientific American*, 237:108–121, 1977.

[3] H. J. Bandelt, H. M. Mulder. Distance-hereditary graphs. *Journal of Combinatorial Theory, Series B*, 41:182–208, 1986.

[4] M. Bonamy, N. Bousquet. Recoloring graphs via tree decompositions. *European Journal of Combinatorics*, 69:200–213, 2018.

[5] M. Bonamy, N. Bousquet. Recoloring graphs via tree decompositions. arXiv:1403.6386 [cs.DM], 2014.

[6] M. Bonamy, N. Bousquet, C. Feghali, M. Johnson. On a conjecture of Mohar concerning Kempe equivalence of regular graphs. *Journal of Combinatorial Theory, Series B*, 135:179–199, 2019.

[7] M. Bonamy, N. Bousquet, G. Perarnau. Frozen colourings of bounded degree graphs. *Electronic Notes in Discrete Mathematics*, 68:167–172, 2018.

[8] M. Bonamy, M. Johnson, I. Lignos, V. Patel, D. Paulusma. Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs. *Journal of Combinatorial Optimization*, 27:132–143, 2014.

[9] P. Bonsma, L. Cereceda. Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theoretical Computer Science*, 410:5215–5226, 2009.

[10] P. Bonsma, A.E. Mouawad, N. Nishimura, V. Raman. The complexity of bounded length graph recoloring and CSP reconfiguration. In M. Cygan and P. Heggernes, editors, *Parametrized and Exact Computation (IPEC 2014)*, volume 8894 of *Lecture Notes in Computer Science*, pages 110–121. Springer, Berlin, 2014.

[11] N. Bousquet, V. Bartier. Linear transformations between colorings in chordal graphs. In M. A. Bender, O. Svensson, and G. Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPIcs*, pages 24:1–24:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.

[12] N. Bousquet, M. Heinrich. A polynomial version of Cereceda's conjecture. arXiv:1903.05619 [cs.DM], 2019.

[13] N. Bousquet, G. Perarnau. Fast recoloring of sparse graphs. *European Journal of Combinatorics*, 52:1-11, 2016.

[14] L. Cereceda. Mixing graph colourings. PhD thesis, London School of Economics and Political Science, 2007.

[15] L. Cereceda, J. van den Heuvel, M. Johnson. Connectedness of the graph of vertex-colourings. *Discrete Mathematics*, 308:913–919, 2008.

[16] L. Cereceda, J. van den Heuvel, M. Johnson. Mixing 3-colourings in bipartite graphs. *European Journal of Combinatorics*, 30:1593–1606, 2009.

[17] L. Cereceda, J. van den Heuvel, M. Johnson. Finding paths between 3-colourings. *Journal of Graph Theory*, 67:69-82, 2011.

[18] C.A. Christen, S.M. Selkow. Some perfect coloring properties of graphs. *Journal of Combinatorial Theory, Series B*, 27:49–57, 1979.

[19] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, K. Vušković. Recognizing Berge graphs. *Combinatorica*, 25:143-186, 2005.

[20] M. Chudnovsky, N. Robertson, P. Seymour, R. Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164:51–229, 2006.

[21] D. Coppersmith, S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9:251–280, 1990.

[22] D.G. Corneil, H. Lerchs, L. Stewart Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3:163–174, 1981.

[23] D.G. Corneil, Y. Perl, L.K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14:926–934, 1985.

[24] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics.* Springer–Verlag, Heidelberg, 2016.

[25] R. G. Downey, M. R. Fellow. *Parameterized Complexity.* Springer, New York, 1997.

[26] Z. Dvořák, C. Feghali. An update on reconfiguring 10-colorings of planar graphs. arXiv:2002.05383 [math.CO], 2020.

[27] M. Dyer, A.D. Flaxman, A.M. Frieze, E. Vigoda. Randomly coloring sparse random graphs with fewer colors than the maximum degree. *Random Structures and Algorithms*, 29:450–465, 2006.

[28] E. Eiben, C. Feghali. Toward Cereceda's conjecture for planar graphs. *Journal of Graph Theory*, 1-11, 2019.

[29] C. Feghali. Paths between colourings of graphs with bounded tree-width. *Information Processing Letters*, 144:37–38, 2019.

[30] C. Feghali. Paths between colourings of sparse graphs. *European Journal of Combinatorics*, 75:169–171, 2019.

[31] C. Feghali. Reconfiguring colourings of graphs with bounded maximum average degree. arXiv:1904.12698 [math.CO], 2019.

[32] C. Feghali. Reconfiguring 10-colourings of planar graphs. arXiv:1902.02278 [math.CO], 2019.

[33] C. Feghali, J. Fiala. Reconfiguration graph for vertex colourings of weakly chordal graphs. *Discrete Mathematics*, 343:111733, 2020.

[34] C. Feghali, M. Johnson, D. Paulusma. Kempe equivalence of colourings of cubic graphs. *European Journal of Combinatorics*, 59:1–10, 2017.

[35] C. Feghali, M. Johnson, D. Paulusma. A reconfigurations analogue of Brooks' theorem and its consequences. *Journal of Graph Theory*, 83:340–358, 2016.

[36] J. Flum, M. Grohe. *Parameterized Complexity Theory.* Springer, Berlin, 2006.

[37] D.R. Fulkerson, O.A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15:835–855, 1965.

[38] P.L. Hammer, F. Maffray. Completely separable graphs. *Discrete Applied Mathematics*, 27:85–99, 1990.

[39] C.T. Hoàng. Perfect graphs. PhD thesis, McGill University, 1989.

[40] C.T. Hoàng. Recognition and optimization algorithms for co-triangulated graphs. Technical report, Institute for Discrete Mathematics, University of Bonn, Germany, Report No. 90637, 1990.

[41] B. Jamison, S. Olariu. P_4-Reducibe graphs–class of uniquely tree-representable graphs. *Studies in Applied Mathematics*, 81:79–87, 1989.

[42] B. Jamison, S. Olariu. A tree representation for P_4-sparse graphs. *Discrete Applied Mathematics*, 35:115–129, 1992.

[43] B. Jamison, S. Olariu. Recognizing P_4-sparse graphs in linear time. *SIAM Journal on Computing*, 21:381–406, 1992.

[44] B. Jamison, S. Olariu. Linear time optimization algorithms for P_4-sparse graphs. *Discrete Applied Mathematics*, 61:155–175, 1995.

[45] M. Jerrum. A very simple algorithm for estimating the number of k-colorings of a low-degree graph. *Random Structures and Algorithms*, 7:157–165, 1995.

[46] M. Johnson, D. Kratsch, S. Kratsch, V. Patel, D. Paulusma. Finding shortest paths between graph colourings. *Algorithmica*, 75:295–321, 2016.

[47] M. Las Vergnas, H. Meyniel. Kempe classes and the Hadwiger conjecture. *Journal of Combinatorial Theory, Series B*, 31:95-104, 1981.

[48] A. Lubiw. Doubly lexical orderings of matrices. *SIAM Journal on Computing*, 16:854-–879, 1987.

[49] H. Meyniel. Les 5-colorations d'un graphe planaire forment une classe de commutation unique. *Journal of Combinatorial Theory, Series B*, 24:251–257, 1978.

[50] B. Mohar. Kempe equivalence of colorings. Graph Theory in Paris: Proceedings of a Conference in Memory of Claude Berge. Editors: A. Bondy, J. Fonlupt, J. -L. Fouquet, J. -C. Fournier, J. L. Ramírez Alfonsín. Birkhäuser Basel, Basel, 287–297, 2007.

[51] N. Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018.

[52] R. Paige, R.E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16:973–989, 1987.

[53] I. Pak. Reduced decompositions of permutations in terms of star transpositions, generalized Catalan numbers and k-ary trees. *Discrete Mathematics*, 204:329–335, 1999.

[54] F.J. Portier, T.P. Vaughan. Whitney numbers of the second kind for the star poset. *European Journal of Combinatorics*, 11:277–288, 1990.

[55] D.J. Rose, R.E. Tarjan, G.S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on computing*, 5:266–283. 1976.

[56] W.J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4:177-192.

[57] M. Sipser. *Introduction to the Theory of Computation.* PWS Publishing, 1997.

[58] J. Spinrad. Doubly lexical ordering of dense 0–1 matrices. *Information Processing Letters*, 45:229–235, 1993.

[59] J. Spinrad. *Efficient Graph Representations.* American Mathematical Society, 2003.

[60] J. Spinrad, R. Sritharan. Algorithms for weakly triangulated graphs. *Discrete Applied Mathematics*, 59:181–191, 1995.

[61] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1:146–160, 1972.