Graph Partitioning With Input Restrictions

Siani Smith

A thesis presented for the degree of
Doctor of Philosophy

Department Of Computer Science
Durham University
January 2022

Abstract

In this thesis we study the computational complexity of a number of graph partitioning problems under a variety of input restrictions. Predominantly, we research problems related to Colouring in the case where the input is limited to hereditary graph classes, graphs of bounded diameter or some combination of the two.

In Chapter 2 we demonstrate the dramatic effect that restricting our input to hereditary graph classes can have on the complexity of a decision problem. To do this, we show extreme jumps in the complexity of three problems related to graph colouring between the class of all graphs and every other hereditary graph class.

We then consider the problems Colouring and k-Colouring for $\mathcal{H}$-*free* graphs of bounded diameter in Chapter 3. A graph class is said to be H-free for some graph H if it contains no induced subgraph isomorphic to H. Similarly, G is said to be $\mathcal{H}$-free for some set of graphs $\mathcal{H}$, if it does not contain any graph in $\mathcal{H}$ as an induced subgraph. Here, the set $\mathcal{H}$ consists usually of a single cycle or tree but may also contain a number of cycles, for example we give results for graphs of bounded diameter and girth.

Chapter 4 is dedicated to three variants of the Colouring problem, Acyclic Colouring, Star Colouring, and Injective Colouring. We give complete or almost complete dichotomies for each of these decision problems restricted to H-free graphs.

In Chapter 5 we study these problems, along with three further variants of 3-Colouring, Independent Odd Cycle Transversal, Independent Feedback Vertex Set and Near-Bipartiteness, for H-free graphs of bounded diameter.

Finally, Chapter 6 deals with a different variety of problems. We study the problems Disjoint Paths and Disjoint Connected Subgraphs for H-free graphs.

Contents

List of Figures

List of Tables

Declaration of Authorship

No part of this thesis has been previously submitted for any degree at any institution. Most of the results presented in this thesis have appeared in the papers [75, 73, 74, 12, 13, 11, 19, 20, 58], all of which have been peer reviewed. Each chapter is based on results from one or more of these papers. The papers used in each chapter are detailed in the thesis overview section of the introduction. Although the results in this thesis are obtained through joint research, I have participated actively in discussions leading to these results and have made significant contributions.

Acknowledgements

Firstly, I thank my supervisors, Barny Martin and Daniël Paulusma, for their help and advice throughout my studies. Thankyou for introducing me to research but also for making my time in Durham even more enjoyable. In addition, I want to acknowledge the other co-authors who contributed to the work described in this thesis: Jan Bok, Christoph Brause, Petr Golovach, Nikola Jedlicková and Erik Jan van Leeuwen.

I'm also grateful to all the friends I've made in Durham, especially to my office mates in MCS2010 for bearing with me during the writing of this thesis! I'd especially like to thank Karl for putting up with me over the course of my PhD and always ensuring I have an abundant supply of coke zero.

Finally, I want to thank my family for their support throughout my education and to reassure them that it really is over this time.

Chapter 1

Introduction

A graph is a computational model encoding relationships between pairs of objects. The 'objects' in this model are known as vertices whilst the relationships between them are represented by edges.

In some applications these relationships indicate links between objects, for example we may construct a graph with a vertex set consisting of users in a social network such that two users are joined by an edge if they are directly linked within the network. Alternatively, an edge may represent a conflict between two objects. For instance, consider a graph modelling a set of tasks to be completed. Each task is represented by a vertex, with two vertices joined by an edge if their corresponding tasks cannot be completed simultaneously.

The problems considered in this thesis arise mainly in graphs modelling situations of the second kind. We consider vertex partitioning problems where the objective is to divide the vertices of a graph into disjoint sets in a way which satisfies some particular properties.

The most famous of these problems is graph colouring, where the objective is to partition the vertices of a graph into the fewest possible distinct classes, known as colour classes, such that there is no edge between any two vertices belonging to the same colour class. This problem arises in a number of practical and theoretical settings but is said to have been first introduced by Francis Guthrie in attempting to colour a map of the counties of England [95].

Map colouring provides some of the most famous problems in graph colouring, for example the Four Colour Theorem proved by Appel and Haken

[7] which states that any planar graph can be coloured using only four colours. Informally, this means that four colours are sufficient to colour the countries of any realistic map such that no nations sharing a border receive the same colour. Additionally, colouring and its variants are useful in a variety of other settings such as scheduling, register allocation and frequency assignment.

In this thesis we study graph colouring alongside related variants such as acyclic colouring, star colouring and injective colouring. These variants each ask for a colouring of a graph whilst placing certain additional requirements on the partition.

Our focus is on the computational complexity of these problems. Specifically, each of the problems we study is known to be computationally hard in the general case where any graph is allowed as an input. We consider the effect of placing certain restrictions on the set of allowable input graphs, determining conditions under which each problem either becomes efficiently solvable or remains computationally hard.

Before presenting our results, we introduce the necessary terminology and notation.

1.1 Basic Graph Terminology

A graph is an ordered pair (V, E) where V is a set whose elements are called the *vertices* of G and E is a set of unordered pairs uv of vertices in V known as *edges*. We say that an edge e is incident to a vertex v if v belongs to e. Given an edge uv, the vertices u and v are called the endpoints of e and are said to be adjacent. Note that some definitions of E allow self loops, where an edge consists of a single vertex whereas we consider only pairs of distinct vertices. Similarly, in some settings graphs are directed, that is its edges are ordered pairs, whilst the graphs considered here are undirected. One variation which is occasionally used here allows the same edge to appear multiple times in a graph. In other words E is a multi-set. For a graph $G = (V, E)$ we call V the vertex set of G and E the edge set. Additionally, we let $n = |V|$ be the number of vertices of G and $m = |E|$ the number of edges.

The *degree* of a vertex is the number of edges incident to it. We denote the minimum degree of a vertex in G by $\delta(G)$ and the maximum degree by $\Delta(G)$. A vertex of degree 1 is called a pendant vertex or a leaf. If every vertex of a graph G has degree p then G is said to be p-regular. The *open* *neighbourhood* $N(v)$ of a vertex is the set of vertices to which it is adjacent. The *closed neighbourhood* $N[v]$ is the open neighbourhood together with v itself. To *identify* two vertices u and v, we delete them both and add a new vertex z with neighbourhood $N(u) \cup N(v)$. If u and v are adjacent then this operation is known as edge contraction. To *subdivide* an edge uv, we delete the edge uv and add a new vertex z adjacent to both u and v.

A graph H is said to be a *subgraph* of another graph G if H can be obtained from G by deleting some combination of vertices and edges. A subgraph H of G is *induced* if it can be obtained by deleting only vertices. We write $H \subseteq G$ to denote that H is a subgraph of G and $H \subseteq_i G$ to denote that H is an induced subgraph of G. For $S \subset V(G)$ we let $G[S]$ denote the subgraph of G induced by the vertices in S and $G - S$ the subgraph of G induced by the vertices of $V(G) \setminus S$. Two graphs G and H are called *isomorphic* if there exists a bijection $f : V(G) \to V(H)$ such that uv is an edge of G if and only if $f(u)f(v)$ is an edge of H.

A *path* P_n is a graph with vertex set $V = \{x_1, \ldots, x_n\}$ and edge set $E = \{x_i x_{i+1} : 1 \leq i \leq n-1\}$. The length of a path is the number of its edges. A *cycle* is a graph with vertex set $V = \{x_1, \ldots, x_n\}$ and edge set $E = \{x_1 x_n\} \cup \{x_i x_{i+1} : 1 \leq i \leq n-1\}$. A graph G is a *forest* if it contains no cycles. It is *bipartite* if it contains no cycles of odd length. In other words, G is bipartite if it can be partitioned into two parts A and B such that $G[A]$ and $G[B]$ are independent. The *girth* of G is the length of its shortest cycle. The disjoint union of two vertex-disjoint graphs F and G is the graph $G + F = (V(F) \cup V(G), E(F) \cup E(G))$. The disjoint union of s copies of a graph G is denoted sG. A *linear forest* is the disjoint union of paths.

A graph G is *connected* if there exists a path between every pair of its vertices. For a disconnected graph G, the maximal connected subgraphs of G are called *connected components*. A connected forest is called a *tree*. The *distance* between two vertices is the minimum length of a path between them. The *diameter* of a graph G is the maximum, over all pairs uv, of the distance between u and v.

The *complement* $\bar{G} = (V, \bar{E})$ of a graph $G = (V, E)$ is a graph with the same vertex set V and with edge set $\bar{E}$ such that, for $u \neq v$, $uv \in \bar{E}$ if and only if $uv \notin E$. A vertex u *dominates* G if $uv \in E$ for every $v \in V(G) \setminus u$. A *matching* in a graph G is a set of edges such that each vertex of G belongs to at most one edge. A matching is *perfect* if every vertex belongs to exactly one edge.

The graph K_n, known as the *complete graph*, is a graph on n vertices whose edge set consists of all possible edges. A set of vertices in a graph G inducing a complete graph is called a *clique*. Conversely, the graph rP_1 is a graph on r vertices whose edge set is empty. A set of vertices inducing rP_1 is called an *independent set*. The Ramsey number $R(k,l)$ is the minimum integer n such that any graph on at least n vertices contains either an independent set of size k or a clique of size l.

The *line graph* L of a given graph G is a graph with one vertex for each edge of G such that two vertices are adjacent in L if and only if the corresponding edges in G share an endpoint. The complete bipartite graph, or *biclique*, denoted $K_{i,j}$, is a bipartite graph where one part A has size i, the other part B has size j and every vertex of A is adjacent to every vertex of B. In particular the graph $K_{1,3}$ is known as the *claw* whilst graphs of the form $K_{1,r}$ for some integer r are called *stars*. These graphs are examples of *polyads*, that is they are trees with exactly one vertex of degree at least 3. Another important family of polyads consists of subdivided stars $K^l_{1,r}$. These are graphs formed from the star $K_{1,r}$ by subdividing one edge l times. For example the graph $K^1_{1,3}$ is known as the *chair*. One well known fact which is used throughout this thesis is that line graphs do not contain claws as induced subgraphs. To see this, note that if three edges each share an endpoint with a fourth then at least two of them must also share an endpoint.

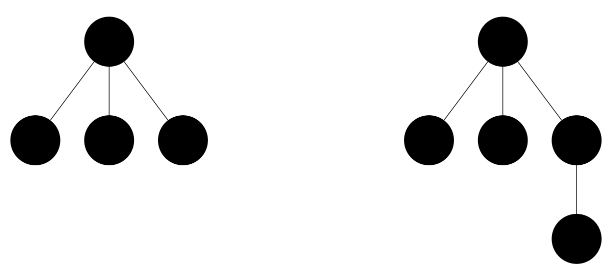

Figure 1.1: Left: The graph known as the claw. Right: The graph known as the chair.

1.2 Graph Partitioning Problems

In this section we define many of the decision problems studied in this thesis. A *(vertex) colouring* of a graph $G = (V, E)$ is a mapping $c : V \to \{1, 2, \ldots\}$

that assigns each vertex $u \in V$ a *colour* $c(u)$ in such a way that $c(u) \neq c(v)$ whenever $uv \in E$. The set of vertices for which $c(v) = i$ is known as the colour class i. If $1 \leq c(u) \leq k$, then c is said to be a *k-colouring* of G and G is said to be *k-colourable*. This leads to the following two NP-complete decision problems [39].

COLOURING	
Instance:	A graph G, integer k
Question:	Is G k-colourable?

If k is *fixed*, that is, k is not part of the input, we denote the problem by k-COLOURING.

k-COLOURING	
Instance:	A graph G
Question:	Is G k-colourable?

The smallest integer k for which a graph G is k-colourable is known as the chromatic number of G.

We now consider a well known generalisation of COLOURING called LIST COLOURING. A *list assignment* of a graph $G = (V, E)$ is a function L which assigns to each vertex a list of available colours $L(v)$. If $L(v) \subset \{1 \dots k\}$ for each $v \in V$ then L is a k-list assignment. The size of a list assignment is the maximum over all vertices of $|L(v)|$. The problem LIST COLOURING is to decide, given as input a graph G with a list assignment L, does there exist a colouring c of G which respects L?

LIST COLOURING	
Instance:	A graph G with a list assignment L
Question:	Is there a colouring c of G such that $c(v) \in L(v)$ for every $v \in V$?

For fixed k, we obtain the problem k-LIST COLOURING problem.

k-LIST COLOURING	
Instance:	Graph G with a list assignment L of size k
Question:	Is there a colouring c of G such that $c(v) \in L(v)$ for every $v \in V$?

If L assigns every vertex a list from the set $\{1, 2 \dots k\}$ we obtain the LIST-k COLOURING problem.

List-k-Colouring	
Instance:	Graph G with a list assignment L such that $L(v) \subseteq \{1, 2 \ldots k\}$
Question:	Is there a colouring c of G such that $c(v) \in L(v)$ for every $v \in V$?

Note that any instance of k-Colouring corresponds to an instance of List k-Colouring where each vertex is assigned the list $\{1, 2, \ldots k\}$. Similarly, every instance of List k-Colouring is an instance of k-List Colouring. Therefore, a polynomial-time result for k-List Colouring leads to polynomial results for the first two problems whilst an NP-completeness result for k-Colouring guarantees hardness for the second two.

Next we consider three variants of the colouring problem. A colouring c of a graph G is called *acyclic* if c assigns at least three different colours to every cycle in G. This leads to the following two decision problems, first studied in [45]. These problems are shown to be NP-complete [61] and [2].

Acyclic Colouring	
Instance:	A graph G, Integer k
Question:	Does there exist an acyclic k-colouring of G?

Acyclic k-colouring	
Instance:	A graph G
Question:	Does there exist an acyclic k-colouring of G?

A colouring c is called a *star colouring* if additionally every path of length 3 is assigned at least three different colours. This problem is first studied in [27]. Again, this gives rise to two decision problems.

Star Colouring	
Instance:	A graph G, an integer k
Question:	Does there exist a star k-colouring of G?

Star k-colouring	
Instance:	Graph G
Question:	Does there exist a star k-colouring of G?

Finally, c is called *injective* if every path of length 2 is assigned at least three different colours. Note that this problem is also known in the literature as distance 2 colouring or $L(1, 1)$-labelling. Once again we obtain two decision problems which are known to be NP-complete [77].

Injective Colouring	
Instance:	A graph G, an integer k
Question:	Does there exist an injective k-colouring of G?

Injective k-colouring	
Instance:	A graph G
Question:	Does there exist an injective k-colouring of G?

Another variant of the the colouring problem is the *distance constrained labelling framework*. See [22] for a survey. An $L(a_1 \dots a_p)$-k-labelling of a graph G is an assignment of the labels $\{1 \dots k\}$ to the vertices of G such that, for $1 \leq i \leq p$, whenever there is a path of length i between two vertices u and v, their labels differ by at least a_i. For example, as noted above, injective colouring is equivalent to $L(1, 1)$-labelling. Just as for our previous colouring variants, this leads to two decision problems.

$L(a_1 \dots a_p)$-labelling	
Instance:	A graph G, an integer k
Question:	Does G have an $L(a_1 \dots a_p)$-labelling with k labels?

$L(a_1 \dots a_p)$-k-labelling	
Instance:	A graph G
Question:	Does G have an $L(a_1 \dots a_p)$-labelling with k labels?

The final group of problems we consider can be seen as variants of the 3-colouring problem. A graph G is *near-bipartite* if it's vertex set can be partitioned into an independent set I and a forest F. Observe that I together with any 2-colouring of F gives a 3-colouring of G with the property that some pair of its colour classes induce a forest. The set I is known as an independent feedback vertex set. We now define two decision problems which are shown to be NP-complete in [18].

Near-Bipartiteness	
Instance:	A graph G
Question:	Is G near-bipartite?

Independent Feedback Vertex Set	
Instance:	A graph G, an integer k
Question:	Does G have an independent feedback vertex set of size at most k?

A subset S of V is an *independent odd cycle transversal* if S is an independent set and $G - S$ is bipartite. Note that a graph is 3-colourable if and only if it has an independent odd cycle transversal. The problem Independent Odd Cycle Transversal can then be viewed as the question of whether a graph G has a 3-colouring where one of the colour classes has at most some given size. See [76] for further information on this problem which is also known as Stable Bipartization.

Independent Odd Cycle Transversal	
Instance:	A graph G, an integer k
Question:	Does G have an independent odd cycle transversal of size at most k?

1.3 Special Graph Classes

The problems described above are known to be in NP-complete in the general case. The goal of this thesis is to investigate them in restricted settings, enhancing our understanding of the reasons for computational hardness by establishing possible boundaries between polynomial-time solvability and NP-completeness.

To do this, we restrict our inputs to graphs belonging to particular classes. Predominantly, we consider classes of H-free graphs. A graph G is said to be H-free if it contains no induced subgraph isomorphic to H. Similarly, G is $\mathcal{H}$-free for some set of graphs $\mathcal{H}$ if it contains none of the graphs in $\mathcal{H}$. For example the class of graphs of girth at least g is equivalently the class of $(C_3, \ldots, C_{g-1})$-free graphs. We may also consider the case where $\mathcal{H}$ is an infinite set. For example bipartite graphs are the class of graphs which are $\mathcal{H}$-free where $\mathcal{H}$ is the set of all odd cycles.

A graph class is called *hereditary* if and only if it is closed under vertex deletion. One reason for studying H-free graphs is the well known fact that a graph class is hereditary if and only if it can be defined as the set of $\mathcal{H}$-free graphs for some, possibly infinite, set $\mathcal{H}$. In this case the set $\mathcal{H}$ is known as the minimal set of forbidden induced subgraphs for the given graph class. Here we focus mainly on the cases $|\mathcal{H}| = 1$ and $|\mathcal{H}| = 2$ but also consider other classes such as graphs of high girth and bipartite graphs.

The main non-hereditary property considered in this thesis is diameter. We often consider bounding the diameter in addition to restricting our inputs to H-free graphs, either because the problem remains NP-complete for H-free graphs or because results for H-free graphs remain elusive.

1.4 Thesis Overview

In the remainder of this thesis we consider the colouring problem and its variants under various restrictions, considering H-free graphs, graphs of bounded diameter or combinations of the two.

First, in Chapter 2, we illustrate our reasoning for studying hereditary graph classes. We provide examples of extreme jumps in complexity when the input is restricted from the class of all graphs to any other hereditary graph class. The results in this chapter are taken from [75].

In Chapter 3 we consider the COLOURING and k-COLOURING problems for H-free graphs of bounded diameter. In particular, we present new polynomial-time and NP-completeness results for graphs of bounded diameter and girth as well as for polyad-free graphs of bounded diameter. We also consider LIST 3-COLOURING for $\mathcal{H}$-free graphs where $\mathcal{H}$ consists of either one or two short cycles. The results presented here are published in [73] and [74].

Chapter 4 is dedicated to studying acyclic, star and injective colouring for H-free graphs. In the case where k is fixed we provide a complete complexity dichotomy for each problem. When k is part of the input we leave finitely many open cases for each problem. The results in this chapter are taken from [11] which is a journal article based on two conference papers, [12] and [13].

In Chapter 5 we consider variants of the colouring problem for graphs of bounded diameter. Polynomial-time results are obtained for chair-free graphs for several variants of the 3-colouring problem. Meanwhile we provide NP-completeness results for $L(1,2)$-labelling graphs of diameter at most 2. The results of this chapter are taken from [19] and [20].

Chapter 6 deals with a different variety of problems, DISJOINT PATHS and DISJOINT CONNECTED SUBGRAPHS. The chapter begins by introducing the required definitions and terminology. We then completely classify the complexity of k-DISJOINT-CONNECTED SUBGRAPHS for H-free graphs. We also determine the complexity of DISJOINT PATHS and DISJOINT CONNECTED SUBGRAPHS for all but three open cases. The results in this chapter are published in [58].

Chapter 2

The Power of Input Restrictions

Here we consider a number of problems related to Colouring, exhibiting extreme jumps in complexity between the class of all graphs and any other hereditary graph class. In Section 2.1 we introduce the Colouring-or-Subgraph problem and show that it is NP-hard in general but constant-time solvable in any hereditary class not equal to the class of all graphs. In Section 2.2 we present the problem Colour-Path-or-Subgraph and show that its complexity jumps from PSPACE-completeness for the class of all graphs to constant-time solvability for any other hereditary class. Finally, in Section 2.3, we introduce the problem Succinct Colouring-or-Subgraph and show that, whilst it is NEXPTIME-complete in general, it is constant-time solvable for any other hereditary graph class.

Before doing so, we introduce some further notions in computational complexity. In Chapter 1 we introduced the complexity classes P and NP along with the concept of NP-completeness. Expanding upon this, a computational problem Π is NP-hard if there is a polynomial-time reduction from every problem in NP to Π.

The complexity class PSPACE is the class of all problems solvable using a polynomial amount of space. PSPACE-completeness is defined analogously to NP-completeness. Meanwhile, the class Σ_2^{P} consists of languages L such that there exists a polynomial-time predicate P and a polynomial q such that a string x belongs to L if and only if there exists a string y of length $q(|x|)$ such that for every string z of length $q(|x|)$, $P(x, y, z) = 1$.

The complexity class EXPTIME is the class of all problems solvable in exponential time, in other words in time $2^{n^{O(1)}}$. The class NEXPTIME is then defined analogously to the class NP with NEXPTIME-completeness defined analogously to NP-completeness.

2.1 NP-Hardness: Graph Colouring

We define the following decision problem and then present our first result, whose proof will serve as a basis for our other proofs.

Colouring-or-Subgraph	
Instance:	An n-vertex graph G
Question:	Is G $\lceil\sqrt{\log n}\rceil$-colourable or H-free for some graph H with $\lvert V(H)\rvert \leq \lceil\sqrt{\log n}\rceil$?

Theorem 2.1. *The* Colouring-or-Subgraph *problem is* NP*-hard, but constant-time solvable for every hereditary graph class not equal to the class of all graphs.*

Proof. To prove NP-hardness we reduce from 3-Colouring, which we recall is NP-complete [68]. Let G be an n-vertex graph. Set $p = \lceil\sqrt{\log 3n}\rceil$. We may assume without loss of generality that $p \geq 4$. Add a clique on $p-3$ vertices to G. Make the new vertices also adjacent to every vertex of G. We denote the new graph by G^*. Let $\{H_1, \ldots, H_r\}$ be the set of all graphs with exactly p vertices. We now define the graph G' as the disjoint union of G^* and the graphs $H_1, \ldots, H_r$; see also Figure 6.2. Note that the number of vertices of the graph $H_1 + \ldots + H_r$ is at most $p2^{\frac{p(p-1)}{2}} \leq \lceil\sqrt{\log 3n}\rceil \cdot \sqrt{3n} \leq n$ as $p \geq 4$. This implies that the number of vertices in G' is

$$\lvert V(G')\rvert = \lvert V(G)\rvert + p - 3 + \lvert V(H_1)\rvert + \ldots + \lvert V(H_r)\rvert \leq n + (p-3) + n < 3n.$$

In particular, the above shows that the number of vertices in G' is bounded by a polynomial in n. We now add $3n - \lvert V(G')\rvert$ isolated vertices to G' such that G' has exactly $3n$ vertices.

We claim that G is 3-colourable if and only if G' is a is a yes-instance of Colouring-or-Subgraph. First suppose that G is 3-colourable. We give each of the $p-3$ vertices of G^* that is not in G a unique colour from $\{4,\ldots,p\}$. As G is 3-colourable, we find that G^* is p-colourable. As G' is the disjoint union of G^* and the graphs $H_1,\ldots,H_r$ (and some isolated vertices, which we give colour 1), we must now consider the graphs $H_1,\ldots,H_r$. By construction, each H_i has p vertices, so we can give each vertex of each H_i a colour from $\{1,\ldots,p\}$ that is not used on any other vertex of H_i. Hence, we find that G' is p-colourable. As $p = \lceil\sqrt{\log 3n}\rceil = \lceil\sqrt{\log|V(G')|}\rceil$, this implies that G' is a yes-instance of Colouring-or-Subgraph.

Now suppose that G' is a yes-instance of Colouring-or-Subgraph. Recall that G' is the disjoint union of the graph G^*, the graphs $H_1,\ldots,H_r$ and some isolated vertices, and recall also that the graphs $H_1,\ldots,H_r$ are all the graphs on exactly p vertices. Hence, G' contains every graph on at most p vertices as an induced subgraph. In other words, G' is not H-free for some graph H with $|V(H)| \leq p$. As G' is a yes-instance of Colouring-or-Subgraph and $p = \lceil\sqrt{\log 3n}\rceil = \lceil\sqrt{\log|V(G')|}\rceil$, this means that G' must be p-colourable. As the $p-3$ vertices of $V(G^*)\setminus V(G)$ form a clique, we may assume without loss of generality that they are coloured $4,\ldots,p$, respectively. All these $p-3$ vertices are adjacent to every vertex of G in G^*. Consequently, every vertex of $V(G)$ must have received a colour from the set $\{1,2,3\}$. Hence, G is 3-colourable.

We now prove the second part of the theorem. Let $\mathcal{G}$ be a hereditary graph class that is not the class of all graphs. Then there exists at least one graph H such that every graph $G \in \mathcal{G}$ is H-free. Let $\ell = |V(H)|$. We claim that Colouring-or-Subgraph is constant-time solvable for $\mathcal{G}$. Let $G \in \mathcal{G}$ be an n-vertex graph. If $n \leq 2^{\ell^2}$, then G has constant size and the problem is constant-time solvable. If $n > 2^{\ell^2}$, then

$$|V(H)| = \ell < \sqrt{\log n} \leq \lceil\sqrt{\log n}\rceil.$$

Hence G is a yes-instance of Colouring-or-Subgraph, as G is H-free and H has at most $\lceil\sqrt{\log n}\rceil$ vertices. □

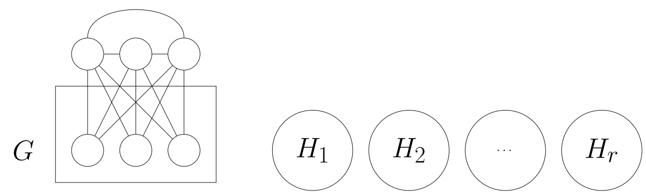

Figure 2.1: An example of a graph G',where $p = 6$. The graph G^* is the connected component on the left. Not all vertices of the subgraph G of G^* are drawn and not all graphs H_i on p vertices are displayed.

We do not know if Colouring-or-Subgraph is in $\mathsf{NP} = \Sigma_1^{\mathrm{P}}$. The problem arises when we try to check if an input G is H-free for some particular H, of size (say) $\lceil\sqrt{\log n}\rceil$, which takes time $n^{\lceil\sqrt{\log n}\rceil}$ by brute force. Note that it is crucial for our proof that the size of H depends on a function of n. We can however show that the problem belongs to the class Σ_2^{P}.

Theorem 2.2. Colouring-or-Subgraph *is in* Σ_2^{P}.

Proof. We can verify whether an input G is $\lceil\sqrt{\log n}\rceil$-colourable in NP. Let us explain how to verify if G is H-free for some graph H such that $|V(H)| \leq \lceil\sqrt{\log n}\rceil$. Plainly, we can guess existentially the graph H whose vertices are ordered $u_1, \ldots, u_{|V(H)|}$. Now we guess universally $|V(H)|$ vertices $v_1, \ldots, v_{|V(H)|}$ in G. Finally, we test whether the respective map of $u_1, \ldots, u_{|V(H)|}$ to $v_1, \ldots, v_{|V(H)|}$ has the property that u_iu_j is an edge in H if and only if v_iv_j is an edge in G. The latter can be accomplished in polynomial time and we are done. $\square$

2.2 PSPACE: Graph Colouring Reconfiguration

Let $G = (V, E)$ be a graph. A *clique* is a set of pairwise adjacent vertices in G. The set of neighbours of a vertex $v \in V$ is denoted by $N_G(v) = \{u \mid uv \in E\}$. The *$k$-colouring reconfiguration graph* $R_k(G)$ of G is the graph whose vertices are k-colourings of G and two vertices are adjacent if and only if the two corresponding k-colourings differ on exactly one vertex of G. In the following problem, k is a fixed constant, that is, k is not part of the input.

k-Colour-Path *Instance:* A graph G with two k-colourings α and β. *Question:* Does $R_k(G)$ contain a path from α to β?

Bonsma and Cereceda [17] proved that 3-Colour-Path is polynomial-time solvable, but for $k \geq 4$ they showed the following result, which holds even for bipartite graphs and which we will need in the next section.

Theorem 2.3 ([17])**.** *For every integer $k \geq 4$, the k-*Colour-Path *problem is* PSPACE*-complete.*

We define the following problem.

Colour-Path-or-Subgraph *Instance:* an n-vertex graph G with two p-colourings α and β for $p = \lceil \sqrt{\log n} \rceil$. *Question:* Does $R_p(G)$ contain a path from α to β, or does there exist a graph H with $\lvert V(H) \rvert \leq p$ such that G is H-free ?

Theorem 2.4. Colour-Path-or-Subgraph *is* PSPACE*-complete, but constant-time solvable for every hereditary graph class not equal to the class of all graphs.*

Proof. Let (G, α, β) be an instance of Colour-Path-or-Subgraph, where G is a graph on n vertices. Set $p = \lceil \sqrt{\log n} \rceil$. We can check if $R_p(G)$ contains a path from α to β using a polynomial amount of space using the same proof as used in Theorem 2.3 for 4-Colour-Path [17]. So we first prove membership to NPSPACE. As a certificate we can take a sequence of p-colourings of G and check in polynomial space if this sequence is an $\alpha - \beta$ path in $R_p(G)$: check if the first p-colouring is α; then check if the next p-colouring differs exactly at one place from the previous p-colouring and if so continue; finally check if the last p-colouring is β. Now, as PSPACE=NPSPACE due to Savitch's Theorem [86], this part of the problem belongs to PSPACE. Moreover, it also takes a polynomial amount of space to enumerate all graphs

H with $|V(H)| \leq \lceil\sqrt{\log n}\rceil$ and check if G is H-free by brute force. We conclude that Colour-Path-or-Subgraph belongs to PSPACE.

To prove PSPACE-hardness, we reduce from 4-Colour-Path, which is PSPACE-complete by Theorem 2.3. Let (G, α, β) be an instance of 4-Colour-Path, where G is an n-vertex graph and α and β are 4-colourings of G. We now set $p = \lceil\sqrt{\log 3n}\rceil$. We may assume without loss of generality that $p \geq 5$. From (G, α, β) we construct an instance (G', α', β') of Colour-Path-or-Subgraph. We first define a graph G^* as follows (see also Figure 2.2):

- take G;
- add a clique K of $p-4$ vertices $x_1, \ldots, x_{p-4}$;
- make each vertex of K adjacent to every vertex of G;
- add a clique L of four vertices $y_1, \ldots, y_4$;
- make each vertex of L adjacent to every vertex of K (so $K \cup L$ is a p-vertex clique and no vertex of L is adjacent to a vertex of G);

Let $\{H_1, \ldots, H_r\}$ be the set of all graphs with exactly p vertices and note that the number of vertices of $H_1 + \ldots + H_r$ is at most $p2^{\frac{p(p-1)}{2}} \leq \lceil\sqrt{\log 3n}\rceil \cdot \sqrt{3n} \leq n$ as $p \geq 5$. We now define the graph G' as the disjoint union of G^* and the graphs $H_1, \ldots, H_r$; see also Figure 2.2. Note that:

$$|V(G')| = |V(G)| + |K| + |L| + |V(H_1)| + \ldots + |V(H_r)| \leq n + p - 4 + 4 + n < 3n.$$

By adding isolated vertices we may assume that $|V(G')| = 3n$.

We now define α' and β' as p-colourings of G':

- let $\alpha' = \alpha$ and $\beta' = \beta$ on G;
- for $h \in \{1, \ldots, 4\}$, let $\alpha'(y_h) = \beta'(y_h) = h$;
- for $i \in \{1, \ldots, p-4\}$, let $\alpha'(x_i) = \beta'(x_i) = i + 4$;

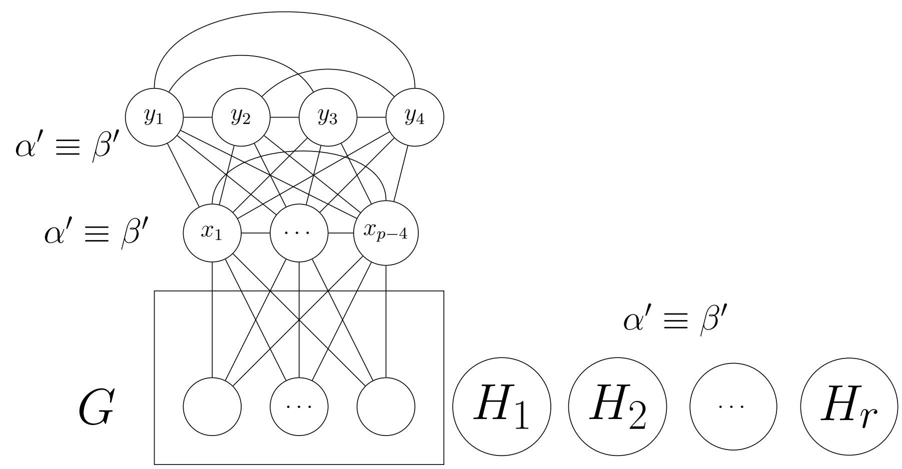

Figure 2.2: The graph G'. The graph G^* is the connected component on the left. By construction, it holds that $\alpha' \equiv \beta'$ on $V(G') \setminus V(G)$.

- for $j \in \{1, \ldots, r\}$, let $V(H_j) = \{z_1^j, \ldots, z_s^j\}$ and let for $q \in \{1, \ldots, p\}$, $\alpha'(z_q^j) = \beta'(z_q^j) = q$.

We set $\alpha'(u) = \beta'(u) = 1$ for each isolated vertex u of G' that we have not yet coloured. By construction, α' and β' are p-colourings of G', in particular because every H_j has p vertices. We claim that (G, α, β) is a yes-instance of 4-Colour-Path if and only if (G', α', β') is a yes-instance of Colour-Path-or-Subgraph.

First suppose that (G, α, β) is a yes-instance of 4-Colour-Path. Then there exists a path from α to β in $R_4(G)$. We mimic this path in $R_p(G')$, as we can keep the colour $\alpha'(u) = \beta'(u)$ of each vertex u of G' that does not belong to G the same. Hence, (G', α', β') is a yes-instance of Colour-Path-or-Subgraph.

Now suppose that (G', α', β') is a yes-instance of Colour-Path-or-Subgraph. By construction, G' contains every graph on p vertices, and thus every graph on at most p vertices, as an induced subgraph. As (G', α', β') is a yes-instance of Colour-Path-or-Subgraph and $p = \lceil \sqrt{\log 3n} \rceil =$

$\lceil\sqrt{\log|V(G')|}\rceil$, this means that $R_p(G')$ contains a path $\alpha'\gamma'_1\cdots\gamma'_t\beta'$ from α' to β'. As α' coincides with β' on every H_j, we may assume without loss of generality that for every $i \in \{1,\ldots,t\}$ and every vertex z of every graph H_j, $\gamma'_i(z) = \alpha'(z) = \beta'(z)$. Moreover, for every vertex v of the clique $K \cup L$, the set of colours used by both α' and β' on the vertices of $N(v)\cup\{v\} = \{1,\ldots,p\}$. Hence, these vertices are "frozen", that is, we cannot change their colour, so for every $i \in \{1,\ldots,t\}$ and every $v \in K \cup L$ we have that $\gamma'_i(v) = \alpha'(v) = \beta'(v)$. Let γ_i be the restriction of γ'_i to $V(G)$. Then, from the above, we conclude that $\alpha\gamma_1\cdots\gamma_t\beta$ corresponds to a path from α to β in $R_4(G)$. Hence, (G,α,β) is a yes-instance of 4-Colour-Path.

We now prove the second part of the theorem. Let $\mathcal{G}$ be a hereditary graph class that is not the class of all graphs. Then there exists at least one graph H such that every graph $G \in \mathcal{G}$ is H-free. Let $\ell = |V(H)|$. We claim that Colour-Path-or-Subgraph is constant-time solvable for $\mathcal{G}$. Let $G \in \mathcal{G}$ be an n-vertex graph and let α and β be two p-colourings of G. If $n \leq 2^{\ell^2}$, then G has constant size and the problem is constant-time solvable. If $n > 2^{\ell^2}$, then

$$|V(H)| = \ell < \sqrt{\log n} \leq \lceil\sqrt{\log n}\rceil.$$

Hence, (G,α,β) is a yes-instance of Colour-Path-or-Subgraph, as G is H-free and H has at most $\lceil\sqrt{\log n}\rceil$ vertices. □

2.3 NEXPTIME: Succinct Graph Colouring

A *Boolean circuit* $\phi(x_1,\ldots,x_m,y_1,\ldots,y_m)$ with $2m$ variables defines a graph G on 2^m vertices, represented by vectors $(x_1,\ldots,x_m)$ of length m, according to the rule that there is an edge $(x_1,\ldots,x_m)(y_1,\ldots,y_m)$ between two vertices $(x_1,\ldots,x_m)$ and $(y_1,\ldots,y_m)$ if and only if $\phi(x_1,\ldots,x_m,y_1,\ldots,y_m)$ is true. This allows that some graph families with an exponential number of vertices 2^m can be expressed by circuits of size polynomial in m. Plainly, this can not be the case in general and indeed graphs whose vertex set is not of size a power of 2 can only be expressed up to the addition of extra isolated vertices.

Let us note how any graph on n vertices can be expressed by a circuit with $2n$ variables that has size at most $2n^3$ in what we call the (naive) *longhand* method. We will apply this method in the proof of the result in this section. The n vertices are represented by vectors of length n, namely as

$$(1,0,\ldots,0,0),(0,1,\ldots,0,0),\ldots,(0,0,\ldots,0,1),$$

and all of the remaining $2^n - n$ vertices are isolated. If we have an edge ij, then this adds a new disjunction to the circuit of the form

$$x_i \wedge y_j \wedge \bigwedge_{i\neq\ell\in[n]} \neg x_\ell \wedge \bigwedge_{j\neq\ell\in[n]} \neg y_\ell.$$

Thus, the circuit is in fact in disjunctive normal form.

Suppose we have a graph G represented by a Boolean circuit denoted $\phi(x_1,\ldots,x_m,y_1,\ldots,y_m)$. We can add k vertices to it, again in a longhand way, by expanding the number of variables from $2m$ to $2(m+k)$. In line with our previous longhand method, all vertices other than those of the form $(x_1,\ldots,x_m,0,\ldots 0)$ (the original vertices of G) and the k new vertices of the form

$$(\overbrace{0,\ldots,0}^{m \text{ times}},\overbrace{1,0,\ldots,0,0}^{k \text{ times}})$$
$$\vdots$$
$$(\overbrace{0,\ldots,0}^{m \text{ times}},\overbrace{0,0,\ldots,0,1}^{k \text{ times}})$$

are isolated. It is known that the problem Succinct 3-Colouring, which takes as input a Boolean circuit $\phi(x_1,\ldots,x_m,y_1,\ldots,y_m)$ defining a graph G on 2^m vertices, and has yes-instances precisely those such that G is properly 3-colourable, is NEXPTIME-complete. For a proof of this result together with a discussion on succinctly encoded problems we refer to [83]. We wish to consider the following variant of the problem.

Succinct Colouring-or-Subgraph	
Instance:	a Boolean circuit $\phi(x_1,\ldots,x_m,y_1,\ldots,y_m)$ defining a graph G on 2^m vertices
Question:	is G $\lceil\sqrt{\log m}\rceil$-colourable or H-free for some graph H with $\lvert V(H)\rvert \leq \lceil\sqrt{\log m}\rceil$?

Note that, relative to Colouring-or-Subgraph, the number of vertices of the graph was mapped to half the number of variables in the circuit.

Theorem 2.5. *The* Succinct Colouring-or-Subgraph *problem is* NEXPTIME*-complete, but constant-time solvable for every hereditary graph class not equal to the class of all graphs.*

Proof. We first argue for NEXPTIME membership. Let $G = (V, E)$ be a succinct graph on 2^m vertices that is defined by the following Boolean circuit $\phi(x_1, \ldots, x_m, y_1, \ldots, y_m)$. Let $p = \lceil \sqrt{\log m} \rceil$. The question as to whether G is p-colourable can be solved in NEXPTIME by guessing the colouring and checking whether adjacent vertices are coloured distinctly. For checking whether G is H-free for some graph H with $|V(H)| \leq p$, it suffices to consider only graphs H on exactly p vertices. This can be answered, even in EXPTIME, by the following naive algorithm that checks all possibilities of choosing such a graph H one by one. To analyze the running time of this algorithm we observe the following:

1. the number of graphs on p vertices is at most $2^{\frac{p(p-1)}{2}} \leq m$; and

2. checking if a graph H with p vertices is isomorphic to an induced subgraph of G takes $O(p^2|V|^p) = O(\log m \cdot 2^{m\lceil\sqrt{\log m}\rceil} = O(2^{m^2})$ time (we can consider all mappings from H to G by brute force and for each of them we check if edges of H map to edges of G).

Hence, the total running time of the naive algorithm for checking if G has no graph H on p vertices is $O(m2^{m^2})$.

To prove NEXPTIME-hardness we reduce from the problem Succinct 3-Colouring. Let G be a succinct graph defined by a Boolean circuit $\phi(x_1, \ldots, x_m, y_1, \ldots, y_m)$. We now set $p = \lceil \sqrt{\log 3m} \rceil$. Add $p - 3$ pairwise adjacent vertices to G in the longhand manner discussed above, so the increase in size is at most $2(p-3)^3 \leq 2 \log 3m \sqrt{\log 3m}$. We will make the new vertices also adjacent to every vertex of G. We can specify the latter by adding to ϕ a series of disjuncts for each $j \in \{1, \dots, p-3\}$, each with $2(p-3)$ variables, encoding the conjunctions

$$\bigwedge_{1 \le i \le p-3} \neg x_{m+i} \wedge \neg \bigwedge_{i \ne j \in \{1,\dots,p-3\}} \neg y_{m+i} \wedge y_{m+j},$$

of total size at most $2(p-3)^2 \le 2 \log 3m$.

We now consider the disjoint union G' of the new graph and all possible graphs on p vertices. Again we do this long hand, so for each graph on p vertices we also add $2^p - p$ isolated vertices to G'. This will require the addition of at most $2p \cdot 2^{\frac{p(p-1)}{2}} \le 2m$ variables, for sufficiently large m, giving a size increase of at most $16m^3$ as we work in longhand.

The circuit ϕ' specifying G' has at most $2m+2(p-3)+2m < 6m$ variables, let us make it up to precisely $6m$, half of which is $3m$. Furthermore, it is of size at most the size of ϕ plus $O(m^3)$. By construction, G' contains every graph on p vertices, and thus every graph on at most p vertices, as an induced subgraph. Hence, we deduce in exactly the same way as in the proof of Theorem 2.1 that G' is a yes-instance of SUCCINCT COLOURING-OR-SUBGRAPH if and only if G' is p-colourable, and that the latter holds if and only if G is 3-colourable.

We now prove the second part of the theorem. We do this in the same way as before. Let $\mathcal{G}$ be a hereditary graph class that is not the class of all graphs. Then there exists at least one graph H such that every graph $G \in \mathcal{G}$ is H-free. Let $\ell = |V(H)|$. We claim that SUCCINCT COLOURING-OR-SUBGRAPH is constant-time solvable for $\mathcal{G}$. Let $G \in \mathcal{G}$ be a graph given by a Boolean circuit $\phi(x_1, \dots, x_m, y_1, \dots, y_m)$ which has 2^m vertices. If $m \le 2^{\ell^2}$, then G has constant size and the problem is constant-time solvable. If $m > 2^{\ell^2}$, then

$$|V(H)| = \ell < \sqrt{\log m} \le \lceil \sqrt{\log m} \rceil.$$

Hence G is a yes-instance of SUCCINCT COLOURING-OR-SUBGRAPH, as G is H-free and H has at most $\lceil \sqrt{\log m} \rceil$ vertices. $\square$

Chapter 3

Colouring H-free Graphs of Bounded Diameter

In this chapter we consider both the Colouring and k-Colouring problems for H-free graphs of bounded diameter. We first survey known results. In Section 3.3 we examine the effect of bounding both the diameter and girth of the input graph. In Section 3.4 we present both NP-completeness and polynomial-time results for polyad-free graphs of bounded diameter. Finally, in Section 3.5 we give both polynomial-time and NP-completeness results for graphs avoiding one or two short cycles.

3.1 Known Results

The computational complexity of Colouring has been fully classified for H-free graphs in the following theorem of Kral et al.

Theorem 3.1 ([62]). *If H is an induced subgraph of $P_1 + P_3$ or of P_4, then* Colouring *for H-free graphs is polynomial-time solvable, otherwise it is* NP-*complete.*

In contrast, the complexity classification for k-Colouring restricted to H-free graphs is still incomplete. It is known that for every $k \geq 3$, k-Colouring for H-free graphs is NP-complete if H contains a cycle [34] or an induced claw [50, 65]. However, the remaining case where H is a linear forest has not been settled yet, even if H consists of a single path. For P_t-free graphs, the cases $k \leq 2$, $t \geq 1$ (trivial), $k \geq 3$, $t \leq 5$ [48], $k = 3$, $6 \leq t \leq 7$ [16] and $k = 4$, $t = 6$ [25] are polynomial-time solvable and the cases $k = 4$, $t \geq 7$ [52] and $k \geq 5$, $t \geq 6$ [52] are NP-complete. The cases where $k = 3$ and $t \geq 8$ are still open. For further details, including for linear forests H of more than one connected component, see the survey paper [41] or [24, 44, 60].

We remark that, unlike the class of H-free graphs, the class of H-free graphs of diameter at most d for some integer d is not hereditary. We also note that, by a straightforward reduction from 3-Colouring, one can show that k-Colouring is NP-complete for graphs of diameter d for all pairs (k, d) with $k \geq 3$ and $d \geq 2$ except for two cases, namely $(k, d) \in \{(3, 2), (3, 3)\}$. Whilst the case $(k, d) = (3, 2)$ is still open, Mertzios and Spirakis settled the case $(k, d) = (3, 3)$ by proving the following theorem:

Theorem 3.2 ([78]). 3-Colouring *is* NP*-complete even for C_3-free graphs of diameter 3.*

Regarding graphs of bounded diameter and girth, we also make use of the following result. It includes the Hoffman-Singleton Theorem which provides a description of regular graphs of diameter 2 and girth 5.

Theorem 3.3 ([31] [49] [91]). *For every $d \geq 1$, every graph of diameter d and girth $2d+1$ is p-regular for some integer p. Moreover, if $d = 2$, there are only four possible values of p (2,3,7 and 57) and if $d \geq 3$ then such graphs are odd cycles of length $2d + 1$.*

Finally, the polynomial-time results in this section are obtained using the strategy of reducing an instance of List 3-Colouring to a polynomial number of instances of 2-List Colouring which can be solved in linear time due to the following well known theorem.

Theorem 3.4 ([33]). 2-List Colouring *is linear-time solvable.*

3.2 Our Results

We complement the bounded diameter results of Mertzios and Spirakis [78] by presenting a set of new results for Colouring and k-Colouring for H-free graphs of bounded diameter when H contains a claw or a cycle. Results for the case where H has a cycle usually follow from stronger results for graphs of girth at least g for some fixed integer g. In particular, Emden-Weinert, Hougardy and Kreuter proved the following theorem.

Theorem 3.5 ([34])**.** *For all integers $k \geq 3$ and $g \geq 3$, k-Colouring is NP-complete for graphs with girth at least g and with maximum degree at most $6k^{13}$.*

First, in Section 3.3 we perform a similar study for graphs of bounded diameter and girth. We provide new polynomial-time and NP-hardness results for k-Colouring, identifying and narrowing the gap between tractability and intractability, in particular for the case where $k = 3$ (see also Figure 3.2).

Second, in Section 3.4 we research the effect of bounding the diameter of k-Colouring and Colouring restricted to polyad-free graphs for various polyads. Our first result, which formed together with the result of [78] the starting point of our investigation, is that k-Colouring is constant-time solvable for $K_{1,r}$-free graphs of diameter d for any fixed integers $d \geq 1$, $k \geq 1$ and $r \geq 1$. We also show that this does not hold for Colouring (when k is part of the input). We then extend these results for larger polyads; see also Figure 3.1. We then focus on 3-Colouring for C_s-free or (C_s, C_t)-free graphs of diameter 2 for small values of s and t; in particular for the case where $s = 4$. In fact we prove our results for the more general problem List 3-Colouring, whose complexity for diameter 2 is also still open. We complement these results with an NP-completeness result for diameter 4.

3.3 Bounded Diameter and Girth

We now examine the trade-offs for k-Colouring between diameter and girth. Recall that Mertzios and Spirakis proved that 3-Colouring is NP-complete for graphs of diameter at most 3 and girth at least 4 in Theorem

Colours	Diameter	H-free	Complexity	Theorem
fixed k	d	$K_{1,r}$	P	3.7
input k	d	$K_{1,4}$	NP-c	3.8
3	d	$K_{1,3}^1$	P	3.10(1)
3	2	$K_{1,r}^2$	P	3.10(2)
3	4	$K_{1,4}^3$	NP-c	3.10(3)
4	2	$K_{1,3}^1$	NP-c	3.10(4)
3	2	$S_{1,2,2}$	P	3.11

Table 3.1: Our polynomial-time (P) and NP-complete (NP-c) results for polyad-free graphs.

3.2. We extend this result in our next theorem. Note that there are still a number of open cases where the complexity of 3-Colouring remains open for graphs of diameter d and girth at least g.

Theorem 3.6. *Let d, g, k be three integers with $d \geq 2$, $g \geq 3$ and $k \geq 3$. Then k-Colouring for graphs of diameter at most d and girth at least g is*

1. *Polynomial-time solvable if $g \geq 2d + 1$*
2. NP*-complete if $d = 3$ and $g \leq 4$ and $k = 3$*
3. NP*-complete if $4p \leq d \leq 4p + 3$ and $g \leq 4p + 2$ for some integer $p \geq 1$ and $k = 3$.*

Proof. **1.** This case follows from Theorem 3.3. **2.** This case is Theorem 3.2.

3. We reduce 3-Colouring for graphs of girth at least $8p - 3$, which is NP-complete by Theorem 3.5, to 3-Colouring for graphs of diameter at most $4p$ and girth at least $4p + 2$. Construct the graph G' as follows (see Figure 3.1 for an example):

- label the vertices of G v_1 to v_n;

girth / diameter	≥ 3	≥ 4	≥ 5	≥ 6	≥ 7	≥ 8	≥ 9	≥ 10	≥ 11	≥ 12
≤ 1	P	P	P	P	P	P	P	P	P	P
≤ 2	?	?	P	P	P	P	P	P	P	P
≤ 3	NP-c	NP-c	?	?	P	P	P	P	P	P
≤ 4	NP-c	NP-c	NP-c	NP-c	?	?	P	P	P	P
≤ 5	NP-c	NP-c	NP-c	NP-c	?	?	?	?	?	P

Table 3.2: The complexity of 3-COLOURING for graphs of diameter at most d and girth at least g.

- for each vertex of G, add a new neighbour $v_{i,1}$;
- for every two vertices v_i and v_j such that $\text{dist}(v_i, v_j) > l = 2p - 1$ add new vertices to form the path $v_{i,1}v_{i,2,j}...v_{i,p+1,j}v_{j,p,i}...v_{j,1}$.

First we show that G' has diameter at most $4p$. For any two vertices v_i and v_j of G either $\text{dist}(v_i, v_j) \leq l$ or we have the path $v_{i,1}v_{i,2,j}...v_{i,p+1,j}v_{j,p,i}...v_{j,1}$ and $\text{dist}(v_i, v_j) \leq 2p+2$. Similarly, $\text{dist}(v_i, v_{j,1}) \leq 2p+1$ and $\text{dist}(v_{i,1}, v_{j,1}) \leq 2p+1$. Now consider two vertices $v_{a,r,b}$ and $v_{c,q,d}$ for $2 \leq r \leq p+1$, $2 \leq q \leq p+1$. If $\text{dist}(v_a, v_c) \leq l$ then $\text{dist}(v_{a,r,b}, v_{c,q,d}) \leq r + q + l \leq (p+1) + (p+1) + (2p-1) \leq 4p+1$.
Otherwise we have the path $v_{a,r,b}..v_{a,1}v_{a,2,c}...v_{a,p+1,c}v_{c,p,a}...v_{c,1}v_{c,2,d}...v_{c,q,d}$. This gives $\text{dist}(v_{a,r,b}, v_{c,q,d}) \leq (r-1) + p + p + (q-1) \leq 4p$.
In fact, if $\text{dist}(v_{a,r,b}, v_{c,q,d}) = 4p+1$, then we must have $r = q = p+1$ and $\text{dist}(v_a, v_c) = \text{dist}(v_a, v_d) = \text{dist}(v_b, v_c) = \text{dist}(v_b, v_d) = 2p-1$. In this case we have two paths of length at most $4p-2$ between v_a and v_b, one containing v_c and the other containing v_d. These paths must be distinct since the existence of the vertex $v_{c,p+1,d}$ implies that $\text{dist}(v_c, v_d) > 2p-1$. Therefore we have a cycle in G of length at most $8p-4$ which contradicts the assumption that G has girth at least $8p-3$. This implies that the diameter of G' is at most $4p$.

Since G has girth at least $8p-3$, every cycle in G' of length less than

$4p+2$ must contain at least one vertex of $V(G')\setminus V(G)$. Since all the vertices of $V(G')\setminus V(G)$ except the vertices $v_{i,1}$ have degree 2, any such cycle C must contain the path $v_{i,1}..v_{i,p+1,j}...v_j$ for some v_i, v_j at distance greater than l. This path has length $2p+1$. If C contains $v_{i,2,m}$ for some m different from j then it contains the path $v_{i,2,m}...v_{m,1}$ and has length at least $4p+2$. Similarly, this is the case if C contains $v_{j,2,m}$ for m different from i. Otherwise C contains v_i and v_j which are at distance at least l and has length at least $(2p+1)+2+(2p-1)=4p+2$.

Finally, we show that G is 3-colourable if and only if G' is 3-colourable. The latter holds if and only if the subgraph G'' of G' induced by $V(G)\cup \{v_{i,1} \mid 1\leq i\leq n\}$ is 3-colourable, since every other vertex of G' has degree 2. The graph G is 3-colourable if and only if G'' is 3-colourable, since G is an induced subgraph of G'' and each vertex of $V(G'')\setminus V(G)$ has degree 1. Therefore, G is 3-colourable if and only if G' is 3-colourable. $\square$

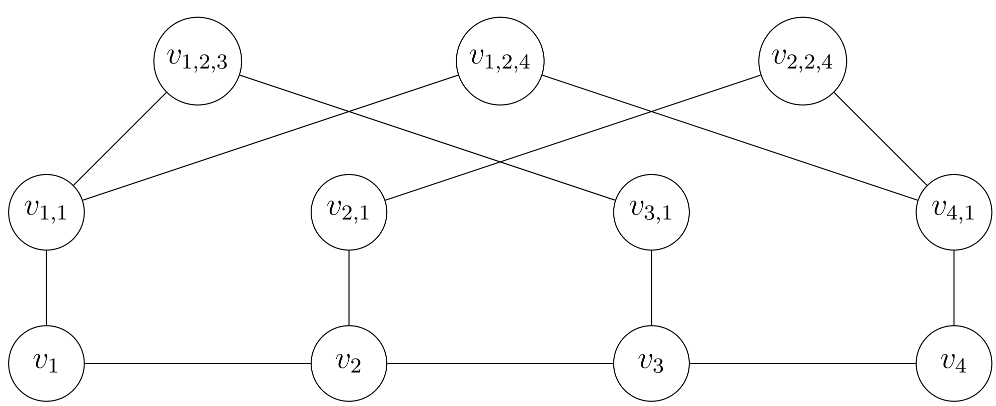

Figure 3.1: An example of a graph G', constructed in the proof of Theorem 3.6(3), for $p=1$.

3.4 Polyad-Free Graphs of Bounded Diameter

We first make an observation required for several of the proofs that follow.

Lemma 3.1. *If G is a graph of diameter d that is not a tree, then G contains an induced cycle of length at most $2d + 1$.*

Proof. As G is not a tree and G is connected, G must contain a cycle C. Suppose that C has length at least $2d + 2$. Since G has diameter d, there exists a path of length at most d in G between any two vertices u and v at distance $d + 1$ in C. The vertices of this path, together with the vertices of the path of length $d+1$ between u and v on C, induce a subgraph of G that contains an induced cycle C' of length at most $2d + 1$. □

We prove a second Lemma which we will use later.

Lemma 3.2. *Let G be a non-bipartite graph of diameter 2. Then G contains a C_3 or induced C_5.*

Proof. As G is non-bipartite, G has an odd cycle. Let C be an odd cycle in G of minimum length. Then C is induced; otherwise we would find a shorter odd cycle. For contradiction, suppose that C has length at least 7. Consider two vertices u and v at distance 3 in C. Then C contains a 4-vertex path $uxyv$ for some $x, y \in V(C)$. As C is induced, u and v are non-adjacent. Hence, there exists a vertex w not on C that is adjacent to u and v (as G has diameter 2). Then the subgraph of G induced by $\{u, v, w, x, y\}$ contains a C_3 or an induced C_5, contradicting the minimality of C. □

We now state our first result.

Theorem 3.7. *For all integers $d, k, r \geq 1$, k-*Colouring *is constant-time solvable for $K_{1,r}$-free graphs of diameter d.*

Proof. Let $G = (V, E)$ be a $K_{1,r}$-free graph of diameter d. We prove that if G has size larger than some constant $\beta(k, r)$, which we determine below, then G is not k-colourable. If $|V(G)| \leq \beta(k,r)$, we can solve k-Colouring in constant time.

As G is $K_{1,r}$-free, Ramsey's Theorem tells us that the neighbourhood of every vertex $u \in V$ with degree at least $R(k,r)$ contains a clique of size k. In that case $N(u) \cup \{u\}$ is a clique of size $k+1$. Hence, to be k-colourable, every vertex of G must have degree less than $R(k,r)$, so G must have at most $\beta(k,r) = 1 + R(k,r) + R(k,r)^2 + \ldots + R(k,r)^d$ vertices. $\square$

The following theorem demonstrates that this result no longer holds if k is part of the input. Note that we exclude graphs $H \subseteq_i P_1 + P_3$ and $H \subseteq_i P_4$ since Colouring is polynomial-time solvable for H-free graphs of this form. Also note that the only graph H whose complexity is not classified for any diameter d in this theorem is $H = K_{1,3}$.

Theorem 3.8. *Let H be a graph with $H \not\subseteq_i P_1 + P_3$ and $H \not\subseteq_i P_4$ and d be an integer. Then* Colouring *for H-free graphs of diameter at most d is*

1. NP*-complete if H has no dominating vertex u such that $H - u \subseteq_i P_1 + P_3$ or $H - u \subseteq_i P_4$ and $d \geq 2$;*
2. NP*-complete if $H \neq K_{1,3}$ and H has a dominating vertex u such that $H - u \subseteq_i P_1 + P_3$ or $H - u \subseteq_i P_4$ and $d \geq 3$.*

Proof. **1.** Let H have no dominating vertex u such that $H - u \subseteq_i P_1 + P_3$ or $H - u \subseteq_i P_4$. We define H' as $H - u$ if H has a dominating vertex u and as H itself otherwise. By construction, $H' \not\subseteq_i P_1 + P_3$ and $H' \not\subseteq_i P_4$. Hence, Colouring is NP-complete for H'-free graphs due to Theorem 3.1. Let G be an H'-free graph. Add a dominating vertex to G. The new graph G' has diameter 2 and is H-free. Moreover, G is k-colourable if and only if G' is $(k+1)$-colourable.

2. Let $H \neq K_{1,3}$ have a dominating vertex u such that $H - u \subseteq_i P_1 + P_3$ or $H - u \subseteq_i P_4$. Then H cannot be a forest, as in that case H would be in $\{P_1, P_2, P_3, K_{1,3}\}$. Hence, H has an induced cycle C_r for some $r \geq 3$. If $r = 3$, then 3-Colouring is NP-complete for H-free graphs of diameter 3, as it is so for C_3-free graphs of diameter 3 due to Theorem 3.2. If $r \geq 4$, then COLOURING is NP-complete even for H-free graphs of diameter 2, as this is the case for C_r-free graphs of diameter 2 due to **1**. $\square$

We may then ask whether the result of Theorem 3.7 can be extended from H-free graphs where H is a star to the case where H is a larger tree. For instance, we first consider the case where H is an l-subdivided star $K_{1,r}^l$ with $r \geq 3, l \geq 1$. We now prove a result on C_5-free graphs of diameter 2 which is necessary in the proofs to follow.

Theorem 3.9. 3-COLOURING *can be solved in polynomial time for C_5-free graphs of diameter at most* 2.

Proof. If G is bipartite, then G is 3-colourable. If G contains a K_4, then G is not 3-colourable. We check these properties in polynomial time, and from now on we assume that G is K_4-free and non-bipartite. The latter implies that G must have an odd induced cycle C_r for some odd integer r. As G has diameter 2, we find that $r \leq 5$ due to Lemma 3.2. As G is C_5-free, it follows that $r = 3$.

Let C be a triangle in G. We write $N_0 = V(C) = \{x_1, x_2, x_3\}$, $N_1 = N(V(C))$ and $N_2 = V(G) \setminus (N_0 \cup N_1)$. As G has diameter 2, for every $i \in \{1, 2, 3\}$, it holds that every vertex in N_2 has a neighbour in N_1 that is adjacent to x_i.

We let T consist of all vertices of N_2 that have a neighbour in N_1 that is adjacent to exactly two vertices of N_0. We claim that $N_2 = T$. In order to see this, let $u \in N_2$. If u has a neighbour $y \in N_1$ adjacent to every x_i, then G contains a K_4, a contradiction. Hence, u must have three distinct neighbours y_1, y_2, y_3, such that for $i \in \{1, 2, 3\}$, it holds that $N(y_i) \cap N_0 = \{x_i\}$. If $\{y_1, y_2, y_3\}$ is a clique, then G has a K_4 on vertices u, y_1, y_2, y_3, a contradiction. Hence, we may assume without loss of generality that y_1 and y_2 are non-adjacent. However, then $\{u, y_1, x_1, x_2, y_2\}$ induces a C_5 in G, another contradiction. We conclude that $N_2 = T$.

If G has a 3-colouring c, then we may assume without loss of generality that $c(x_i) = i$ for $i \in \{1,2,3\}$. Hence, our algorithm assigns colours 1, 2, 3 to x_1, x_2, x_3, respectively. This reduces the list of admissible colours of the vertices of N_1 by at least one colour. In particular, vertices in N_1 that have two neighbours in N_0 can be coloured with only one colour. Our algorithm assigns this colour to such vertices. This means that any of their neighbours in $T = N_2$ can be coloured with at most two colours. So, after propagation, we have obtained either two adjacent vertices that are coloured alike, in which case G is not 3-colourable, or we have constructed an instance of 2-List Colouring. We can solve such an instance in linear time due to Theorem 3.4. □

We are now ready to state our results for $K_{1,r}^{l}$-free graphs of diameter at most d. Note that we exclude the cases which are tractable in general, namely $d = 1$, $k \leq 2$ or $r \leq 2$.

Theorem 3.10. *Let d, k, ℓ, r be four integers with $d \geq 2$, $k \geq 3$, $\ell \geq 1$ and $r \geq 3$. Then k-Colouring for $K_{1,r}^{\ell}$-free graphs of diameter at most d is:*

1. *Polynomial-time solvable if $d \geq 2$, $k = 3$, $\ell = 1$ and $r = 3$*
2. *Polynomial-time solvable if $d = 2$, $k = 3$, $\ell = 2$ and $r \geq 3$*
3. NP*-complete if $d \geq 4$, $k = 3$, $\ell \geq 3$ and $r \geq 4$*
4. NP*-complete if $d \geq 2$, $k \geq 4$, $\ell \geq 1$ and $r \geq 3$.*

Proof. **1.** Recall that $K_{1,3}^{1}$ is the chair $S_{1,1,2}$. Let G be a chair-free graph of diameter d. If G is a tree, then G is even 2-colourable. We check in $O(n^4)$ time if G has a K_4. If so, then G is not 3-colourable. From now on we assume that G is not a tree and that G is K_4-free. As G is not a tree and G is connected, G contains an induced cycle of length at most $2d + 1$ by Lemma 3.2. We can find a largest induced cycle C of length at most $2d + 1$ in $O(n^{2d+1})$ time. Let $|V(C)| = p$. We write $N_0 = V(C) = \{x_1, x_2, \ldots, x_p\}$

and for $i \geq 1$, $N_i = N(N_{i-1}) \setminus N_{i-2}$. So the sets N_i partition $V(G)$, and the distance of a vertex $u \in N_i$ to N_0 is i. **Case 1.** $4 \leq p \leq 2d + 1$.

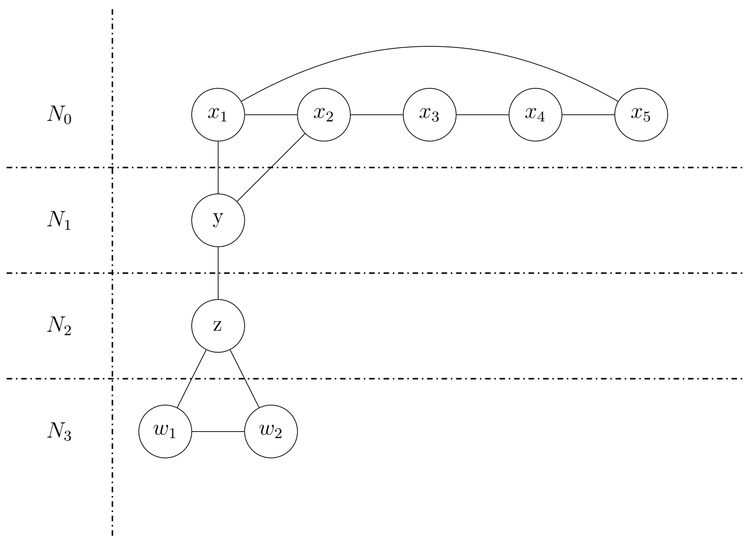

Figure 3.2: An example of a decomposition of a chair-free graph of diameter 3 into sets $N_0, \ldots, N_3$ where $p = 5$ and $y \in N_1$ has two "descendants" in N_3. To prevent an induced chair, y must be adjacent to exactly two (adjacent) vertices of N_0, and w_1 and w_2 must be adjacent to each other.

This case is illustrated in Figure 3.2. We consider every possible 3-colouring of C. Let c be such a 3-colouring. Every vertex with two differently coloured neighbours can only be coloured with one remaining colour. We assign this unique colour to such a vertex and apply this rule as long as possible. This takes polynomial time. The remaining vertices have a list of admissible colours that either consists of two or three colours, and vertices in the latter case belong to $V(G) \setminus (N_0 \cup N_1)$ (as $N(N_0) = N_1$).

If $N_2 = \emptyset$, then $V(G) = N_0 \cup N_1$. Then, we obtained an instance of 2-List Colouring, which we can solve in linear time due to Theorem 3.4. Now assume that $N_2 \neq \emptyset$. Let $z \in N_2$. Then z has a neighbour $y \in N_1$, which in turn has a neighbour $x \in N_0$. If y is adjacent to neither neighbour of x in N_0, then z, y, x and these two neighbours induce a chair in G, a contradiction. Hence, y must be adjacent to at least one neighbour of x on N_0, meaning that y must have received a colour by our algorithm. Consequently, z must have a list of admissible colours of size at most 2.

From the above we deduce that every vertex in N_2 has only two available colours in its list. We now consider the vertices of N_3. Let $z' \in N_3$. Then z' has a neighbour $z \in N_2$, which in turn has a neighbour $y \in N_1$, which in turn has a neighbour $x \in N_0$, say $x = x_1$. If y has two non-adjacent neighbours in N_0, then z', z, y and these two non-adjacent neighbours of y induce a chair in G, a contradiction. Combined with the fact deduced above, we conclude that y must have exactly two neighbours in N_0 and these two neighbours must be adjacent, say x_2 is the other neighbour of y in N_0.

Suppose x_1 and x_2 are both adjacent to a vertex $y' \in N_1 \setminus \{y\}$ that is adjacent to a vertex in N_2 that has a neighbour in N_3. Then, just as in the case of vertex y, the two vertices x_1 and x_2 are the only two neighbours of y' in N_0. If y and y' are not adjacent, this means that x_2, x_3, x_4, y, y' induce a chair in G, a contradiction. Hence y and y' must be adjacent. However, then x_1, x_2, y, y' form a K_4, a contradiction. This means that every pair of adjacent vertices of N_0 can have at most one common neighbour in N_1 that is adjacent to a vertex in N_2 with a neighbour in N_3. We already deduced that every vertex of N_1 with a "descendant" in N_3 has exactly two neighbours in N_0, which are adjacent. Hence, we conclude that the number of such vertices of N_1 is at most p.

We now observe that for $i \geq 2$, every vertex in N_i has at most two neighbours in N_{i+1}. This can be seen as follows. If $v \in N_i$ has two non-adjacent neighbours w_1, w_2 in N_{i+1}, then we pick a neighbour u of v in N_{i-1}, which has a neighbour t in N_{i-2}. Then v, u, t, w_1, w_2 induce a chair in G, a contradiction. Hence, the N_{i+1} neighbourhood of every vertex in N_i is a clique, which must have size at most 2 due to the K_4-freeness of G. As the number of vertices in N_1 with a 'descendant' in N_3 is at most p, this means that there are at most $2^{i-1}p$ vertices in N_i with a neighbour in N_{i+1}. Therefore the total number of vertices not belonging to any of the sets N_0, N_1 or N_2 is at most $\sum_{i=3}^{d} 2^{i-1}p$.

This means the total number of vertices not belonging to N_1 or N_2 is at most $\beta(d) = \sum_{i=3}^{d} 2^{i-1}p + p \leq \sum_{i=3}^{d} 2^{i-1}(2d+1) + 2d + 1$. Let T_c be this set. We consider every possible 3-colouring of $G[T_c]$. As we already deduced that the vertices in $N_1 \cup N_2$ have a list of size at most 2, for each case we obtain an instance of 2-List Colouring, which we can solve in linear time due to Theorem 3.4. As the total number of instances we need to consider is at most $3^p \times 3^{\beta(d)} \leq 3^{2d+1} \times 3^{\beta(d)}$, our algorithm runs in polynomial time.

Case 2. $p = 3$.
As p was the size of a largest induced cycle of length at most $2d + 1$ and $2d + 1 \geq 5$, we find that G is C_4-free. As G is K_4-free, each vertex of N_1 is adjacent to at most two vertices of N_0. We call a vertex of N_1 which is adjacent to exactly one N_0 vertex a private neighbour of this vertex.

If a vertex $x \in N_0$ has two independent private neighbours u and v in N_1 with respect to N_0, then every neighbour w of u in N_2 must also be a neighbour of v and vice versa, since G is chair-free. However, this is not possible, as x, u, w, v induce a C_4. We conclude that u and v must be adjacent. Therefore, as G is K_4-free, every vertex of N_0 has at most two private neighbours in N_1, with respect to N_0, that have a neighbour in N_2.

By the same arguments as above we deduce that every two vertices of N_0 have at most one common neighbour in N_1 that is adjacent to a vertex in N_2. Combined with the above, we find that there at most $6 + 3 = 9$ vertices in N_1 that have a neighbour in N_2. If a vertex in N_1 has two independent neighbours in N_2, then G contains an induced chair, which is not possible. Hence the neighbourhood of a vertex in N_1 in N_2 is a clique, which has size at most 2 due to the K_4-freeness of G. We conclude that $|N_2| \leq 9 \times 2 = 18$. Similarly, every vertex in N_i for $i \geq 3$ has at most two neighbours in N_{i+1}. Therefore the number of vertices in N_i for $i \geq 3$ is at most $18 \times 2^{i-2}$. This means that the total number of vertices outside $N_0 \cup N_1 \cup N_2$ is at most $\beta(d) = \sum_{i=3}^{d} 18 \times 2^{i-2}$. Let T be this set. We consider every possible 3-colouring of $G[T]$ and every possible 3-colouring of C. For each case we obtain an instance of 2-List Colouring, which we can solve in linear time due to Theorem 3.4. As the total number of instances we need to consider is at most $3^d \times 3^{\beta(d)}$, our algorithm runs in polynomial time.

2. Let G be a $K_{1,r}^2$-free graph of diameter at most 2. We first check in $O(n^4)$ time if G is K_4-free. If not, then G is not 3-colourable. We then check in $O(n^5)$ time if G has an induced C_5. If G is C_5-free, then we use Theorem 3.9. From now on, suppose that G is K_4-free and that G contains an induced cycle C of length 5, say on vertices $x_1, \ldots, x_5$ in that order. We write $N_0 = V(C) = \{x_1, \ldots, x_5\}$, $N_1 = N(V(C))$ and $N_2 = V(G) \setminus (N_0 \cup N_1)$.

Let N_2' be the set of vertices in N_2 that are adjacent to some vertex in N_1 that is a private neighbour of some vertex in N_0 with respect to N_0. As G is K_4-free, the private neighbourhood $P(x_i)$ of each vertex $x_i \in N_0$ with respect to N_0 does not contain a clique of size 3. Moreover, if $P(x_i)$ contains an independent set I of size $r-1$ for some $i \in \{1, \ldots, 5\}$, then $I \cup \{x_i, x_{i+1}, x_{i+2}, x_{i+3}\}$ induces a $K_{1,r}^2$, which is not possible. Now let $v \in P(x_i)$ for some $i \in \{1, \ldots 5\}$, say $i = 1$. As G is K_4-free, the set $N(v) \cap N_2$ does not contain a clique of size 3. Moreover, if $N(v) \cap N_2$ contains an independent set I' of size $r-1$, then $I' \cup \{v, x_1, x_2, x_3, \}$ induces a $K_{1,r}^2$, which is not possible. Hence, $|N(v) \cap N_2| \leq R(3, r-1)$ by Ramsey's Theorem. We conclude that $|N_2'| \leq 5R(3, r-1)^2$.

We now consider all possible 3-colourings of C. Let c be such a 3-colouring. We assume without loss of generality that $c(x_1) = c(x_3) = 1$, $c(x_2) = c(x_4) = 2$ and $c(x_5) = 3$. Moreover, every vertex that has two differently coloured neighbours can only be coloured with one remaining colour. We assign this unique colour to such a vertex and apply this rule as far as possible. This takes polynomial time. The remaining vertices have a list of admissible colours that either consists of two or three colours, and vertices in the latter case must belong to N_2 (as $N(N_0) = N_1$).

Let T_c be the set of vertices in N_2 that still have a list of size 3. We will prove that $T_c \subseteq N_2'$. Let $u \in T_c$. As G has diameter 2, we find that u has a neighbour v adjacent to x_5. Then v cannot be adjacent to any of $x_1, \ldots, x_4$, as otherwise v would have a unique colour and u would not be in T_c. Hence, v is a private neighbour of x_5 with respect to N_0. We conclude that all vertices in T_c belong to N_2', which implies that $|T_c| \leq |N_2'| \leq 5R(3, r-1)^2$.

We now consider every possible 3-colouring of $G[T_c]$. Then all uncoloured vertices have a list of size at most 2. In other words, we created an instance of 2-List Colouring, which we solve in linear time by theorem3.4 . As the number of 3-colourings of C is at most 3^5 and for each 3-colouring c of C the number of 3-colourings of $G[T_c]$ is at most $3^{5R(3,r-1)^2}$, the total running time of our algorithm is polynomial.

3. We consider the standard reduction from the NP-complete problem NAE 3-SAT [87], where each variable appears in at most three clauses and each literal appears in at most two. Given a CNF formula ϕ, we construct the graph G as follows:

- Add a vertex v_{x_i} for each literal x_i.
- Add an edge between each literal and its negation.
- Add a vertex z adjacent to every literal vertex.
- For each clause C_i add a triangle T_i with vertices $c_{i_1}, c_{i_2}, c_{i_3}$.
- Fix an arbitrary order of the literals of C_i, $x_{i_1}, x_{i_2}, x_{i_3}$ and add an edge $v_{x_{i_j}} c_{i_j}$.

Given a 3-colouring of G, assume z is assigned colour 1. Then each literal vertex is assigned either colour 2 or colour 3. If, for some clause C_i, the vertices x_{i_1}, x_{i_2} and $x_{i,3}$ are all assigned the same colour, then T_i cannot be coloured. Therefore, if we set literals whose vertices are coloured with colour

2 to be true and those coloured with colour 3 to be false, each clause must contain at least one true literal and at least one false literal.

If ϕ is satisfiable then we can colour vertex z with colour 1, each true literal with colour 2 and each false literal with colour 3. Then, since each clause has at least one true literal and at least one false literal, each triangle has neighbours in two different colours. This implies that each triangle is 3-colourable. Therefore G is 3-colourable if and only if ϕ is satisfiable.

We next show that G has diameter at most 4. First note that any literal vertex is adjacent to z and any clause vertex is adjacent to some literal vertex so any vertex is at distance at most 2 from z. Therefore any two vertices are at distance at most 4.

Finally we show that G is $K_{1,4}^3$-free. Any literal vertex has degree at most 4 since it appears in at most two clauses. However it has at most 3 independent neighbours since its negation is adjacent to z. Each clause vertex has at most 3 neighbours so the only vertex with four independent neighbours is d. The longest induced path including z has length at most 4 since any such path contains at most one literal and at most two vertices of any triangle. Therefore G is $K_{1,4}^3$-free.

4. This follows from the NP-completeness of k-colouring for claw-free graphs [50]. Let $k^* \geq 3$. We take a claw-free graph G and add a dominating vertex to it. The new graph G' has diameter at most 2 and is $K_{1,3}^1$-free. Let $k = k^* + 1 \geq 4$. Then G is k^*-colourable if and only if G' is k-colourable. □

The next interesting case is the graph $S_{1,2,2}$ obtained by subdividing two edges of the claw. For $k \geq 4$, Theorem 3.10 implies that k-Colouring is NP-complete. For $k = 3$ we prove that the case $d = 2$ is polynomial-time solvable. This leaves open the cases $k = 3, d \geq 2$.

Theorem 3.11. 3-Colouring *can be solved in polynomial time for* $S_{1,2,2}$-*free graphs of diameter at most* 2.

Proof. Let G be an $S_{1,2,2}$-free graph of diameter at most 2. We first check in $O(n^5)$ time if G has an induced C_5. If G is C_5-free, then we use Theorem 3.9.

Suppose G contains an induced cycle C of length 5, say on vertices $x_1, \ldots, x_5$ in that order. We write $N_0 = V(C) = \{x_1, \ldots, x_5\}$, $N_1 = N(V(C))$ and $N_2 = V(G) \setminus (N_0 \cup N_1)$. As G has diameter 2, for every $i \in \{1, 2, 3\}$, every vertex in N_2 has a neighbour in N_1 that is adjacent to x_i.

We let T consist of all vertices of N_2 that have a neighbour in N_1 that is adjacent to two adjacent vertices of N_0. So the colour of any vertex of T will be fixed in any 3-colouring after colouring the five vertices of N_0. We claim that $N_2 = T$. In order to see this, let $u \in N_2$. As G has diameter 2, we find that u must have a neighbour $v \in N_1$ adjacent to a vertex of N_0, say x_1. Then v is not adjacent to x_5 or x_2. If v is not adjacent to x_3 either, then the vertices x_1, x_5, x_2, x_3, v, u induce a $S_{1,2,2}$ with center x_1, a contradiction. So v must be adjacent to x_3, meaning v is not adjacent to x_4. However, now x_3, x_2, x_4, x_5, v, u induce a $S_{1,2,2}$ with center x_3, another contradiction.

We now "guess" the 3-colouring of C by considering all 3^5 possibilities if necessary. We then proceed as in the proof of Theorem 3.9. That is, we observe that every vertex of N_1 can only be coloured with two possible colours and that after propagation, every uncoloured vertex of N_2 can only be coloured with two possible colours as well (as $T = N_2$). Then it remains to solve an instance of 2-List Colouring, which takes linear time by Theorem 3.4. As we need to do this at most 3^5 times, the total running time of our algorithm is polynomial. □

3.5 Graphs Avoiding Short Cycles

In the previous sections our results for 3-colouring graphs of diameter 2 focussed largely on graph classes characterised by some forbidden induced subdivided star. However, we also obtained results for C_5-free graphs and for graphs of girth at least 5, ((C_3, C_4)-free graphs). Here we continue our study for (C_s, C_t)-free graphs of diameter 2 where s and t are small. In particular, we focus on the case $s = 4$. The results in this section are proved for the more general problem List 3-colouring whose complexity also remains open for graphs of diameter 2.

3.5.1 The Propagation Algorithm and Three Results

We now present our propagation algorithm, based on a number of (well-known) rules, before using it to prove our first three polynomial-time results. The purpose of the propagation algorithm is to minimise the number of colours in the list of available colours for each vertex by exhaustively applying these rules to a given instance of List 3-Colouring.

Rule 1. **(no empty lists)** If $L(u) = \emptyset$ for some $u \in V$, then return `no`.

Rule 2. **(not only lists of size 2)** If $|L(u)| \leq 2$ for every $u \in V$, then apply Theorem 3.4.

Rule 3. **(single colour propagation)** If u and v are adjacent, $|L(u)| = 1$, and $L(u) \subseteq L(v)$, then set $L(v) := L(v) \setminus L(u)$.

Rule 4. **(diamond colour propagation)** If u and v are adjacent and share two common non-adjacent neighbours x and y such that $|L(x)| = |L(y)| = 2$ and $L(x) \neq L(y)$, then set $L(x) := L(x) \cap L(y)$ and $L(y) := L(x) \cap L(y)$ (so $L(x)$ and $L(y)$ get size 1). See figure 3.3.

Rule 5. **(bull colour propagation)** If u and v are the two vertices of an induced bull B of G having degree 1 and $L(u) = L(v) = \{i\}$ for some $i \in \{1, 2, 3\}$ and moreover $L(w) \neq \{i\}$ for the degree-2 vertex w of B, then set $L(w) := L(w) \cap \{i\}$. Displayed in figure 3.4.

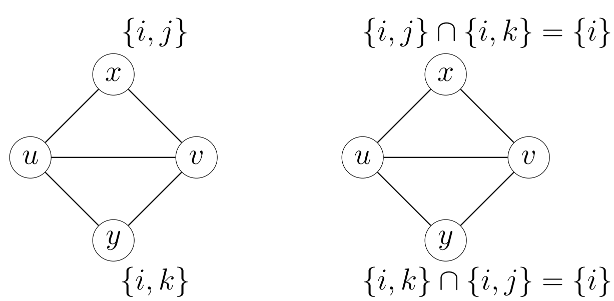

Figure 3.3: Left: A diamond graph before applying Rule 4. Right: After applying Rule 4.

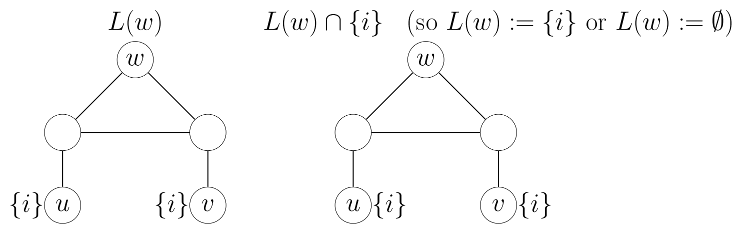

Figure 3.4: Left: A bull graph before applying Rule 5. Right: After applying Rule 5.

We say that a propagation rule is *safe* if the new instance is a yes-instance of List 3-Colouring if and only if the original instance is so. We make the following observation, which is straightforward (see also [60]).

Lemma 3.3. *Each of the Rules 1–5 is safe and can be applied in polynomial time.*

Consider again an instance (G, L). Let N_0 be a subset of $V(G)$ that has size at most some constant. Assume that $G[N_0]$ has a colouring c that respects the restriction of L to N_0. We say that c is an *L-promising N_0-precolouring* of G.

In our algorithms we first determine a set N_0 of constant size and consider every L-promising N_0-precolouring of G. That is, we modify L into a list assignment L_c with $L_c(u) = \{c(u)\}$ (where $c(u) \in L(u)$) for every $u \in N_0$ and $L_c(u) = L(u)$ for every $u \in V(G) \setminus N_0$). We then apply Rules 1–5 on (G, L_c) *exhaustively*, that is, until none of the rules can be applied anymore. This is the *propagation algorithm* and we say that it did a *full c-propagation*. The propagation algorithm may output `yes` and `no` (when applying Rules 1 or 2); else it will output `unknown`.

If the algorithm returns `yes`, then (G, L) is a yes-instance of List 3-Colouring by Lemma 3.3. If it returns `no`, then (G, L) has no L-respecting colouring coinciding with c on N_0, again by Lemma 3.3. If the algorithm returns `unknown`, then (G, L) may still have an L-respecting colouring that coincides with c on N_0. In that case the propagation algorithm did not apply Rule 1 or 2. Hence, it modified L_c into a list assignment L'_c of G such that $L'_c(u) \neq \emptyset$ for every $u \in V(G)$ and at least one vertex v of G still has a list $L'_c(v)$ of size 3, that is, $L'_c(v) = \{1, 2, 3\}$. We say that L'_c (if it exists) is the *c-propagated list assignment* of G.

After performing a full c-propagation for every N_0-precolouring c of G which is L-promising we say that we performed a *full N_0-propagation.* We say that (G, L) is *N_0-terminal* if after the full N_0-propagation one of the following cases hold:

1. for some L-promising N_0-precolouring, the propagation algorithm returned `yes`;
2. for every L-promising N_0-precolouring, the propagation algorithm returned `no`.

Note that if (G, L) is N_0-terminal for some set N_0, then we have solved List 3-Colouring on instance (G, L). The next lemma formalizes our approach.

Lemma 3.4. *Let (G, L) be an instance of* List 3-Colouring. *Let N_0 be a subset of $V(G)$ of constant size. Performing a full N_0-propagation takes polynomial time. Moreover, if (G, L) is N_0-terminal, then we have solved* List 3-Colouring *on instance (G, L).*

Proof. The first part of the lemma follows from the facts that (i) each application of each rule is safe and takes polynomial time by Lemma 3.3; (ii) if a rule does not return `yes` or `no`, then it reduces the list size of at least one vertex and the latter can happen at most $3|V|$ times; and (iii) the number of L-promising N_0-precolourings of G is at most $3^{|N_0|}$, which is a constant as N_0 has constant size. The second part of the lemma follows from the definition of a full N_0-propagation and Lemma 3.3. $\square$

We now prove our first three results on List 3-Colouring for diameter-2 graphs. The first result generalizes Theorem 3.9.

Theorem 3.12. List 3-Colouring *can be solved in polynomial time for C_5-free graphs of diameter at most* 2.

Proof. Let $G = (V, E)$ be a C_5-free graph of diameter 2 with a list 3-assignment L. We first check in polynomial time if G is bipartite. Suppose that we find that G is bipartite, say with partition classes A and B. As G

has diameter 2, we find that G must be complete bipartite. This implies that either A or B must be monochromatic. For each $i \in \bigcap_{u \in A} L(u)$ (which might be empty) we set $L(u) = \{i\}$ for every $u \in A$ and $L(v) := L(v) \setminus \{i\}$ for every $i \in B$ and solve the resulting instance of 2-List Colouring. If we do not find a colouring respecting L, then we reverse the role of A and B and perform the same step.

Now suppose that we find that G is not bipartite. If G contains a K_4, then G is not 3-colourable, and hence (G, L) is a no-instance of List 3-Colouring. We can check this in $O(|V|^4)$ time. From now on we assume that G is K_4-free and non-bipartite. The latter implies that G must have a triangle or an induced C_5, due to Lemma 3.2. As G is C_5-free, it follows that G has at least one triangle.

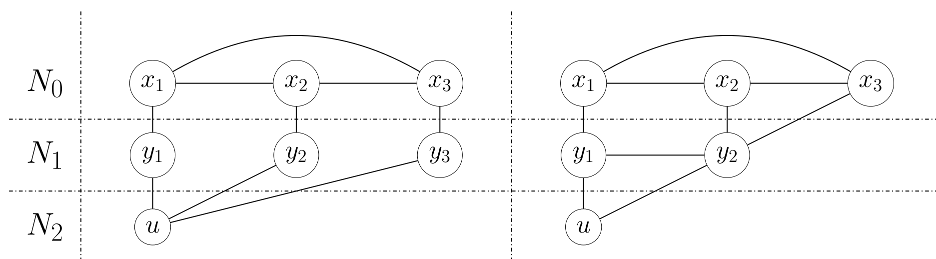

Figure 3.5: Left: Examining the situation in the proof of Theorem 3.12 where a vertex $u \in N_2$ does not belong to T; we show that y_1, y_2, y_3 and u either form a K_4 or we would find an induced C_5 (both of these cases are not possible). Right: A situation where $u \in T$.

Let C be a triangle in G. We write $N_0 = V(C) = \{x_1, x_2, x_3\}$, $N_1 = N(V(C))$ and $N_2 = V(G) \setminus (N_0 \cup N_1)$. As N_0 has size 3, we can apply a full N_0-propagation in polynomial time by Lemma 3.4. By the same lemma we are done if we can prove that (G, L) is N_0-terminal. We prove this claim below after first showing a structural result.

As G has diameter 2, for every $i \in \{1, 2, 3\}$, it holds that every vertex in N_2 has a neighbour in N_1 that is adjacent to x_i. Now let T consist of all vertices of N_2 that have a neighbour in N_1 that is adjacent to exactly two vertices of N_0.

Claim 1. $N_2 = T$.

We prove Claim 1 as follows. Let $u \in N_2$. For contradiction, assume $u \notin T$. If u has a neighbour $y \in N_1$ adjacent to every x_i, then G contains a K_4, a contradiction. Hence, as $u \notin T$, we find that u must have three distinct neighbours y_1, y_2, y_3, such that for $i \in \{1, 2, 3\}$, it holds that $N(y_i) \cap N_0 = \{x_i\}$. If $\{y_1, y_2, y_3\}$ is a clique, then G has a K_4 on vertices u, y_1, y_2, y_3, a contradiction. Hence, we may assume without loss of generality that y_1 and y_2 are non-adjacent. However, then $\{u, y_1, x_1, x_2, y_2\}$ induces a C_5 in G, another contradiction. See also Figure 3.5. We conclude that $T = N_2$. This proves Claim 1.

Now, for contradiction, assume that (G, L) is not N_0-terminal. Then there must exist an L-promising N_0-precolouring c for which we obtain the c-propagated list assignment L'_c. By definition of L'_c we find that G contains a vertex u with $L'_c(u) = \{1, 2, 3\}$. Then $u \notin N_0$, as every $v \in N_0$ has $L'_c(v) = \{c(v)\}$. Moreover, $u \notin N_1$, as vertices in N_1 have a list of size at most 2 after applying Rule 3. Hence, we find that $u \in N_2$. As $N_2 = T$ by Claim 1, we find that $u \in T$. From the definition of T it follows that u has a neighbour $v \in N_1$ with two neighbours in N_0. By Rule 3, we find that $|L_c(v)| = 1$. By the same rule, this implies that $|L'_c(u)| \leq 2$, a contradiction. We conclude that (G, L) is N_0-terminal. $\square$

Theorem 3.13. List 3-Colouring *can be solved in polynomial time for* C_6*-free graphs of diameter at most* 2.

Proof. Let $G = (V, E)$ be a C_6-free graph of diameter 2 with a list 3-assignment L. If G is C_5-free, then we apply Theorem 3.12. If G contains a K_4, then G is not 3-colourable and hence, (G, L) is a no-instance of List 3-Colouring. We check these properties in polynomial time. So, from now

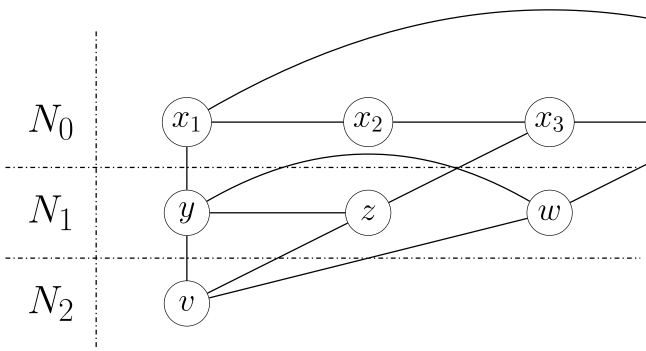

Figure 3.6: The situation in the proof of Theorem 3.13, which is similar to the situation in the proof of Theorem 3.14.

on, we assume that G is a K_4-free graph that contains an induced 5-vertex cycle C, say with vertex set $N_0 = \{x_1, \ldots, x_5\}$ in this order. Let N_1 be the set of vertices that do not belong to C but that are adjacent to at least one vertex of C. Let $N_2 = V \setminus (N_0 \cup N_1)$ be the set of remaining vertices.

As N_0 has size 5, we can apply a full N_0-propagation in polynomial time by Lemma 3.4. By the same lemma we are done if we can prove that (G, L) is N_0-terminal. We prove this claim below.

For contradiction, assume that (G, L) is not N_0-terminal. Then there must exist an L-promising N_0-precolouring c for which we obtain the c-propagated list assignment L'_c. By definition of L'_c we find that G contains a vertex v with $L'_c(v) = \{1, 2, 3\}$. Then $v \notin N_0$, as every $u \in N_0$ has $L'_c(u) = \{c(u)\}$. Moreover, $v \notin N_1$, as vertices in N_1 have a list of size at most 2 after applying Rule 3. Hence, we find that $v \in N_2$.

We first note that some colour of $\{1, 2, 3\}$ appears exactly once on N_0, as $|N_0| = 5$. Hence, we may assume without loss of generality that $c(x_1) = 1$ and that $c(x_i) \in \{2, 3\}$ for every $i \in \{2, 3, 4, 5\}$.

As G has diameter 2, there exists a vertex $y \in N_1$ that is adjacent to x_1 and v. As $L'_c(v) = \{1, 2, 3\}$ and $c(x_1) = 1$, we find that $L'_c(y) = \{2, 3\}$. As $c(x_i) \in \{2, 3\}$ for every $i \in \{2, 3, 4, 5\}$, the latter means that y is not adjacent to any x_i with $i \in \{2,3,4,5\}$. Hence, as G has diameter 2, there exists a vertex $z \in N_1$ with $z \neq y$, such that z is adjacent to x_3 and v. We assume without loss of generality that $c(x_3) = 3$ and thus $c(x_2) = c(x_4) = 2$ and thus $c(x_5) = 3$. As $L'_c(v) = \{1,2,3\}$ and $c(x_3) = 3$, we find that $L'_c(z) = \{1,2\}$. Hence, z is not adjacent to any vertex of $\{x_1, x_2, x_4\}$. Now the set $\{x_1, x_2, x_3, z, v, y\}$ forms a cycle on six vertices. As G is C_6-free, this cycle cannot be induced. Hence, the above implies that y and z must be adjacent; see also Figure 3.6.

As G has diameter 2, there exists a vertex $w \in N_1$ that is adjacent to x_4 and v. As both y and z are not adjacent to x_4, we find that $w \notin \{y, z\}$. As $L'_c(v) = \{1,2,3\}$ and $c(x_4) = 2$, we find that $L'_c(w) = \{1,3\}$. As $c(x_1) = 1$ and $c(x_3) = c(x_5) = 3$, the latter implies that w is not adjacent to any vertex of $\{x_1, x_3, x_5\}$. Consequently, w must be adjacent to y, as otherwise the 6-vertex cycle with vertex set $\{x_1, x_5, x_4, w, v, y\}$ would be induced, contradicting the C_6-freeness of G. We refer again to Figure 3.6 for a display of the situation.

If w and z are adjacent, then $\{v, w, y, z\}$ induces a K_4, contradicting the K_4-freeness of G. Hence, w and z are not adjacent. Then $\{v, w, y, z\}$ induces a diamond, in which w and z are the two non-adjacent vertices. However, as $L'_c(w) = \{1,3\}$ and $L'_c(z) = \{1,2\}$, our algorithm would have applied Rule 4. This would have resulted in lists of w and z that are both equal to $\{1,3\} \cap \{1,2\} = \{1\}$. Hence, we obtained a contradiction and conclude that (G, L) is N_0-terminal. $\square$

Theorem 3.14 is proven in a similar way to Theorem 3.13.

Theorem 3.14. List 3-Colouring *can be solved in polynomial time for* (C_4, C_7)*-free graphs of diameter* 2.

Proof. Let $G = (V, E)$ be a C_4-free graph of diameter 2 with a list 3-assignment L. If G is C_5-free, then we apply Theorem 3.12. Hence we may assume that G contains an induced 5-vertex cycle C, say with vertex set $N_0 = \{x_1, \ldots, x_5\}$ in this order. As before, we let N_1 be the set of vertices that do not belong to C but that are adjacent to at least one vertex of C. We also let $N_2 = V \setminus (N_0 \cup N_1)$ denote the set of remaining vertices again.

As N_0 has size 5, we can apply a full N_0-propagation in polynomial time by Lemma 3.4. By the same lemma we are done if we can prove that (G, L) is N_0-terminal. We prove this claim in exactly the same way in which we proved a similar claim in the proof of Theorem 3.13 except for the following differences:

1. instead of using the 6-vertex set $\{x_1, x_2, x_3, z, v, y\}$ we use the 7-vertex set $\{x_1, x_5, x_4, x_3, z, v, y\}$ after observing that z cannot be adjacent to x_5 due to the C_4-freeness of G, and

2. instead of using the 6-vertex set $\{x_1, x_5, x_4, w, v, y\}$ we use the 7-vertex set $\{x_1, x_2, x_3, x_4, w, v, y\}$ after observing that w cannot be adjacent to x_2, again due to the C_4-freeness of G.

We refer again to Figure 3.6 for a display of the situation. □

3.5.2 The Extended Propagation Algorithm and Two Results

For our next two results, we need a more sophisticated method. Let (G, L) be an instance of List 3-Colouring. Let p be some positive constant. We consider each set $N_0 \subseteq V(G)$ of size at most p and perform a full N_0-propagation. Afterwards we say that we performed a *full p-propagation*. We say that (G, L) is *p-terminal* if after the full p-propagation one of the following cases hold:

1. for some $N_0 \subseteq V(G)$ with $|N_0| \leq c$, there is an L-promising N_0-precolouring c, such that the propagation algorithm returns `yes`; or

2. for every set $N_0 \subseteq V(G)$ with $|N_0| \leq c$ and every L-promising N_0-precolouring c, the propagation algorithm returns `no`.

We can now prove the following lemma.

Lemma 3.5. *Let* (G, L) *be an instance of* List 3-Colouring *and* $p \geq 1$ *be some constant. Performing a full p-propagation takes polynomial time. Moreover, if* (G, L) *is p-terminal, then we have solved* List 3-Colouring *on instance* (G, L).

Proof. For every set $N_0 \subseteq V(G)$, a full N_0-propagation takes polynomial time by Lemma 3.4. Then the first statement of the lemma follows from this observation and the fact that we need to perform $O(n^p)$ full N_0-propagations, which is a polynomial number, as p is a constant.

Now suppose that (G, L) is p-terminal. First assume that for some $N_0 \subseteq V(G)$ with $|N_0| \leq c$, there exists an L-promising N_0-precolouring c, such that the propagation algorithm returns `yes`. Then (G, L) is a yes-instance due to Lemma 3.3. Now assume that for every set $N_0 \subseteq V(G)$ with $|N_0| \leq c$ and every L-promising N_0-precolouring c, the propagation algorithm returns `no`. Then (G, L) is a no-instance. This follows from Lemma 3.3 combined with the observation that if (G, L) was a yes-instance, the restriction of a colouring c that respects L to any set N_0 of size at most p would be an L-promising N_0-precolouring of G. □

In our next two algorithms, we perform a full p-propagation for some appropriate constant p. If we find that an instance (G, L) is p-terminal, then we are done by Lemma 3.5. In the other case, we exploit the new information on the structure of G that we obtain from the fact that (G, L) is not p-terminal.

Theorem 3.15. List 3-Colouring *can be solved in polynomial time for* (C_4, C_8)*-free graphs of diameter* 2.

Proof. Let $G = (V, E)$ be a (C_4, C_8)-free graph of diameter 2 with a list 3-assignment L. If G is C_6-free, then we apply Theorem 3.13. If G contains a K_4, then G is not 3-colourable and hence, (G, L) is a no-instance of List 3-Colouring. We check these properties in polynomial time. So, from now on, we assume that G is a K_4-free graph that contains at least one induced cycle on six vertices.

We set $p = 6$ and perform a full p-propagation. This takes polynomial time by Lemma 3.5. By the same lemma, we have solved List 3-Colouring on (G, L) if (G, L) is p-terminal. Suppose we find that (G, L) is not p-terminal.

We first prove the following claim.

Claim 1. For each induced 6-vertex cycle C, the propagation algorithm returned `no` *for every $V(C)$-promising colouring c that assigns the same colour i on two vertices of C that have a common neighbour on C.*

We prove Claim 1 as follows. Consider an induced 6-vertex cycle C, say with vertex set $N_0 = \{x_1, \ldots, x_6\}$ in this order. Let N_1 be the set of vertices that do not belong to C but that are adjacent to at least one vertex of C. Let $N_2 = V \setminus (N_0 \cup N_1)$ be the set of remaining vertices. For contradiction, let c be a $V(C)$-promising colouring that assigns two vertices of C with a common neighbour on C the same colour, say $c(x_1) = 1$ and $c(x_3) = 1$, such that a full c-propagation does not yield a `no` output. As (G, L) is not p-terminal, this means that we obtained the c-propagated list assignment L'_c. By definition of L'_c we find that G contains a vertex v with $L'_c(v) = \{1, 2, 3\}$. Then $v \notin N_0$, as every $u \in N_0$ has $L'_c(u) = \{c(u)\}$. Moreover, $v \notin N_1$, as vertices in N_1 have a list of size at most 2 after applying Rule 3. Hence, we find that $v \in N_2$.

As G has diameter 2, there exist a vertex $y \in N_1$ that is adjacent to both v and x_1. As $c(x_1) = 1$, we find that $c(x_2) \in \{2, 3\}$ and $c(x_6) \in \{2, 3\}$. As $c(x_3) = 1$, we find that $c(x_4) \in \{2, 3\}$. Hence, y is not adjacent to any vertex of $\{x_2, x_4, x_6\}$; otherwise y would have a list of size 1 due to Rule 3, and by the same rule, v would have a list of size 2. We note that y is not adjacent to x_3 or x_5 either, as otherwise $\{x_1, x_2, x_3, y\}$ or $\{x_1, x_6, x_5, y\}$ induces a C_4, contradicting the C_4-freeness of G.

As G has diameter 2 and $yx_3 \notin E$, there exists a vertex $y' \in N_1 \setminus \{y\}$ that is adjacent to both v and x_3. By the same arguments as above, y' is not adjacent to any vertex of $\{x_1, x_2, x_4, x_5, x_6\}$. If y and y' are adjacent, then v would have list $\{1\}$ due to Rule 5. Hence y and y' are not adjacent. However,

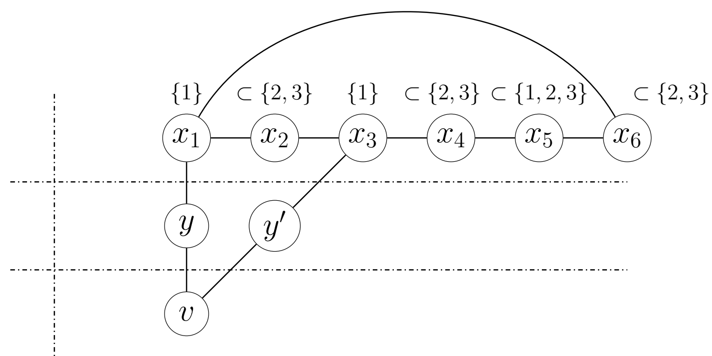

Figure 3.7: The situation that is described in Claim 1 in the proof of Theorem 3.15: the set $\{x_1, y, v, y', x_3, x_4, x_5, x_6\}$ induces a C_8, which is not possible.

we now find that $\{x_1, y, v, y', x_3, x_4, x_5, x_6\}$ induces a C_8, contradicting the C_8-freeness of G; see also Figure 3.7. This proves Claim 1.

Due to Claim 1, we know that if G has a colouring c respecting L, then any such colouring c gives a different colour to every two non-adjacent vertices that are of distance 2 on some induced 6-vertex cycle. Hence, we can safely use the following new rule. To explain this, x_5 cannot get the same colour as either x_1 or x_3, which are both of distance 2 from x_5 on an induced C_6, thus x_5 must get the remaining colour, which is the colour of x_2. Moreover, an application of the new rule takes polynomial time. Note that we must also have that $L(x_4) = L(x_1)$ and $L(x_6) = L(x_3)$ but this will be irrelevant for our purposes.

Rule 6. ($\mathbf{C_6}$ colour propagation) Let C be an induced six vertex cycle $x_1, x_2, \ldots, x_6$ in that order. If $|L(x_1)| = |L(x_2)| = |L(x_3)| = 1$, $L(\{x_1, x_2, x_3\}) = \{1, 2, 3\}$ and $L(x_2) \neq L(x_5)$, then set $L(x_5) := L(x_2) \cap L(x_5)$ (so x_5 gets a list of size at most 1).

We can now do as follows. Consider an induced 6-vertex cycle C in G,

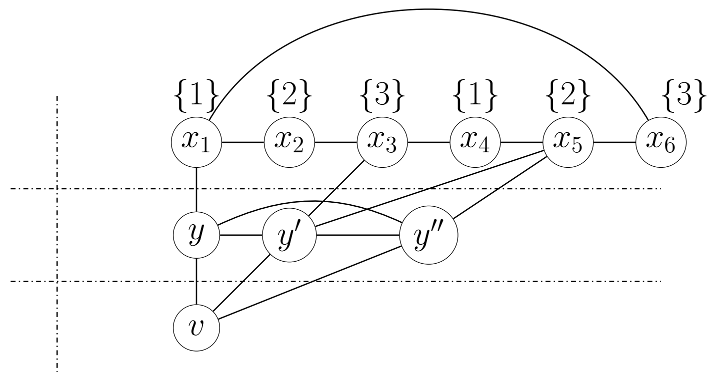

Figure 3.8: The situation in the proof of Theorem 3.15, where a vertex $v \in N_2$ still has a list of three available colours after a full propagation including Rule 6: we show that in this case G contains a K_4, namely on vertices v, y, y', y'', a contradiction.

say on vertices $x_1, \ldots, x_6$ in that order. Then we may assume without loss of generality that if G has a colouring c that respects L, then $c(x_1) = 1$, $c(x_2) = 2$, $c(x_3) = 3$, $c(x_4) = 1$, $c(x_5) = 2$ and $c(x_6) = 3$ (otherwise we can do some permutation of the colours). See also Figure 3.8.

We let again $N_0 = \{x_1, \ldots, x_6\}$, N_1 be the set of vertices that do not belong to C but that are adjacent to at least one vertex of C, and $N_2 = V \setminus (N_0 \cup N_1)$ be the set of remaining vertices. We define a colouring c of $G[N_0]$ by setting $c(x_1) = 1$, $c(x_2) = 2$, $c(x_3) = 3$, $c(x_4) = 1$, $c(x_5) = 2$ and $c(x_6) = 3$. We do a full c-propagation but now we also include the exhaustive use of Rule 6. By combining Lemma 3.5 with the observation that Rule 6 runs in polynomial time and reduces the list size of at least one vertex, this takes polynomial time. By combining the same lemma with the fact that Rule 6 is safe (due to Claim 1) and the above observation that every L-respecting colouring of G coincides with c on N_0 (subject to colour permutation), we are done if we can prove that the propagation algorithm either outputs `yes` or `no`.

For contradiction, assume that the propagation algorithm returns `unknown`. Then we obtained the c-propagated list assignment L'_c. By definition of L'_c we find that G contains a a vertex v with $L'_c(v) = \{1, 2, 3\}$. Then $v \notin N_0$, as every $u \in N_0$ has $L'_c(u) = \{c(u)\}$. Moreover, $v \notin N_1$, as vertices in N_1 have a list of size at most 2 after applying Rule 3. Hence, we find that $v \in N_2$.

As G has diameter 2, there exists a vertex $y \in N_1$ that is adjacent to x_1 and v. Hence, y is not adjacent to any vertex in $\{x_2, x_3, x_5, x_6\}$; otherwise y would have a list of size 1 due to Rule 3, and by the same rule, v would have a list of size 2. As G has diameter 2 and $yx_3 \notin E$, there exists a vertex $y' \in N_1 \setminus \{y\}$ that is adjacent to x_3 and v. By the same arguments as above, y' is not adjacent to any vertex in $\{x_1, x_2, x_4, x_5\}$. If $yy' \notin E$, then $\{x_1, x_2, x_3, y', v, y\}$ induces a C_6. However, in that case we would have applied Rule 6 and v would have had list $\{2\}$. Hence, we find that y and y' are adjacent; see also Figure 3.8.

As G has diameter 2, $yx_5 \notin E$ and $y'x_5 \notin E$, there exists a vertex $y'' \in N_1 \setminus \{y, y'\}$ that is adjacent to x_5 and v. By using exactly the same arguments as above but now applied to y'' and to the pairs (y, y'') and (y', y''), respectively, we find that y'' is adjacent to both y and y'. However, now the vertices v, y, y', y'' induce a K_4, contradicting the K_4-freeness of G (see again Figure 3.8). We conclude that the propagation algorithm returned either `yes` or `no`. □

Theorem 3.16. List 3-Colouring *can be solved in polynomial time for* (C_4, C_9)*-free graphs of diameter* 2.

Proof. Let $G = (V, E)$ be a (C_4, C_9)-free graph of diameter 2 with a list 3-assignment L. If G is C_7-free, then we apply Theorem 3.14. If G contains a K_4, then G is not 3-colourable and hence, (G, L) is a no-instance of List 3-Colouring. We check these properties in polynomial time. So, from now on, we assume that G is a K_4-free graph that contains at least one induced cycle on seven vertices.

We set $p = 7$ and perform a full p-propagation. This takes polynomial time by Lemma 3.5. By the same lemma, we have solved List 3-Colouring on (G, L) if (G, L) is p-terminal. Suppose we find that (G, L) is not p-terminal.

We first prove the following claim.

Claim 1. For each induced 7-vertex cycle C, the propagation algorithm returned `no` *for every L-promising $V(C)$-colouring c that assigns the same colour i on two vertices of C that have a common neighbour on C and that gives every other vertex of C a colour different from i.*

We prove Claim 1 as follows. Consider an induced 7-vertex cycle C, say with vertex set $N_0 = \{x_1, \ldots, x_7\}$ in this order. Let N_1 be the set of vertices that do not belong to C but that are adjacent to at least one vertex of C. Let $N_2 = V \setminus (N_0 \cup N_1)$ be the set of remaining vertices. Let c be an L-promising $V(C)$-colouring that assigns two vertices of C with a common neighbour on C the same colour, say $c(x_1) = 1$ and $c(x_3) = 1$, and moreover, that assigns every vertex x_i with $i \in \{2, 4, 5, 6, 7\}$ colour $c(x_i) \neq 1$.

For contradiction, suppose that a full c-propagation does not yield a `no` output. As (G, L) is not p-terminal, this means that we obtained the c-propagated list assignment L'_c. By definition of L'_c we find that G contains a vertex v with $L'_c(v) = \{1, 2, 3\}$. Then $v \notin N_0$, as every $u \in N_0$ has $L'_c(u) = \{c(u)\}$. Moreover, $v \notin N_1$, as vertices in N_1 have a list of size at most 2 after applying Rule 3. Hence, we find that $v \in N_2$.

As G has diameter 2, there exist a vertex $y \in N_1$ that is adjacent to both v and x_1. Then y is not adjacent to any x_i with $i \in \{2, 4, 5, 6, 7\}$; in that case y would have a list of size 1 (as each x_i other than x_1 and x_3 is coloured 2 or 3) meaning that $L'_c(v)$ would have size at most 2. Hence, y is not adjacent to x_3 either, as otherwise $\{y, x_1, x_2, x_3\}$ would induce a C_4. As G has diameter 2, this means that there exists a vertex $y' \in N_1$ with $y' \neq y$ such that y' is adjacent to both v and x_3. By the same arguments we used for y', we find that x_3 is the only neighbour of y' on C.

If yy' is an edge then, by Rule 5, v would have had list $\{1\}$ instead of $\{1,2,3\}$. Hence, y and y' are not adjacent. However, in this case the vertices $\{y, v, y', x_3, x_4, x_5, x_6, x_7, x_1\}$ induces a C_9, a contradiction; see also Figure 3.9. This proves Claim 1.

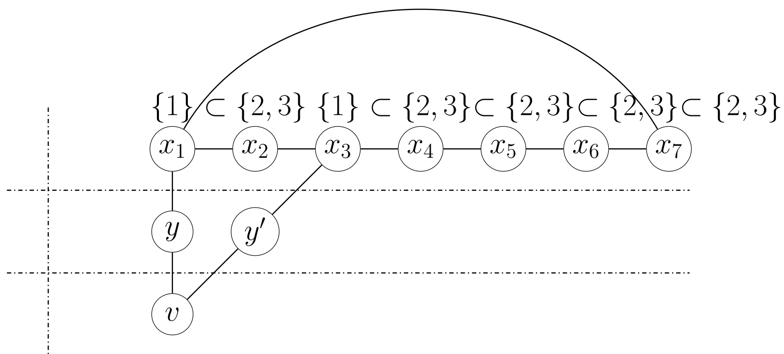

Figure 3.9: The situation that is described in Claim 1 in the proof of Theorem 3.16. The set $\{x_1, y, v, y', x_3, x_4, x_5, x_6, x_7\}$ induces a C_9, which is not possible.

Claim 1 tells us that if G has a colouring c respecting L, then c only gives the same colour to two vertices x and x' that are of distance 2 on some induced 7-vertex cycle C if there is a third vertex x'' that is of distance 2 from either x or x' on C with $c(x'') = c(x') = c(x)$. Hence, we can safely use the following new rule, whose execution takes polynomial time (in this rule, $c(x_1) = c(x_6)$ is not possible: view x_1 as x and x_6 as x' and note that x'' can neither be x_3 or x_4).

Rule 7. ($\mathbf{C_7}$ colour propagation) Let C be an induced seven vertex cycle $x_1, x_2, \ldots, x_7$ in that order. If $|L(x_i)| = 1$ for $i \in \{1,2,3,4\}$, $L(\{x_1, x_2, x_3\}) = \{1,2,3\}$, $L(x_4) = L(x_2)$, and $L(x_1) \subseteq L(x_6)$, then set $L(x_6) := \{1,2,3\} \setminus L(x_1)$ (so $L(x_6)$ gets size at most 2).

We now consider an induced 7-vertex cycle C in G, say on vertices $x_1, \ldots, x_7$ in that order. Then either one colour appear once on C, or two colours appear exactly twice on C, with distance 3 from each other on C. Hence, we may assume without loss of generality that if G has a colouring c that respects L, then one of the following holds for such a colouring c (see also Figures 3.10 and 3.11):

(1) $c(x_1) = 1$, $c(x_2) = 2$, $c(x_3) = 3$, $c(x_4) = 2$, $c(x_5) = 3$, $c(x_6) = 2$, $c(x_7) = 3$; or

(2) $c(x_1) = 1$, $c(x_2) = 2$, $c(x_3) = 3$, $c(x_4) = 1$, $c(x_5) = 3$, $c(x_6) = 2$, $c(x_7) = 3$.

We let again $N_0 = \{x_1, \ldots, x_7\}$, N_1 be the set of vertices that do not belong to C but that are adjacent to at least one vertex of C, and $N_2 = V \setminus (N_0 \cup N_1)$ be the set of remaining vertices. We do a full c-propagation but now we also include the exhaustive use of Rule 7. By combining Lemma 3.5 with the observation that Rule 7 runs in polynomial time and reduces the list size of at least one vertex, this takes polynomial time. By combining the same lemma with the fact that Rule 7 is safe (due to Claim 1) and the above observation that every L-respecting colouring of G coincides with c on N_0 (subject to colour permutation), we are done if we can prove that the propagation algorithm either outputs `yes` or `no`. We show that this is the case for each of the two possibilities (1) and (2) of c.

For contradiction, assume that the propagation algorithm returns `unknown`. Then we obtained the c-propagated list assignment L'_c. By definition of L'_c we find that G contains a vertex v with $L'_c(v) = \{1, 2, 3\}$. Then $v \notin N_0$, as every $u \in N_0$ has $L'_c(u) = \{c(u)\}$. Moreover, $v \notin N_1$, as vertices in N_1 have a list of size at most 2 after applying Rule 3. Hence, we find that $v \in N_2$. We now need to distinguish between the two possibilities of c.

Case 1 $c(x_1) = 1$, $c(x_2) = 2$, $c(x_3) = 3$, $c(x_4) = 2$, $c(x_5) = 3$, $c(x_6) = 2$, $c(x_7) = 3$

As G has diameter 2, there exists a vertex $y \in N_1$ that is adjacent to x_1 and v. Hence, y is not adjacent to any vertex in $\{x_2, \ldots, x_7\}$; otherwise y would have a list of size 1 due to Rule 3, and by the same rule, v would have a list of size 2. As G has diameter 2, there exists a vertex $y' \in N_1$ that is adjacent to x_4 and v. By the same arguments as above, y' is not adjacent to any vertex of $\{x_1, x_3, x_5, x_7\}$. The latter, together with the C_4-freeness of G, implies that y' is not adjacent to x_2 and x_6 either.

First suppose that $yy' \in E$. Then $\{x_1, x_7, x_6, x_5, x_4, y', y\}$ induces a C_7; see also Figure 3.10. As $c(x_1) = 1$, $c(x_7) = 3$, $c(x_6) = 2$ and $c(x_5) = 3$, we find that $L_c(\{x_1, x_7, x_6\}) = \{1, 2, 3\}$ and $L_c(x_5) = L_c(x_7)$. Then $1 \notin L_c(y')$, as otherwise the propagation algorithm would have applied Rule 7. Moreover, $2 \notin L_c(y')$, as otherwise the propagation algorithm would have applied Rule 3. Hence, $L_c(y') = \{3\}$. However, then $|L_c(v)| \leq 2$, again due to Rule 3, a contradiction.

Now suppose that $yy' \notin E$. Then $\{x_1, x_2, x_3, x_4, y', v, y\}$ induces a C_7. As $c(x_1) = 1$, $c(x_2) = 2$, $c(x_3) = 3$, $c(x_4) = 2$, we find that $L_c(\{x_1, x_2, x_3\}) = \{1, 2, 3\}$ and $L_c(x_4) = L_c(x_2)$. Then $1 \notin L_c(v)$ due to Rule 7. This is a contradiction, as we assumed $L_c(v) = \{1, 2, 3\}$. We conclude that the propagation algorithm returned either `yes` or `no`.

Case 2 $c(x_1) = 1$, $c(x_2) = 2$, $c(x_3) = 3$, $c(x_4) = 1$, $c(x_5) = 3$, $c(x_6) = 2$, $c(x_7) = 3$

As G has diameter 2, there is a vertex $y \in N_1$ adjacent to x_3 and v. Hence, y is not adjacent to any vertex in $\{x_1, x_2, x_4, x_6\}$; otherwise y would have a list of size 1 due to Rule 3, and by the same rule, v would have a list of size 2. As $yx_4 \notin E$, we find that $yx_5 \notin E$ either; otherwise $\{y, x_3, x_4, x_5\}$ induces a C_4. As G has diameter 2, this means there is a vertex $y' \in N_1 \setminus \{y\}$ adjacent to x_5 and v. By the same arguments as above, y' is not adjacent to any vertex of $\{x_1, x_2, x_4, x_6\}$. As G is C_4-free, the latter implies that $y'x_3 \notin E$ and $y'x_7 \notin E$.

If $yy' \in E$, then v would have a list of size at most 2 due to Rule 5. Hence

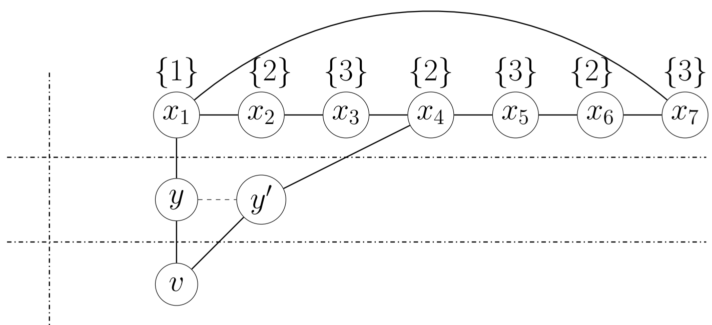

Figure 3.10: The situation that is described in Case 1 in the proof of Theorem 3.16. If the edge yy' exists, then $\{x_1, x_7, x_6, x_5, x_4, y', y\}$ induces a C_7 to which Rule 7 should have been applied. Otherwise the vertices $\{x_1, x_2, x_3, x_4, y', v, y\}$ induce such a C_7.

$yy' \notin E$. If $yx_7 \notin E$, this means that $\{x_1, x_2, x_3, y, v, y', x_5, x_6, x_7\}$ induces a C_9, which is not possible. Hence, $yx_7 \in E$.

To summarize, we found that v has two distinct neighbours y and y', where y has exactly two neighbours on C, namely x_3 and x_7, and y' has exactly one neighbour on C, namely x_5. As G has diameter 2, this means that there exists a vertex $z \in N_1$ with $z \notin \{y, y'\}$ that is adjacent to x_6 and v. Then z is not adjacent to any vertex of $\{x_1, x_3, x_4, x_5, x_7\}$, as otherwise z would have a list of size 1 due to Rule 3, and by the same rule, v would have a list of size 2. If $zy \in E$, then $\{y, z, x_6, x_7\}$ induces a C_4, which is not possible. Hence $zy \notin E$.

From the above, we find that $\{x_6, x_5, x_4, x_3, y, v, z\}$ induces a C_7; see also Figure 3.11. As $c(x_6) = 2$, $c(x_5) = 3$, $c(x_4) = 1$ and $c(x_3) = 3$, we find that $L_c(\{x_6, x_5, x_4\}) = \{1, 2, 3\}$ and $L_c(x_3) = L_c(x_5)$. Then $2 \notin L_c(v)$, due to Rule 7. Hence, $|L_c(v)| \leq 2$, a contradiction. We conclude that the propagation algorithm returned either `yes` or `no` in Case 2 as well. □

Finally, we complement our polynomial-time results by proving the following hardness result for graphs of diameter 4.

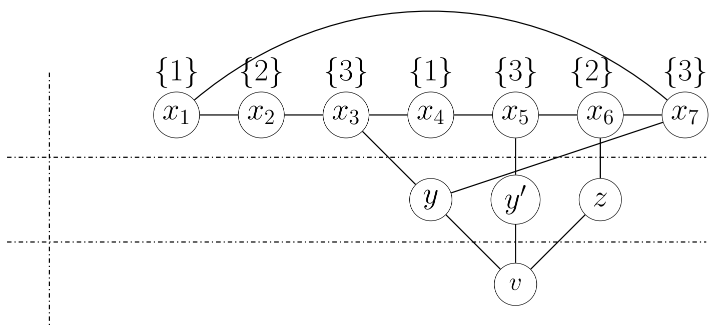

Figure 3.11: The situation that is described in Case 2 in the proof of Theorem 3.16. The set $\{x_6, x_5, x_4, x_3, y, v, z\}$ induces a C_7 to which Rule 7 should have been applied.

Theorem 3.17. *For every integer* $t \geq 8$*,* 3-Colouring *is* NP*-complete on the class of* $(C_4, C_6, C_7 \ldots, C_t)$*-free graphs of diameter* 4*.*

Proof. Note that the problem is readily seen to be in NP. To prove NP-hardness we modify the standard reduction for Colouring from the NP-complete problem Not-All-Equal 3-Satisfiability [87], where each variable appears in at most three clauses. So, given a CNF formula ϕ, we first construct a graph G as follows (see also Figure 3.12):

- add literal vertices v_i and v'_i for each variable x_i;
- add an edge between each v_i and v'_i;
- add a vertex z adjacent to every v_i and every v'_i;
- for each clause C_i add a triangle T_i with clause vertices $c_{i_1}, c_{i_2}, c_{i_3}$;
- fix an arbitrary order of the literals $x_{i_1}, x_{i_2}, x_{i_3}$ of C_i and for $j \in \{1, 2, 3\}$, add the edge $v_{i_j}c_{i_j}$ if x_{i_j} is positive and the edge $v'_{i_j}c_{i_j}$ if x_{i_j} is negative.

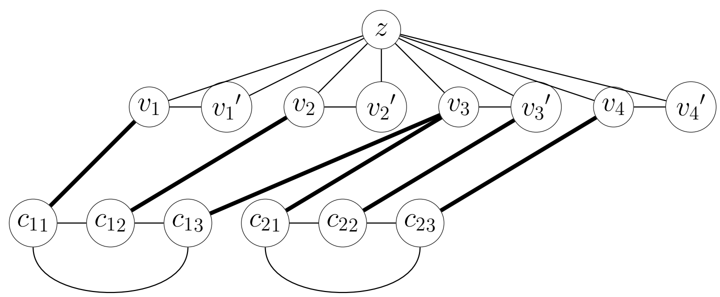

Figure 3.12: An example of a graph G in the reduction from Not-All-Equal 3-Satisfiability to 3-Colouring with clauses $C_1 = x_1 \wedge x_2 \wedge x_3$ and $C_2 = x_3 \wedge \neg x_3 \wedge x_4$. We obtain the graph G' by subdividing the thick edges (edges between literal and clause vertices) the same number of times and connecting the newly introduced vertices to z.

It is well known that ϕ has a truth assignment τ such that each clause contains at least one true literal and at least one false literal (call such a τ satisfying) if and only if G has a 3-colouring. For completeness we give a proof below.

First suppose ϕ has a satisfying truth assignment. Colour vertex z with colour 1, each true literal with colour 2 and each false literal with colour 3. Then, as each clause has a true literal and a false literal, each triangle T_i has neighbours in two different colours. Hence, we can complete the 3-colouring.

Now suppose G has a 3-colouring. Say z is assigned colour 1. Then each literal vertex has either colour 2 or colour 3. Moreover, each T_i must be adjacent to at least one literal vertex coloured 2 and to at least one literal vertex coloured 3. Hence, the truth assignment that sets literals whose vertices are coloured with colour 2 to be true and those coloured with colour 3 to be false is satisfying.

As every clause vertex is adjacent to a literal vertex and literal vertices are adjacent to z, every vertex has distance at most 2 from z. So G has diameter 4.

We modify G into a graph G': for some $p \geq 0$, subdivide each edge $v_{i_j}c_{i_j}$ and each edge $v'_{i_j}c_{i_j}$ p times and make each newly introduced vertex adjacent to z; see also Figure 3.12. Then G' has a 3-colouring if and only if G has a 3-colouring, as the new vertices will be alternatingly coloured by 2 and 3 if z has colour 1. Moreover, G' still has diameter 4, and it can be readily checked that every induced cycle of G of length at most p is either a C_3 (either a triangle T_i or a triangle containing z) or a C_5 (which must contain z). As we can make p arbitrarily large, the result follows. $\square$

3.6 Conclusions

In this chapter we have studied the effect on the complexity of the COLOURING and k-COLOURING problems of simultaneously restricting the input graph to some class characterised by a set of forbidden induced subgraphs and bounding its diameter. We classified the complexity of COLOURING for H-free graphs of diameter at most d for all but finitely many graphs H. The open cases are those where H is a claw and d is any integer and the case where H is a triangle and $d = 2$. For the k-COLOURING problem, we obtain both polynomial and NP-completeness results for H-free graphs for a number of polyads H. We also give both NP-completeness and polynomial-time results for graphs of girth at least g as well as for $\mathcal{H}$-free graphs where $\mathcal{H}$ is a set of one or more cycles. We now highlight the most interesting open problems from this section.

Resolving the first two would complete our classification of COLOURING for H-free graphs of diameter at most d.

Open Problem 1. *What is the complexity of* COLOURING *for claw-free graphs of diameter* 2*? Does there exist an integer* d *such that* COLOURING *is* NP*-complete for claw-free graphs of diameter at most* d*?*

Open Problem 2. *What is the complexity of* COLOURING *for* C_3*-free graphs of diameter* 2*?*

Next we consider some natural next steps towards classification of polyads for graphs of bounded diameter.

Open Problem 3. *What is the complexity of* 3-Colouring *for* $K_{1,4}^1$*-free graphs of diameter at most* 3 *and for* $K_{1,3}^2$*-free graphs of diameter at most* 3*?*

Open Problem 4. *Does there exist a polyad* S *such that* 3-Colouring *is* NP*-complete for* S*-free graphs of diameter at most* 3*?*

We also have a number of open cases for graphs of bounded diameter and girth.

Open Problem 5. *What are the complexities of the remaining open cases in table 3.2? In particular what is the complexity of* 3-Colouring *for (*C_3*-free) graphs of diameter at most* 2*?*

We are also left with many open cases for 3-Colouring and List 3-Colouring graphs of diameter at most 2 in the absence of one or two short cycles.

Open Problem 6. *What is the complexity of* 3-Colouring *(or* list 3-Colouring*) for* C_t*-free graphs of diameter at most* 2*,* $t \in \{3, 4, 7, 8 \ldots\}$*?*

Open Problem 7. *What is the complexity of* 3-Colouring *(or* List 3-Colouring*) for* (C_4, C_t)*-free graphs of diameter at most* 2*,* $t \geq 10$*?*

We may also consider k-Colouring for triangle-free graphs of diameter at most 2 where $k \geq 4$.

Open Problem 8. *What is the complexity of* k-Colouring *for triangle-free graphs of diameter at most* 2 *with* $k \geq 4$*?*

Finally we note that the construction of Mertzios and Spirakis for triangle-free graphs of diameter 3 seems to contain cycles C_s of arbitrary length for $s \geq 4$. With this in mind we pose the following open problems for graphs of diameter at most 3.

Open Problem 9. *What is the complexity of* 3-Colouring *(*List 3-Colouring*) for* C_t*-free graphs of diameter at most* 3 *with* $t \geq 4$*?*

Open Problem 10. *What is the complexity of* 3-Colouring *(Similarly* List 3-Colouring*) for* (C_4, C_t)*-free graphs of diameter at most* 3 *with* $t \in \{3, 5, 6 \ldots\}$*?*

Chapter 4

Variants of the Colouring Problem

In this chapter we study three variants of the colouring problem; ACYCLIC COLOURING, STAR COLOURING and INJECTIVE COLOURING. After outlining known results, we present a general polynomial-time result applicable to all three problems in Section 4.3. We then prove almost complete complexity dichotomies for each of the three problems in Sections 4.4, 4.5 and 4.6.

4.1 Known Results

Before discussing our new results and techniques, we first briefly discuss some known results.

Coleman and Cai [26] proved that for every $k \geq 3$, the problem ACYCLIC k-COLOURING is NP-complete for bipartite graphs. Afterwards, a number of hardness results appeared for other hereditary graph classes. Alon and Zaks [5] showed that ACYCLIC 3-COLOURING is NP-complete for line graphs of maximum degree 4. Kostochka [61] proved that ACYCLIC 3-COLOURING is NP-complete for planar graphs. This result was improved to planar bipartite graphs of maximum degree 4 by Ochem [81]. Mondal et al. [79] proved that ACYCLIC 4-COLOURING is NP-complete for graphs of maximum degree 5 and for planar graphs of maximum degree 7. Ochem [81] showed that ACYCLIC 5-COLOURING is NP-complete for planar bipartite graphs of maximum degree 8. We refer to the paper of Angelini and Frati [6] for a further discussion on acyclic colourable planar graphs.

Albertson et al. [2] and recently, Lei et al. [64] proved that Star 3-Colouring is NP-complete for planar bipartite graphs and line graphs, respectively. Shalu and Antony [89] showed that Star Colouring is NP-complete for co-bipartite graphs. Bodlaender et al. [10], Sen and Huson [88] and Lloyd and Ramanathan [67] proved that Injective Colouring is NP-complete for split graphs, unit disk graphs and planar graphs, respectively. Mahdian [72] proved that for every $k \geq 4$, Injective k-Colouring is NP-complete for line graphs. Injective 4-Colouring is also known to be NP-complete for cubic graphs (see [22]). Observe that Injective 3-Colouring is trivial for general graphs.

On the positive side, Lyons [71] proved that Acyclic Colouring and Star Colouring are polynomial-time solvable for P_4-free graphs; in particular, he showed that every acyclic colouring of a P_4-free graph is, in fact, a star colouring. We note that Injective Colouring is trivial for P_4-free graphs, as every injective colouring must assign each vertex of a connected P_4-free graph a unique colour. Afterwards, the results of Lyons have been extended to P_4-tidy graphs and $(q, q-4)$-graphs by Linhares-Sales et al. [66].

Cheng et al. [23] complemented the aforementioned result of Alon and Zaks [5] by proving that Acyclic Colouring is polynomial-time solvable for claw-free graphs of maximum degree at most 3. Calamoneri [22] observed that Injective Colouring is polynomial-time solvable for comparability and co-comparability graphs. Zhou et al. [96] proved that Injective Colouring is polynomial-time solvable for graphs of bounded treewidth.

4.2 Our Results

We focus on two important graph classes, namely the classes of graphs of high girth and line graphs of multigraphs, which are interesting classes on their own. If a problem is NP-complete for both classes, then it is NP-complete for H-free graphs whenever H has a cycle or a claw. It then remains to analyze the case when H is a *linear forest*, i.e., a disjoint union of paths; see [15, 21, 38, 62] for examples of this approach, which we discuss in detail below.

The construction of graph families of high girth and large chromatic number is well studied in graph theory (see, e.g. [35]). To prove their complexity dichotomy for COLOURING on H-free graphs, Král' et al. [62] first showed that for every integer $g \geq 3$, 3-COLOURING is NP-complete for the class of graphs of girth at least g. This approach can be readily extended to any integer $k \geq 3$ [34, 69]. The basic idea is to replace edges in a graph by graphs of high girth and large chromatic number, such that the resulting graph has sufficiently high girth and is k-colourable if and only if the original graph is so (see also [42, 53]).

By a more intricate use of the above technique we are able to prove that for every $g \geq 3$ and every $k \geq 3$, ACYCLIC k-COLOURING is NP-complete for the class of 2-degenerate bipartite graphs of girth at least g. This implies that ACYCLIC k-COLOURING is NP-complete for H-free graphs whenever H has a cycle. For STAR 3-COLOURING we are also able to prove that the problem remains NP-complete, for the class of graphs of girth at least g, for each $g \geq 3$. This implies that STAR 3-COLOURING is NP-complete for H-free graphs whenever H has a cycle. We prove the latter result for every $k \geq 4$ by combining known results, just as we use known results to prove that INJECTIVE k-COLOURING ($k \geq 4$) is NP-complete for H-free graphs if H has a cycle.

A classical result of Holyer [50] is that 3-COLOURING is NP-complete for line graphs (and Leven and Galil [65] proved the same for $k \geq 4$). As line graphs are claw-free, Král' et al. [62] used Holyer's result to show that 3-COLOURING is NP-complete for H-free graphs whenever H has an induced claw. For ACYCLIC 3-COLOURING, this follows from Alon and Zaks' result [5], which we extend to work for $k \geq 4$. For INJECTIVE k-COLOURING ($k \geq 4$) we can use the aforementioned result on line graphs of Mahdian [72].

The above hardness results leave us to consider the case where H is a linear forest. In Section 4.3 we will use a result of Atminas et al. [8] to prove a general result from which it follows that for fixed k, all three problems are polynomial-time solvable for H-free graphs if H is a linear forest. Hence, we have full complexity dichotomies for the three problems when k is fixed. However, these positive results do not extend to the case where k is part of the input: we prove NP-completeness for graphs that are P_r-free for some small value of r or have a small independence number, i.e., that are sP_1-free for some small integer s.

Our complexity results for H-free graphs are summarized in the following three theorems, proven in Sections 4.4–4.6, respectively; see Table 4.2 for a comparison.

	polynomial time	NP-complete
COLOURING [62]	$H \subseteq_i P_4$ or $P_1 + P_3$	Otherwise
ACYCLIC COLOURING	$H \subseteq_i P_4$	Otherwise except for $2P_2$
STAR COLOURING	$H \subseteq_i P_4$	Otherwise except for $2P_2$
INJECTIVE COLOURING	$H \not\subseteq_i 2P_1 + P_4$	Oth. except for $2P_1 + P_4$
k-COLOURING (see [24, 41, 60])	depends on k	infinitely many open H
ACYCLIC k-COLOURING ($k \geq 3$)	H is a linear forest	Otherwise
STAR k-COLOURING ($k \geq 3$)	H is a linear forest	Otherwise
INJECTIVE k-COLOURING ($k \geq 4$)	H is a linear forest	Otherwise

Table 4.1: The state-of-the-art for the three problems in this paper and the original COLOURING problem; both when k is fixed and part of the input. The open case for both ACYCLIC COLOURING and STAR COLOURING is $2P_2$. The open case for INJECTIVE COLOURING is $2P_1 + P_4$

.

Theorem 4.1. *Let H be a graph. For the class of H-free graphs it holds that:*

(i) ACYCLIC COLOURING *is polynomial-time solvable if $H \subseteq_i P_4$ and* NP*-complete otherwise for $H \neq 2P_2$;*

(ii) for every $k \geq 3$, ACYCLIC k-COLOURING *is polynomial-time solvable if H is a linear forest and* NP*-complete otherwise.*

Theorem 4.2. *Let H be a graph. For the class of H-free graphs it holds that:*

(i) STAR COLOURING *is polynomial-time solvable if $H \subseteq_i P_4$ and* NP*-complete if $H \not\subseteq_i P_4$ and $H \neq 2P_2$;*

(ii) for every $k \geq 3$, Star k-Colouring *is polynomial-time solvable if H is a linear forest and* NP*-complete otherwise.*

Theorem 4.3. *Let H be a graph. For the class of H-free graphs it holds that:*

(i) Injective Colouring *is polynomial-time solvable if $H \subsetneq_i 2P_1 + P_4$ and* NP*-complete if $H \not\subseteq 2P_1 + P_4$;*

(ii) for every $k \geq 4$, Injective k-Colouring *is polynomial-time solvable if H is a linear forest and* NP*-complete otherwise.*

In Section 4.7 we give a number of open problems that resulted from our systematic study.

	polynomial time	NP-complete
Colouring [62]	$H \subseteq_i P_4$ or $P_1 + P_3$	Otherwise
Acyclic Colouring	$H \subseteq_i P_4$	Otherwise except for $2P_2$
Star Colouring	$H \subseteq_i P_4$	Otherwise except for $2P_2$
Injective Colouring	$H \not\subseteq_i 2P_1 + P_4$	Oth. except for $2P_1 + P_4$
k-Colouring (see [24, 41, 60])	depends on k	infinitely many open H
Acyclic k-Colouring ($k \geq 3$)	H is a linear forest	Otherwise
Star k-Colouring ($k \geq 3$)	H is a linear forest	Otherwise
Injective k-Colouring ($k \geq 4$)	H is a linear forest	Otherwise

Table 4.2: The state-of-the-art for the three problems in this paper and the original Colouring problem; both when k is fixed and part of the input. The open case for both Acyclic Colouring and Star Colouring is $2P_2$. The open case for Injective Colouring is $2P_1 + P_4$

.

4.3 A General Polynomial Result

A biclique $K_{s,t}$ is called balanced if $s = t$. We say that a colouring c of a graph G satisfies the *balance biclique condition* (BB-condition) if c uses at least $k+1$ colours to colour G, where k is the order of a largest biclique that is contained in G as a (not necessarily induced) subgraph.

Let π be some colouring property; e.g., π could mean being acyclic, star or injective. Then π *can be expressed in* MSO_2 (monadic second-order logic with edge sets) if for every $k \geq 1$, the graph property of having a k-colouring with property π can be expressed in MSO_2. The general problem COLOURING(π) is to decide, on a graph G and integer $k \geq 1$, if G has a k-colouring with property π. If k is fixed, we write k-COLOURING(π). We now prove the following result.

Theorem 4.4. *Let H be a linear forest, and let π be a colouring property that can be expressed in MSO_2, such that every colouring with property π satisfies the BB-condition. Then, for every integer $k \geq 1$, k-*COLOURING(π) *is linear-time solvable for H-free graphs.*

Proof. Atminas, Lozin and Razgon [8] proved that that for every pair of integers ℓ and k, there exists a constant $b(\ell, k)$ such that every graph of treewidth at least $b(\ell, k)$ contains an induced P_ℓ or a (not necessarily induced) biclique $K_{k,k}$. Let G be an H-free graph, and let ℓ be the smallest integer such that $H \subseteq_i P_\ell$; observe that ℓ is a constant. Hence, we can use Bodlaender's algorithm [9] to test in linear time if G has treewidth at most $b(\ell, k) - 1$.

First suppose that the treewidth of G is at most $b(\ell, k) - 1$. As π can be expressed in MSO_2, the result of Courcelle [29] allows us to test in linear time whether G has a k-colouring with property π. Now suppose that the treewidth of G is at least $b(\ell, k)$. As G is H-free, G is P_ℓ-free. Then, by the result of Atminas, Lozin and Razgon [8], we find that G contains $K_{k,k}$ as a subgraph. As π satisfies the BB-condition, G has no k-colouring with property π. □

We now apply Theorem 4.4 to obtain the polynomial cases for fixed k in Theorem 4.1–4.3.

Corollary 1. *Let H be a linear forest. For every $k \geq 1$,* Acyclic k-Colouring*,* Star k-Colouring *and* Injective k-Colouring *are polynomial-time solvable for H-free graphs.*

Proof. All three kinds of colourings use at least s colours to colour $K_{s,s}$ (as the vertices from one bipartition class of $K_{s,s}$ must receive unique colours). Hence, every acyclic, star and injective colouring of every graph satisfies the BB-condition. Moreover, it is readily seen that the colouring properties of being acyclic, star or injective can all be expressed in MSO_2. Hence, we may apply Theorem 4.4. □

4.4 Acyclic Colouring

In this section, we prove Theorem 4.1. For a graph G and a colouring c, the pair (G, c) has a *bichromatic* cycle C if C is a cycle of G with $|c(V(C)| = 2$, i.e., the vertices of C are coloured by two alternating colours (so C is even). A path P in G is an *i-j-path* if the vertices of P have alternating colours i and j. We now prove the following result.

Lemma 4.1. *For every $k \geq 3$ and every $g \geq 3$,* Acyclic k-Colouring *is* NP*-complete for 2-degenerate bipartite graphs of girth at least g.*

Proof. We reduce from Acyclic k-Colouring, which is known to be NP-complete for bipartite graphs [26]. Recall that the *arboricity* of a graph is the minimum number of forests needed to partition its edge set. By counting the edges, a graph with arboricity at most t is $(2t-1)$-degenerate and thus $2t$-colourable. We start by taking a graph F that has no $2k(k-1)$-colouring and that is of girth at least g. By a seminal result of Erdős [35], such a graph F exists (and its size is constant, as it only depends on g which is a fixed integer). Notice that F does not admit a vertex-partition into k subgraphs with arboricity at most $k-1$, since otherwise F would be $2k(k-1)$-colourable. Now we consider the graph S obtained by subdividing every edge of G exactly once. The graph S is 2-degenerate and bipartite with the *old* vertices from F in one part and the *new* vertices of degree 2 in the other part. For contradiction, assume that S has an acyclic k-colouring. Assign the colour of every old vertex to the corresponding vertex of F and assign the colour of every new vertex to the corresponding edge of F. For every colour i, we consider the subgraph F_i of F induced by the vertices coloured i. For every $j \neq i$, the subgraph of S induced by the colours i and j is a forest. This implies that the subgraph of F_i induced by the edges coloured j is a forest. So the arboricity of F_i is at most $k-1$, that is, the number of colours distinct from i. By previous discussion, the chromatic number of F_i is at most $2(k-1)$, so that F is $2k(k-1)$-colourable. This contradiction shows that S has no acyclic k-colouring.

We repeatedly remove new vertices from S until we obtain a graph S' that is acyclically k-colourable. Let x_2 be the last vertex that we removed and let x_1 and x_3 be the neighbours of x_2 in S. By construction, S' is acyclically k-colourable and every acyclic k-colouring c of S' is such that:

- $c(x_1) = c(x_3)$, since otherwise setting $c(x_2) \notin \{c(x_1), c(x_3)\}$ would extend c to x_2. Without loss of generality, $c(x_1) = c(x_3) = 1$
- For every colour $i \neq 1$, S' contains a bichromatic path coloured 1 and i between x_1 and x_3, since otherwise setting $c(x_2) = i$ would extend c to x_2.

We are ready to describe the reduction. Let G be a bipartite instance of ACYCLIC k-COLOURING. We construct an equivalent instance G' with large girth as follows. For every vertex z of G, we fix an arbitrary order on the neighbours of z. We replace z of G by d vertices $\{z_1, z_2, \cdots, z_d\}$, where d is the degree of z. Then for $1 \leq i \leq d-1$, we take a copy of S' and we identify the vertex x_1 of S' with z_i and the vertex x_3 of S' with z_{i+1}. Now for every edge mn of G, say n is the i^{th} neighbour of m and m is the j^{th} neighbour of n, we add the edge $m_i n_j$ in G'.

Given an acyclic k-colouring of G, we assign the colour of z to $\{z_1, \cdots, z_d\}$ and extend the colouring to the copies of F', which gives an acyclic colouring of G'. Given an acyclic k-colouring of G', the copies of F' force the same colour on $\{z_1, \cdots, z_d\}$ and we assign this common colour to z, which gives an acyclic colouring of G.

Finally, notice that since G is bipartite, G' is bipartite, 2-degenerate and with girth at least g. □

In Lemma 4.2 we modify the construction of [5] for line graphs from $k = 3$ to $k \geq 3$.

Lemma 4.2. *For every $k \geq 3$,* Acyclic k-Colouring *is* NP*-complete for line graphs of multigraphs.*

Proof. For an integer $k \geq 1$, a *k-edge colouring* of a graph $G = (V, E)$ is a mapping $c : E \to \{1, \ldots, k\}$ such that $c(e) \neq c(f)$ whenever the edges e and f share an end-vertex. A *colour class* consists of all edges of G that are mapped by c to a specific colour i. The pair (G, c) has a *bichromatic* cycle C if C is a cycle of G with its edges coloured by two alternating colours. The notion of a *bichromatic* path is defined in a similar manner. We say that c is *acyclic* if (G, c) has no bichromatic cycle. For a fixed integer $k \geq 1$, the Acyclic k-Edge Colouring problem is to decide if a given graph has an acyclic k-edge colouring. Alon and Zaks proved that Acyclic 3-Edge Colouring is NP-complete for multigraphs [5]. We note that a graph has an acyclic k-edge colouring if and only if its line graph has an acyclic k-colouring. Hence, it remains to generalize the construction of Alon and Zaks [5] from $k = 3$ to $k \geq 3$. Our main tool is the gadget graph F_k, depicted in Figure 4.1, about which we prove the following two claims.

(i) The edges of F_k can be coloured acyclically using k colours, with no bichromatic path between v_1 and v_{14}.

(ii) Every acyclic k-edge colouring of F_k using k colours assigns e_1 and e_2 the same colour.

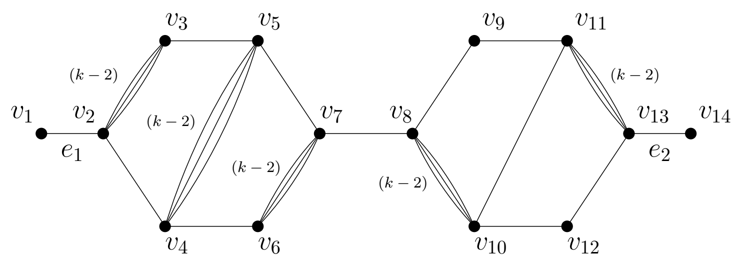

Figure 4.1: The gadget multigraph F_k. The labels on edges are multiplicities.

We first prove (ii). We assume, without loss of generality, that e_1 is coloured by 1, v_2v_4 by 2 and the edges between v_2 and v_3 by colours $3, \ldots, k$. The edge v_3v_5 has to be coloured by 1, otherwise we have a bichromatic cycle on $v_2v_3v_5v_4$. This necessarily implies that

- the edges between v_4 and v_5 are coloured by $3, \ldots, k$,
- the edge v_5v_7 is coloured by 2,
- the edge v_4v_6 is coloured by 1,
- the edges between v_6 and v_7 are coloured by $3, \ldots, k$, and
- the edge v_7v_8 is coloured by 1.

Now assume that the edge v_8v_9 is coloured by $a \in \{2, \ldots, k\}$ and the edges between v_8 and v_{10} by colours from the set A, where $A = \{2, \ldots, k\} \setminus a$. The edge $v_{10}v_{11}$ is either coloured a or 1. However, if it is coloured 1, v_9v_{11} is assigned a colour $b \in A$ and necessarily we have either a bichromatic cycle on $v_8v_9v_{11}v_{13}v_{12}v_{10}$, coloured by b and a, or a bichromatic cycle on $v_{10}v_{11}v_{13}v_{12}$, coloured by a and 1. Thus $v_{10}v_{11}$ is coloured by a. To prevent a bichromatic cycle on $v_8v_9v_{11}v_{10}$, the edge v_9v_{11} is assigned colour 1. The rest of the colouring is now determined as $v_{10}v_{12}$ has to be coloured by 1, the edges between v_{11} and v_{13} by A, $v_{12}v_{13}$ by a, and e_2 by 1. We then have a k-colouring with no bichromatic cycles of size at least 3 in F_k for every possible choice of a. This proves that e_1 and e_2 are coloured alike under every acyclic k-edge colouring.

We prove (i) by choosing a different from 2. Then there is no bichromatic path between v_1 and v_{14}.

We now reduce from k-Edge-Colouring to Acyclic k-Edge Colouring as follows. Given an instance G of k-Edge Colouring we construct an instance G' of Acyclic k-Edge Colouring by replacing each edge uv in G by a copy of F_k where u is identified with v_1 and v is identified with v_{14}.

If G' has an acyclic k-edge colouring c' then we obtain a k-edge colouring c of G by setting $c(uv) = c'(e_1)$ where e_1 belongs to the gadget F_k corresponding to the edge uv. If G has a k-edge colouring c then we obtain an acyclic k-edge colouring c' of G' by setting $c'(e_1) = c(uv)$ where e_1 belongs to the gadget corresponding to the edge uv. The remainder of each gadget F_k can then be coloured as described above. □

Lemma 4.3. Acyclic Colouring *is* NP-*complete for co-bipartite graphs.*

Proof. Alon et al. [4, Theorem 3.5] proved that deciding if a balanced bipartite graph on $2n$ vertices has a connected matching of size n is NP-complete. A matching is called *connected* if no two edges of the matching induce $2K_2$ in the given graph. We shall reduce from this problem to prove our theorem.

To this end, we claim that a balanced bipartite graph G with parts A and B such that $|A| = |B| = n$ has a connected matching of size n if and only if its complement has an acyclic colouring with n colours.

Suppose that there is an acyclic colouring c of $\overline{G}$ with n colours. Clearly, such colouring uses n colours on A and n colours on B. Vertices coloured with the same colour do not have an edge between them in $\overline{G}$ and thus are connected by an edge in G. Let us take the set of edges formed by each of the n colour classes. By the property of colouring, this is a matching in G and it is of size n. To see that it is also connected, suppose for a contradiction that there are two edges of the matching, say a_1b_1 and a_2b_2, forming an induced

$2K_2$ in G. Without loss of generality, $c(a_1) = c(b_1) = 1$ and $c(a_2) = c(b_2) = 2$. Now the induced $2K_2$ in G corresponds to a 4-cycle in $\overline{G}$ coloured with two colours, a contradiction with c being an acyclic colouring.

In the opposite direction, let us have a connected matching of size n in G. Colour the n vertices in A by $1, \ldots, n$. Let us colour the vertices of B with respect to the connected matching so that each vertex of B gets the colour of the vertex in A it is matched to. Indeed, this is a colouring of $\overline{G}$ by n colours. It remains to prove that it is acyclic. Any cycle in G having more than five vertices has by the definition of our colouring at least three colours. Therefore, a possible bichromatic cycle in $\overline{G}$ must be of size 4. The only possibility for such 4-cycle is that it is formed by two pairs of vertices, each one forming an edge of the connected matching in G. However, this implies that these two matching edges induce $2K_2$ in G, a contradiction with connectedness of the original matching. This completes the proof. $\square$

We combine the above results with a result of Lyons [71] to prove Theorem 4.1.

Theorem 4.1 (restated). *Let H be a graph. For the class of H-free graphs it holds that:*

(i) Acyclic Colouring *is polynomial-time solvable if $H \subseteq_i P_4$ and* NP-*complete if $H \not\subseteq_i P_4$ and $H \neq 2P_2$;*

(ii) for every $k \geq 3$, Acyclic k-Colouring *is polynomial-time solvable if H is a linear forest and* NP-*complete otherwise.*

Proof. We first prove (ii). First suppose that H contains an induced cycle C_p, then we use Lemma 4.1. Now assume H has no cycle so H is a forest. If H has a vertex of degree at least 3, then H has an induced $K_{1,3}$. As every line graph of a multigraph is $K_{1,3}$-free, we can use Lemma 4.2. Otherwise H is a linear forest and we use Corollary 1.

We now prove (i). Due to (ii), we may assume that H is a linear forest. If $H \subseteq_i P_4$, then we use the result of Lyons [71] that states that Acyclic Colouring is polynomial-time solvable for P_4-free graphs. Now suppose $3P_1 \subseteq_i H$. By lemma 4.3, acyclic colouring is NP-complete for co-bipartite and thus for $3P_1$-free graphs. It remains to consider the case where $H = 2P_2$, but this case is excluded from the theorem. □

4.5 Star Colouring

In this section we prove Theorem 4.2. We first prove the following lemma.

Lemma 4.4. *For every $g \geq 3$,* Star 3-Colouring *is* NP*-complete for planar graphs of girth at least g and maximum degree* 3.

Proof. We reduce from 3-Colouring, which is NP-complete even for planar graphs with maximum degree 4 [39]. Let G be an instance of this restricted version of graph of 3-Colouring. The vertex gadget D contains

- a cycle of length $12g$ with vertices $d_1, \cdots, d_{12g}$,
- $12g$ independent vertices $e_1, \cdots, e_{12g}$ such that e_i is adjacent to d_i for every $1 \leq i \leq 12g$, and
- 4 independent vertices f_1, f_2, f_3, f_4 such that f_i is adjacent to e_{3ig} for every $1 \leq i \leq 4$.

We construct an instance G' of Star 3-Colouring from G as follows. We consider a planar embedding of G and for every vertex x, we order the neighbours of x in a clockwise way. Then we replace x by a copy D_x of D. Now for every edge mn of G, say n is the i^{th} neighbour of m and m is the j^{th} neighbour of n, we add the edge between the vertex f_i of D_m and the vertex f_j of D_n, see Figure **??**.

It is not hard to check that in every star 3-colouring of D, the 4 vertices f_i get the same colour. moreover, there is no bichromatic path between any two vertices f_i.

Suppose that G admits a 3-colouring c of with colours in $\{0, 1, 2\}$. For every vertex x in G, we assign $c(x)$ to the vertices f_i in D_x and we assign $(c(x) + 1) \pmod 3$ to the vertices e_{3ig}. Then we extend this pre-colouring into a star 3-colouring of D_x. This gives a star 3-colouring of G'. Given a star 3-colouring of G', we assign to every vertex x in G the colour of the vertices f_i in D_x, which gives a 3-colouring of G.

Finally, notice that since G is planar with maximum degree 4, G' is planar with maximum degree 3 and girth at least g.

$\square$

Now we begin our development for Theorem 4.2.

Lemma 4.5. *Let $p \geq 4$ be a fixed integer. Then, for every $k \geq 3$,* Star k-Colouring *is* NP*-complete for C_p-free graphs.*

Proof. The case $k = 3$ follows from Lemma 4.4. We obtain NP-completeness for $k \geq 4$ by a reduction from Star 3-Colouring for C_p-free graphs by adding a dominating clique of size $k - 3$. $\square$

In Lemma 4.6 we extend the recent result of Lei et al. [64] from $k = 3$ to $k \geq 3$.

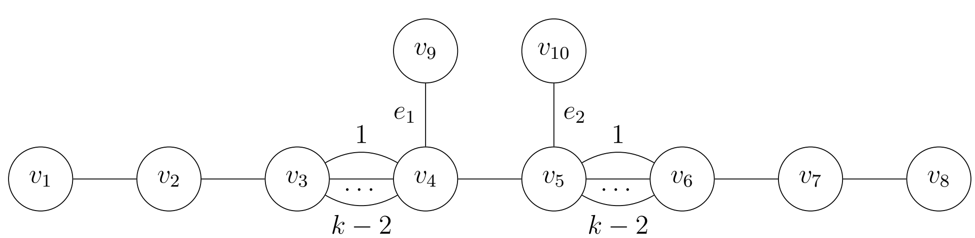

Figure 4.2: The gadget F_k in the proof of Lemma 4.6.

Lemma 4.6. *For every $k \geq 3$,* Star k-Colouring *is* NP*-complete for line graphs of multigraphs.*

Proof. A proper edge k-colouring c is a *star* k-edge colouring if the union of any two colour classes does not contain a path or cycle of on four edges. For a fixed integer $k \geq 1$, the Star k-Edge Colouring problem is to decide if a given graph has an star k-edge colouring. Lei et al. [64] proved that Star 3-Edge Colouring is NP-complete. Dvořák et al. [32] observed that a graph has a star k-edge colouring if and only if its line graph has a star k-colouring. Hence, it suffices to follow the proof of Lei et al.[64] in order to generalize the case $k = 3$ to $k \geq 3$. As such, we give a reduction from k-Edge Colouring to Star k-Edge Colouring which makes use of the gadget F_k in Figure 4.2. First we consider separately the case where the edges $e_1 = v_4v_9$ and $e_2 = v_5v_{10}$ are coloured alike and the case where they are coloured differently to show that in any star k-edge colouring of the gadget F_k shown in Figure 4.2, v_1v_2 and v_7v_8 are assigned the same colour.

Assume $c(e_1) = c(e_2) = 1$. We may then assume that the edge v_4v_5 is assigned colour 2 and the remaining $k-2$ colours are used for the multiple edges v_3v_4 and v_5v_6. The edge v_2v_3, and similarly v_6v_7, must then be assigned colour 1 to avoid a bichromatic P_5 on the vertices $\{v_2, v_3, v_4, v_5, v_6\}$ using any two of the multiple edges in a single colour. The edge v_1v_2, and similarly v_7v_8 must then be assigned colour 2 to avoid a bichromatic P_5 on the vertices $\{v_1, v_2, v_3, v_4, v_9\}$.

Next assume e_1 and e_2 are coloured differently. Without loss of generality, let $c(e_1) = 1$, $c(e_2) = 2$ and $c(v_4v_5) = 3$. The multiple edges v_3v_4 must then be assigned colours 2 and $4 \dots k$ and v_5v_6 colour 1 and colours $4 \dots k$. To avoid a bichromatic P_5 on the vertices $\{v_2, v_3, v_4, v_5, v_6\}$, v_2v_3 must be coloured 1. Similarly, v_6v_7 must be assigned colour 2. Finally, to avoid a bichromatic P_5 on the vertices $\{v_1, v_2, v_3, v_4, v_9\}$, v_1v_2 must be coloured 3. By a similar argument, v_7v_8 must also be coloured 3, hence v_1v_2 and v_7v_8 must be coloured alike.

We can then replace every edge e in some instance G of the problem k-Edge-Colouring by a copy of F_k, identifying its endpoints with v_1 and v_8, to obtain an instance G' of Star k-Edge-Colouring. If G is k-edge colourable we can star k-edgecolour G' by setting $c'(v_1v_2) = c'(v_7v_8) = c(e)$. If G' is star k-edge colourable, we obtain a k-edge colouring of G by setting $c(e) = c'(v_1v_2)$. $\square$

We now combine the above results with results of Albertson et al. [2], Lyons [71] and Shalu and Anthony [89] to prove Theorem 4.2.

Theorem 4.2 (restated). *Let H be a graph. For the class of H-free graphs it holds that:*

(i) Star Colouring *is polynomial-time solvable if $H \subseteq_i P_4$ and* NP-*complete if $H \not\subseteq_i P_4$ and $H \neq 2P_2$;*

(ii) for every $k \geq 3$, Star k-Colouring *is polynomial-time solvable if H is a linear forest and* NP-*complete otherwise.*

Proof. We first prove (ii). First suppose that H contains an induced odd cycle. Then the class of bipartite graphs is contained in the class of H-free graphs. Lemma 7.1 in Albertson et al. [2] implies, together with the fact that for every $k \geq 3$, k-Colouring is NP-complete, that for every $k \geq 3$, Star k-Colouring is NP-complete for bipartite graphs. If H contains an induced even cycle, then we use Lemma 4.5. Now assume H has no cycle, so H is a forest. If H contains a vertex of degree at least 3, then H contains an induced $K_{1,3}$. As every line graph of a multigraph is $K_{1,3}$-free, we can use Lemma 4.6. Otherwise H is a linear forest, in which case we use Corollary 1.

We now prove (i). Due to (ii), we may assume that H is a linear forest. If $H \subseteq_i P_4$, then we use the result of Lyons [71] that states that Star Colouring is polynomial-time solvable for P_4-free graphs. Now suppose $3P_1 \subseteq_i H$. A graph is *co-bipartite* if it is the complement of a bipartite graph. As bipartite graphs are C_3-free, co-bipartite graphs are $3P_1$-free.

Hence, we can use the result of Shalu and Antony [89] who proved that Star Colouring is NP-complete for co-bipartite graphs. It remains to consider the case where $H = 2P_2$, but this case was excluded from the statement of the theorem. □

4.6 Injective Colouring

In this section we prove Theorem 4.3. We first show a hardness result for fixed k.[1]

Lemma 4.7. *For every $k \geq 4$,* Injective k-Colouring *is* NP-*complete for bipartite graphs.*

Proof. We reduce from Injective k-Colouring; recall that this problem is NP-complete for every $k \geq 4$. Let $G = (V, E)$ be a graph. We construct a graph G' as follows. For each edge uv of G, we remove the edge uv and add two vertices u'_v, which we make adjacent to u, and v'_u, which we make adjacent to v. Next, we place an independent set I_{uv} of $k-2$ vertices adjacent to both u'_v and v'_u. Note that G' is bipartite: we can let one partition class consist of all vertices of $V(G)$ and the vertices of the I_{uv}-sets and the other one consist of all the remaining vertices (that is, all the "prime" vertices we added). It remains to show that G' has an injective k-colouring if and only if G has an injective k-colouring.

First assume that G has an injective k-colouring c. Colour the vertices of G' corresponding to vertices of G as they are coloured by c. We can extend this to an injective k-colouring c' of G' by considering the gadget corresponding to each edge uv of G. Set $c'(u'_v) = c'(v)$ and $c'(v'_u) = c'(u)$. We can now assign the remaining $k-2$ colours to the vertices of the independent sets. Clearly c' creates no bichromatic P_3 involving vertices in at most one

[1] We note that Janczewski et al. [55] proved that $L(p,q)$-Labelling is NP-complete for planar bipartite graphs, but in their paper they assumed that $p > q$.

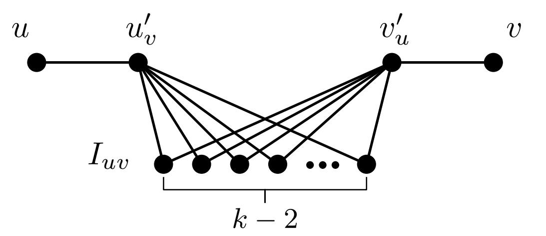

Figure 4.3: The edge gadget used in the proof of Lemma 4.7.

edge gadget. Assume there exists a bichromatic P_3 involving vertices in more than one edge gadget, then this path must consist of a vertex u of G together with two gadget vertices u'_v and u'_w which are coloured alike. This is a contradiction since it implies the existence of a bichromatic path v, u, w in G.

Now assume that G' has an injective k-colouring c'. Let c be the restriction of c' to those vertices of G' which correspond to vertices of G. To see that c is an injective colouring of G, note that we must have $c'(u'_v) = c'(v)$ and $c'(v'_u) = c'(u)$ for any edge uv. Therefore, if c induces a bichromatic P_3 on u, v, w, then c' induces a bichromatic P_3 on v'_u, v, v'_w. We conclude that c is injective. □

We now turn to the case where k is part of the input and first prove a number of positive results. An injective colouring c of a graph G is *optimal* if G has no injective colouring using fewer colours than c. An injective colouring c is *ℓ-injective* if every colour class of c has size at most ℓ. An ℓ-injective colouring c of a graph G is *ℓ-optimal* if G has no ℓ-injective colouring that uses fewer colours than c. We start with a useful lemma for the case where $\ell = 2$ that we will also use in our proofs.

Lemma 4.8. *A 2-optimal 2-injective colouring of a graph G can be found in polynomial time.*

Proof. Let c be a 2-injective colouring of G. Then each colour class of size 2 in G corresponds to a *dominating* edge of $\overline{G}$ (an edge uv of a graph is dominating if every other vertex in the graph is adjacent to at least one of u, v). Hence, the end-vertices of every non-dominating edge in $\overline{G}$ have different colours in G. Algorithmically, this means we may delete every non-dominating edge of $\overline{G}$ from $\overline{G}$; note that we do not delete the end-vertices of such an edge.

Let μ^* be the size of a maximum matching in the graph obtained from $\overline{G}$ after deleting all non-dominating edges of $\overline{G}$. The edges in such a matching will form exactly the colour classes of size 2 of an optimal 2-injective colouring of G. Hence, the injective chromatic number of G is equal to $\mu^* + (|V(G)| - 2\mu^*)$. It remains to observe that we can find a maximum matching in a graph in polynomial time by using a standard algorithm. □

We can now prove our first positive result.

Lemma 4.9. INJECTIVE COLOURING *is polynomial-time solvable for* $(P_1 + P_4)$*-free graphs.*

Proof. Let G be a $(P_1 + P_4)$-free graph. Since connected P_4-free graphs have diameter at most 2, no two vertices can be coloured alike in an injective colouring. Hence, the injective chromatic number of a P_4-free graph is equal to the number of its vertices. Consequently, INJECTIVE COLOURING is polynomial-time solvable for P_4-free graphs. From now on, we assume that G is not P_4-free.

We first show that any colour class in any injective colouring of G has size at most 2. For contradiction, assume that c is an injective colouring of G such that there exists some colour, say colour 1, that has a colour class of size at least 3. Let $P = x_1x_2x_3x_4$ be some induced P_4 of G.

We first consider the case where colour 1 appears at least twice on P. As no vertex has two neighbours coloured with the same colour, the only way in which this can happen is when $c(x_1) = c(x_4) = 1$. By our assumption, $G - P$ contains a vertex u with $c(u) = 1$. As G is $(P_1 + P_4)$-free, u has a neighbour on P. As every colour class is an independent set, this means that u must be adjacent to at least one of x_2 and x_3. Consequently, either x_2 or x_3 has two neighbours with colour 1, a contradiction.

Now we consider the case where colour 1 appears exactly once on P, say $c(x_h) = 1$ for some $h \in \{1, 2, 3, 4\}$. Then, by our assumption, $G - P$ contains two vertices u_1 and u_2 with colour 1. As G is $(P_1 + P_4)$-free, both u_1 and u_2 must be adjacent to at least one vertex of P, say u_1 is adjacent to x_i and u_2 is adjacent to x_j. Then $x_i \neq x_j$, as otherwise G has a vertex with two neighbours coloured 1. As every colour class is an independent set, we have that $x_h \notin \{x_i, x_j\}$, and hence, x_h, x_i, x_j are distinct vertices. Moreover, x_h is not a neighbour of x_i or x_j, as otherwise x_i or x_j has two neighbours coloured 1. Hence, we may assume without loss of generality that $h = 1$, $i = 3$ and $j = 4$. As every colour class is an independent set, u_1 and u_2 are non-adjacent. However, now $\{x_1, u_1, x_3, x_4, u_2\}$ induces a $P_1 + P_4$, a contradiction.

Finally, we consider the case where colour 1 does not appear on P. Let u_1, u_2, u_3 be three vertices of $G - P$ coloured 1. As before, $\{u_1, u_2, u_3\}$ is an independent set and each u_i has a different neighbour on P. We first consider the case where x_1 or x_4, say x_4 is not adjacent to any u_i. Then we may assume without loss of generality that u_1x_1 and u_2x_2 are edges. However, now $\{x_4, u_1, x_1, x_2, u_2\}$ induces a $P_1 + P_4$, which is not possible. Hence, we may assume without loss of generality that u_1x_1, u_2x_2 and u_4x_4 are edges of G. Again we find that $\{x_4, u_1, x_1, x_2, u_2\}$ induces a $P_1 + P_4$, a contradiction.

From the above, we find that each colour class in an injective colouring of G has size at most 2. This means we can use Lemma 4.8. □

We use the next lemma in the proofs of the results for $H = 2P_1 + P_3$ and $H = 3P_1 + P_2$.

Lemma 4.10. Injective Colouring *is polynomial-time solvable for* $4P_1$-*free graphs.*

Proof. Let $G = (V, E)$ be a $4P_1$-free graph on n vertices. We first analyze the structure of injective colourings of G. Let c be an optimal injective colouring of G. As G is $4P_1$-free, every colour class of c has size at most 3. From all optimal injective colourings, we choose c such that the number of size-3 colour classes is as small as possible. We say that c is *class-3-optimal*.

Suppose c contains a colour class of size 3, say colour 1 appears on three distinct vertices u_1, u_2 and u_3 of G. As G is $4P_1$-free, $\{u_1, u_2, u_3\}$ dominates G. As c is injective, this means that every vertex in $G - \{u_1, u_2, u_3\}$ is adjacent to exactly one vertex of $\{u_1, u_2, u_3\}$. Hence, we can partition $V \setminus \{u_1, u_2, u_3\}$ into three sets T_1, T_2 and T_3, such that for $i \in \{1, 2, 3\}$, every vertex of T_i is adjacent to u_i and not to any other vertex of $\{u_1, u_2, u_3\}$. If two vertices t, t' in the same T_i, say T_1, are non-adjacent, then $\{t, t', u_2, u_3\}$ induces a $4P_1$, a contradiction. Hence, we partitioned V into three cliques $T_i \cup \{u_i\}$. We call the cliques T_1, T_2, T_3, the *T-cliques* of the triple $\{u_1, u_2, u_3\}$.

Let $t \in T_i$ for some $i \in \{1, 2, 3\}$. For $i \in \{0, 1, 2\}$ we say that t is *i-clique-adjacent* if t has a neighbour in zero, one or two cliques of $\{T_1, T_2, T_3\} \setminus T_i$, respectively. By the definition of an injective colouring and the fact that every T_i is a clique, a 1-clique-adjacent vertex of $T_1 \cup T_2 \cup T_3$ belongs to a colour class of size at most 2, and a 2-clique-adjacent vertex of $T_1 \cup T_2 \cup T_3$ belongs to a colour class of size 1. Hence, all the vertices that belong to a colour class of size 3 are 0-clique-adjacent. The partition of $V(G)$ is illustrated in Figure 4.4.

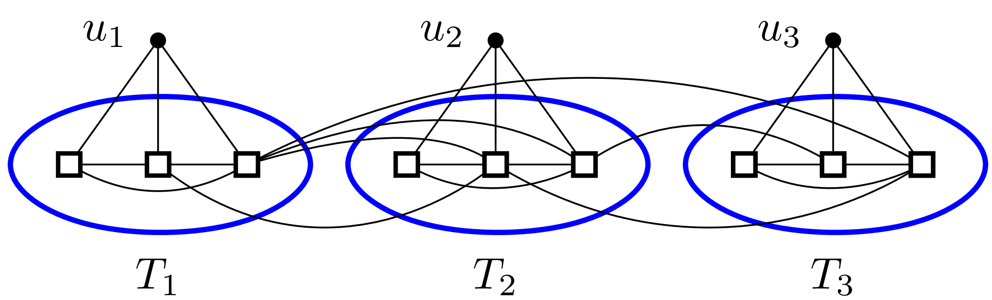

Figure 4.4: The partition of $V(G)$ from Lemma 4.10. The squares inside each T_i, $i \in \{1, 2, 3\}$, represent, from left to right, the sets of 0-clique-adjacent, 1-clique-adjacent and 2-clique-adjacent vertices in T_i, respectively.

We now use the fact that c is class-3-optimal. Let $t \in V \setminus \{u_1, u_2, u_3\}$, say $t \in T_1$, be i-clique-adjacent for $i = 0$ or $i = 1$. Then we may assume without loss of generality that t has no neighbours in T_2. If t belongs to a colour class of size 1, then we can set $c(u_2) := c(t)$ to obtain an optimal injective colouring with fewer size-3 colour classes, contradicting our choice of c.

We now consider the 0-clique-adjacent vertices again. Recall that these are the only vertices, other than u_1, u_2 and u_3, that may belong to a colour class of size 3. As every T_i is a clique, every colour class of size 3 (other than $\{u_1, u_2, u_3\}$) has exactly one vertex of each T_i. Let $\{w_1, w_2, w_3\}$ be another colour class of size 3 with $w_i \in T_i$ for every $i \in \{1, 2, 3\}$. Let $x \in T_1 \setminus \{w_1\}$ be another 0-clique-adjacent vertex. Then swapping the colours of w_1 and x yields another class-3-optimal injective colouring of G. Hence, we derived the following claim, which summarizes the discussion above and where statement (iv) follows from (i)–(iii).

Claim. *Let c be a class-3-optimal injective colouring of G such that $c(u_1) = c(u_2) = c(u_3)$ for three distinct vertices u_1, u_2, u_3 and with $p \geq 0$ other colour classes of size 3. Then the following four statements hold:*

(i) All 0-clique-adjacent and 1-clique-adjacent vertices belong to a colour class of size at least 2.

(ii) Let $S = \{y_1, \ldots, y_s\}$ be the set of 2-clique-adjacent vertices. Then $\{y_1\}, \ldots, \{y_s\}$ are exactly the size-1 colour classes.

(iii) For $i \in \{1, 2, 3\}$, let $x^i_1, \ldots, x^i_{q_i}$ be the 0-clique-adjacent vertices of T_i and assume without loss of generality that $q_1 \leq q_2 \leq q_3$. Then $p \leq q_1$ and if $p \geq 1$, we may assume without loss of generality that the size-3 classes, other than $\{u_1, u_2, u_3\}$, are $\{x^1_1, x^2_1, x^3_1\}, \ldots, \{x^1_p, x^2_p, x^3_p\}$.

(iv) The number of colours used by c, or equivalently, the number of colour classes of c is equal to $1+s+p+\frac{1}{2}(n-s-3(p+1)) = \frac{1}{2}n+\frac{1}{2}s-\frac{1}{2}p-\frac{1}{2}$.

We are now ready to present our algorithm. We first find, in polynomial time, an optimal 2-injective colouring of G by Lemma 4.8. We remember the number of colours used. Recall that the colour classes of every injective colouring of G have size at most 3. So, it remains to compute an optimal injective colouring for which at least one colour class has size 3.

We consider each triple u_1, u_2, u_3 of vertices of G and check if $\{u_1, u_2, u_3\}$ can be a colour class. That is, we check if $\{u_1, u_2, u_3\}$ is an independent set and has corresponding T-cliques T_1, T_2, T_3. This takes polynomial time. If not, then we discard $\{u_1, u_2, u_3\}$. Otherwise we continue as follows. Let $S = \{y_1, \ldots, y_s\}$ be the set of 2-clique adjacent vertices in $T_1 \cup T_2 \cup T_3$. Exactly the vertices of S will form the size-1 colour classes by Claim (ii). For $i \in \{1, 2, 3\}$, let $x_1^i, \ldots, x_{q_i}^i$ be the 0-clique-adjacent vertices of T_i, where we assume without loss of generality that $q_1 \leq q_2 \leq q_3$. By Claim (iii), any injective colouring of G which has $\{u_1, u_2, u_3\}$ as one of its colour classes has at most q_1 other colour classes of size 3 besides $\{u_1, u_2, u_3\}$. As can be seen from Claim (iv), the value $\frac{1}{2}n + \frac{1}{2}s - \frac{1}{2}p - \frac{1}{2}$ is minimized if the number p of size-3 colour classes is maximum.

From the above we can now do as follows. For $p = q_1, \ldots, 1$, we check if G has an injective colouring with exactly p colour classes of size 3. We stop as soon as we find a yes-answer or if p is set to 0. We first set $\{x_1^1, x_1^2, x_1^3\}, \ldots, \{x_p^1, x_p^2, x_p^3\}$ as the colour classes of size 3 by Claim (iii). Let Z be the set of remaining 0-clique-adjacent and 1-clique-adjacent vertices. We use Lemma 4.8 to check in polynomial time if the subgraph of G induced by $S \cup Z$ has an injective colouring that uses $s + \frac{1}{2}(n - s - 3(p+1))$ colours (which is the minimum number of colours possible). If so, then we stop and note that after adding the size-3 colour classes we obtained an injective colouring of G that uses $\frac{1}{2}n + \frac{1}{2}s - \frac{1}{2}p - \frac{1}{2}$ colours, which we remember. Otherwise we repeat this step after first setting $p := p - 1$.

As the above procedure for a triple u_1, u_2, u_3 takes polynomial time and the number of triples we must check is $O(n^3)$, our algorithm runs in polynomial time. We take the 3-injective colouring that uses the smallest number of colours and compare it with the number of colours used by the optimal 2-injective colouring that we computed at the start. Our algorithm then returns a colouring with the smallest of these two values as its output. $\square$

We use the previous lemma to prove our next lemma.

Lemma 4.11. Injective Colouring *is polynomial-time solvable for* $(2P_1+P_3)$*-free graphs.*

Proof. Let $G = (V, E)$ be a $(2P_1 + P_3)$-free graph. We may assume without loss of generality that G is connected and by Lemma 4.10 that G has an induced $4P_1$. We first show that any colour class in any injective colouring of G has size at most 2. For contradiction, assume that c is an injective colouring of G such that there exists some colour, say colour 1, that has a colour class of size at least 3. Let $U = \{u_1, \ldots, u_p\}$ for some $p \geq 3$ be the set of vertices of G with $c(u_i) = 1$ for $i \in \{1, \ldots, p\}$.

As c is injective, every vertex in $G - U$ has at most one neighbour in U. Hence, we can partition $G - U$ into (possibly empty) sets $T_0, \ldots, T_p$, where T_0 is the set of vertices with no neighbour in U and for $i \in \{1, \ldots, p\}$, T_i is the set of vertices of $G - U$ adjacent to u_i.

We first claim that T_0 is empty. For contradiction, assume $v \in T_0$. As G is connected, we may assume without loss of generality that v is adjacent to some vertex $t \in T_1$. Then $\{u_2, u_3, u_1, t, v\}$ induces a $2P_1+P_3$, a contradiction. Hence, $T_0 = \emptyset$.

We now prove that every T_i is a clique. For contradiction, assume that t and t' are non-adjacent vertices of T_1. Then $\{u_2, u_3, t, u_1, t'\}$ induces a $2P_1 + P_3$, a contradiction. Hence, every T_i and thus every $T_i \cup \{u_i\}$ is a clique.

We now claim that $p = 3$. For contradiction, assume that $p \geq 4$. As G is connected and U is an independent set, we may assume without loss of generality that there exist vertices $t_1 \in T_1$ and $t_2 \in T_2$ with $t_1t_2 \in E$. Then $\{u_3, u_4, u_1, t_1, t_2\}$ induces a $2P_1 + P_3$, a contradiction. Hence, $p = 3$.

Now we know that V can be partitioned into three cliques $T_1 \cup \{u_1\}$, $T_2 \cup \{u_2\}$ and $T_3 \cup \{u_3\}$. However, then G is $4P_1$-free, a contradiction. We conclude that every colour class of every injective colouring of G has size at most 2. This means we can use Lemma 4.8. $\square$

We also use Lemma 4.10 in the proof of our next result.

Lemma 4.12. Injective Colouring *is polynomial-time solvable for* $(3P_1+P_2)$*-free graphs.*

Proof. Let G be a $(3P_1 + P_2)$-free graph on n vertices. We may assume without loss of generality that G is connected and by Lemma 4.10 that G has an induced $4P_1$. As before, we will first analyze the structure of injective colourings of G. We will then exploit the properties found algorithmically.

Let c be an injective colouring of G that has a colour class U of size at least 3. So let $U = \{u_1, \ldots, u_p\}$ for some $p \geq 3$ be the set of vertices of G with, say colour 1. As c is injective, every vertex in $G - U$ has at most one neighbour in U. Hence, we can partition $G - U$ into (possibly empty) sets $T_0, \ldots, T_p$, where T_0 is the set of vertices with no neighbour in U and for $i \in \{1, \ldots, p\}$, T_i is the set of vertices of $G - U$ adjacent to u_i.

Assume that $p \geq 4$. As G is connected, there exists a vertex $v \notin U$ but that has a neighbour in U, say $v \in T_1$. Then $\{u_2, u_3, u_4, u_1, v\}$ induces a $3P_1 + P_2$, a contradiction. Hence, we have shown the following claim.

Claim 1. Every injective colouring of G is ℓ-injective for some $\ell \in \{1, 2, 3\}$.

We continue as follows. As $p = 3$ by Claim 1, we have $V(G) = U \cup T_0 \cup T_1 \cup T_2 \cup T_3$. Suppose T_0 contains two adjacent vertices x and y. Then $\{u_1, u_2, u_3, x, y\}$ induces a $3P_1 + P_2$, a contradiction. Hence, T_0 is an independent set. As G is connected, this means each vertex in T_0 has a neighbour in $T_1 \cup T_2 \cup T_3$.

Suppose T_0 contains two vertices x and y with the same colour, say $c(x) = c(y) = 2$. Let $v \in T_1 \cup T_2 \cup T_3$, say $v \in T_1$ be a neighbour of x. Then, as $c(x) = c(y)$ and c is injective, v is not adjacent to y. As T_0 is independent, x and y are not adjacent. However, now $\{u_2, u_3, y, x, v\}$ induces a $3P_1 + P_2$, a contradiction. Hence, every vertex in T_0 has a unique colour. Suppose T_0 contains a vertex x and $T_1 \cup T_2 \cup T_3$ contains a vertex v such that $c(x) = c(v)$. We may assume without loss of generality that $v \in T_1$. Then $\{u_2, u_3, x, v, u_1\}$ induces a $3P_1 + P_2$, a contradiction.

Finally, suppose that $T_1 \cup T_2 \cup T_3$ contain two distinct vertices v and v' with $c(v) = c(v')$. Let $x \in T_0$. Then x is not adjacent to at least one of v, v', say $xv \notin E$ and also assume that $v \in T_1$. Then $\{u_2, u_3, x, v, u_1\}$ induces a $3P_1 + P_2$. Hence, we have shown the following claim.

Claim 2. If c is 3-injective and U is a size-3 colour class such that G has a vertex not adjacent to any vertex of U, then all colour classes not equal to U have size 1.

We note that the injective colouring c in Claim 2 uses $n-2$ distinct colours.

We continue as follows. From now on we assume that $T_0 = \emptyset$. Every T_i is $(P_1 + P_2)$-free, as otherwise, if say T_1 contains an induced $P_1 + P_2$, then this $P_1 + P_2$, together with u_2 and u_3, forms an induced $3P_1 + P_2$, which is not possible. Hence, each T_i induces a complete r_i-partite graph for some integer r_i (that is, the complement of a disjoint union of r_i complete graphs). Hence, we can partition each T_i into r_i independent sets $T_i^1, \ldots, T_i^{r_i}$ such that there exists an edge between every vertex in T_i^a and every vertex in T_i^b if $a \neq b$. See also Figure 4.5.

Suppose G contains another colour class of size 3, say v_1, v_2 and v_3 are three distinct vertices coloured 2. If two of these vertices, say v_1 and v_2, belong to the same T_i, say T_1, then u_1 has two neighbours with the same colour. This is not possible, as c is injective. Hence, we may assume without loss of generality that $v_i \in T_i^1$ for $i \in \{1, 2, 3\}$.

Suppose that T_1^2 contains two vertices s and t. Then, as s and t are adjacent to v_1, both of them are not adjacent to v_2 (recall that $c(v_1) = c(v_2)$ and c is injective). Hence, $\{s, t, u_3, v_2, u_2\}$ induces a $3P_1 + P_2$ (see Figure 4.5). We conclude that for every $i \in \{1, 2, 3\}$, the sets $T_i^2, \ldots, T_i^{r_i}$ have size 1.

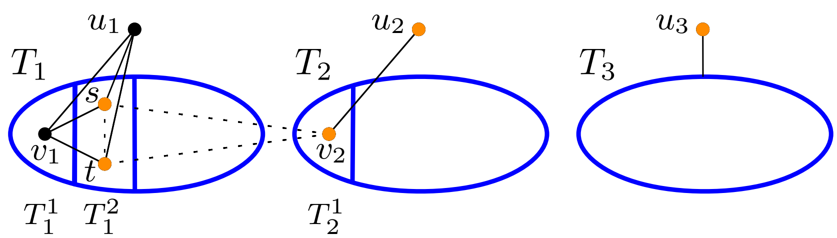

Figure 4.5: The situation in Lemma 4.12 where T_1^2 contains two vertices s and t. We show that this situation cannot happen, as it would lead to a forbidden induced $3P_1 + P_2$. Note that each u_i is adjacent to all vertices of T_i and not to any vertices of T_j for $j \neq i$. There may exist edges between vertices of different sets, but these are not drawn.

We will now make use of the fact that G contains an induced $4P_1$. We note that each $T_i \cup \{u_i\}$ is a clique, unless $|T_i^1| \geq 2$. As $V(G) = T_1 \cup T_2 \cup T_3 \cup \{u_1, u_2, u_3\}$ and G contains an induced $4P_1$, we may assume without loss of generality that T_1^1 has size at least 2. Recall that $v_1 \in T_1^1$. Let $z \neq v_1$ be some further vertex of T_1^1. If z is not adjacent to v_2, then $\{z, v_1, u_3, v_2, u_2\}$ induces a $3P_1 + P_2$, which is not possible. Hence, z is adjacent to v_2. For the same reason, z is adjacent to v_3. This is not possible, as c is injective and v_2 and v_3 both have colour 2. Hence, we have proven the following claim.

Claim 3. If c is 3-injective and U is a size-3 colour class such that each vertex of $G - U$ is adjacent to a vertex of U, then c has no other colour class of size 3.

We are now ready to present our polynomial-time algorithm. We first use Lemma 4.8 to find in polynomial time an optimal 2-injective colouring of G. We remember the number of colours it uses.

By Claim 1, it remains to find an optimal 3-injective colouring with at least one colour class of size 3. We now consider each set $\{u_1, u_2, u_3\}$ of three vertices. We discard our choice if u_1, u_2, u_3 do not form an independent set or if $V(G) \setminus \{u_1, u_2, u_3\}$ cannot be partitioned into sets $T_0, \ldots, T_4$ as described above. Suppose we have not discarded our choice of vertices u_1, u_2, u_3. We continue as follows.

If $T_0 \neq \emptyset$, then by Claim 2 the only 3-injective colouring of G (subject to colour permutation) with colour class $\{u_1, u_2, u_3\}$ is the colouring that gives each u_i the same colour and a unique colour to all the other vertices of G. This colouring uses $n-2$ colours and we remember this number of colours.

Now suppose $T_0 = \emptyset$. By Claim 3, we find that $\{u_1, u_2, u_3\}$ is the only colour class of size 3. Recall that no vertex in $G - \{u_1, u_2, u_3\} = T_1 \cup T_2 \cup T_3$ is adjacent to more than one vertex of $\{u_1, u_2, u_3\}$. Hence, we can apply Lemma 4.8 on $G - \{u_1, u_2, u_3\}$. This yields an optimal 2-injective colouring of $G - \{u_1, u_2, u_3\}$. We colour u_1, u_2, u_3 with the same colour and choose a colour that is not used in the colouring of $G - \{u_1, u_2, u_3\}$. This yields a 3-injective colouring of G that is optimal over all 3-injective colourings with colour class $\{u_1, u_2, u_3\}$. We remember the number of colours.

As the above procedure takes polynomial time and there are $O(n^3)$ triples to consider, we find in polynomial time an optimal 3-injective colouring of G that has at least one colour class of size 3 (should it exist). We compare the number of colours used with the number of colours of the optimal 2-injective colouring of G that we found earlier. Our algorithm returns the minimum of the two values as the output. Since both colourings are found in polynomial time, we conclude that our algorithm runs in polynomial time. $\square$

To prove our next hardness result, we first need to introduce some terminology and prove a lemma on COLOURING. A k-colouring of G can be seen as a partition of $V(G)$ into k independent sets. Hence, a (k-)colouring of G corresponds to a *(k-)clique-covering* of $\overline{G}$, which is a partition of $V(\overline{G}) = V(G)$ into k cliques. The *clique covering number* $\overline{\chi}(G)$ of G is the smallest number of cliques in a clique-covering of G. Note that $\overline{\chi}(G) = \chi(\overline{G})$.

Lemma 4.13. Colouring *is* NP-*complete for graphs with* $\overline{\chi} \leq 3$.

Proof. The List Colouring problem takes as input a graph G and a *list assignment* L that assigns each vertex $u \in V(G)$ a list $L(u) \subseteq \{1, 2, \ldots\}$. The question is whether G admits a colouring c with $c(u) \in L(u)$ for every $u \in V(G)$. Jansen [56] proved that List Colouring is NP-complete for co-bipartite graphs. This is the problem we reduce from.

Let G be a graph with a list assignment L and assume that $V(G)$ can be split into two (not necessarily disjoint) cliques K and K'. We set $A_1 := K$ and $A_2 := K \setminus K'$. As both A_1 and A_2 are cliques, we have that $\overline{\chi}(G) \leq 2$. We may assume without loss of generality that the union of all the lists $L(u)$ is $\{1, \ldots, k\}$ for some integer k. We now extend G by adding a clique A_3 of k new vertices $v_1, \ldots, v_k$ and by adding an edge between a vertex x_ℓ and a vertex $u \in V(G)$ if and only if $\ell \notin L(u)$. This yields a new graph G' with $\overline{\chi}(G') \leq 3$. It is readily seen that G has a colouring c with $c(u) \in L(u)$ for every $u \in V(G)$ if and only if G' has a k-colouring. □

We use Lemma 4.13 to prove the next lemma.

Lemma 4.14. Injective Colouring *is* NP-*complete for* $5P_1$*-free graphs.*

Proof. The problem is readily seen to belong to NP. We reduce from Colouring. Let (G, k) be an instance of this problem. By Lemma 4.13 we may assume that $V(G)$ can be partitioned into three cliques A_1, A_2 and A_3 with $|A_1| \leq |A_2| \leq |A_3|$. We may assume that $k \geq |A_3|$; otherwise (G, k) is a no-instance. Moreover, we may assume that every vertex u in every A_i has at least one neighbour in $V \setminus A_i$; otherwise u has degree $|A_i| - 1 \leq k - 1$ and hence, $G - u$ is k-colourable if and only if G is k-colourable.

We now construct a graph G' as follows. Let E^* be the set of edges in G whose end-vertices belong to different cliques of $\{A_1, A_2, A_3\}$. We add a clique A_0 of $|E^*|$ new vertices, so with exactly one vertex v_e for each edge $e = xy$ in E^*. We replace each $e \in E^*$ by the edges xv_e and yv_e. We denote the resulting graph by G' (see also Figure 4.6). We claim that G has a k-colouring if and only if G' has an injective $(k + |E^*|)$-colouring.

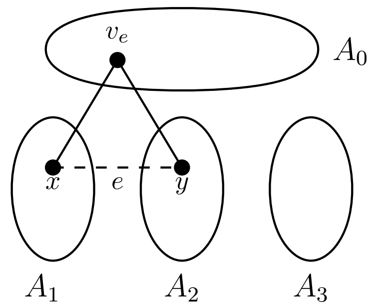

Figure 4.6: The graph G' constructed in the proof of Lemma 4.14.

First suppose that G has a k-colouring c. We give each vertex of A_0 a unique colour from $\{k + 1, \ldots, k + |E^*|\}$. This yields a $(k + |E^*|)$-colouring c' of G'. We claim that c' is injective. In order to see this, suppose that G' contains a vertex s that has two neighbours x and y with $c'(x) = c'(y)$. Every vertex in $A_1 \cup A_2 \cup A_3$ is only adjacent to vertices from its own clique A_i and A_0 and the colour sets used on those two cliques do not intersect. Hence, s belongs to A_0. Then, by definition of G', we find that x and y must belong to different cliques A_h and A_i. By construction, xy is an edge in E. As c is a k-colouring, this means that $c'(x) = c(x) \neq c(y) = c'(y)$, a contradiction. We conclude that c' is an injective $(k + |E^*|)$-colouring of G'.

Now suppose that G' has a $(k+|E^*|)$-colouring c'. Let $e \in A_0$ and suppose $c'(e) = 1$. We assume without loss of generality that e corresponds to an edge $e = xy$ in G with $x \in A_1$ and $y \in A_2$. Then, in G', we have that e is adjacent to x and to y. Hence, x and y are not coloured 1. As c' is injective, the neighbours of x and y have different colours. As A_1 and A_2 are cliques, x is adjacent to every vertex in $A_1 \setminus \{x\}$ and y is adjacent to every vertex in $A_2 \setminus \{y\}$. Hence, no vertex in $A_1 \cup A_2$ can have colour 1.

Now suppose that there exists a vertex $z \in A_3$ with $c'(z) = 1$. In G each vertex in every A_i has at least one neighbour in a different clique A_j. Hence,

z has a neighbour $f \in A_0$ in G' by construction of G'. However, now f has two neighbours, e and z, each with colour 1, contradicting the fact that c' is injective. We conclude that the colours of A_0 do not occur on $A_1 \cup A_2 \cup A_3$.

Recall that A_0 is a clique of size $|E^*|$. Hence, c' uses $|E^*|$ different colours. As no colour of A_0 occurs on $A_1 \cup A_2 \cup A_3$, this means that $|E^*|$ colours are not used on $V(G)$. Hence, the restriction c of c' to $V(G) = A_1 \cup A_2 \cup A_3$ is a k-colouring of the subgraph of G' induced by $A_1 \cup A_2 \cup A_3$.

We claim that c is even a k-colouring of G. Otherwise, if there exists an edge $e = xy$ with $c(x) = c(y)$, then e must be an edge in G that is not in G'. This means that x and y must belong to different cliques A_i and A_j. By construction, G' then contains the vertex $e = xy$. However, then $c'(x) = c(x) = c(y) = c(y')$ and e' has two neighbours with the same colour. This contradicts our assumption that c' is injective. We conclude that c is a k-colouring of G. $\square$

We combine the above results with results of Bodlaender et al. [10] and Mahdian [72] to prove Theorem 4.3.

Theorem 4.3 (restated). Let H be a graph. For the class of H-free graphs it holds that:

(i) Injective Colouring is polynomial-time solvable if $H \subsetneq_i 2P_1 + P_4$ and NP-complete if $H \not\subseteq_i 2P_1 + P_4$;

(ii) for every $k \geq 4$, Injective k-Colouring is polynomial-time solvable if H is a linear forest and NP-complete otherwise.

Proof. We first prove (ii). If $C_3 \subseteq_i H$, then we use Lemma 4.7. Now suppose $C_p \subseteq_i H$ for some $p \geq 4$. Mahdian [72] proved that for every $g \geq 4$ and $k \geq 4$, Injective k-Colouring is NP-complete for line graphs of bipartite graphs of girth at least g. These graphs may not be C_3-free but are C_p-free for $g \geq p + 1$. Now assume H has no cycle, so H is a forest. If H contains a vertex of degree at least 3, then H contains an induced $K_{1,3}$. As every line graph is $K_{1,3}$-free, we can use the aforementioned result of Mahdian [72] again. Otherwise H is a linear forest, in which case we use Corollary 1.

We now prove (i). Due to (ii), we may assume that H is a linear forest. If $H \subseteq_i P_1+P_4$ or $H \subseteq_i 2P_1+P_3$ or $H \subseteq_i 3P_1+P_2$, then we use Lemma 4.9, 4.11, or 4.12, respectively. Hence, if $H \subsetneq_i 2P_1+P_4$, then INJECTIVE COLOURING is polynomial-time solvable for H-free graphs. Now suppose that $H \not\subseteq_i 2P_1 + P_4$. If $2P_2 \subseteq_i H$, then the class of $(2P_2, C_4, C_5)$-free graphs are contained in the class of H-free graphs. The latter class coincides with the class of split graphs [37]. Recall that Bodlaender et al. [10] proved that INJECTIVE COLOURING is NP-complete for split graphs. In the remaining case it holds that $5P_1 \subseteq_i H$, and for this case we can use Lemma 4.14. $\square$

4.7 Conclusions

Our complexity study led to three complete and three almost complete complexity classifications (Theorems 4.1–4.3). Due to our systematic approach we were able to identify a number of open questions for future research, which we collect below.

In Lemma 4.1 we prove that for every $k \geq 3$ and every $g \geq 3$, ACYCLIC k-COLOURING is NP-complete for graphs of girth at least g. We would like to prove an analogous result for the third problem we considered. We recall that INJECTIVE 3-COLOURING is polynomial-time solvable for general graphs. Moreover, for every $k \geq 4$, INJECTIVE k-COLOURING is NP-complete for bipartite graphs (by Lemma 4.7) and thus for graphs of girth at least 4. Hence, we pose the following open problem.

Open Problem 11. *For every $g \geq 5$, determine the complexity of* INJECTIVE COLOURING *and* INJECTIVE k-COLOURING *($k \geq 4$) for graphs of girth at least g.*

This problem has eluded us and remains open and is, we believe, challenging. We have made progress for the corresponding high-girth problem for STAR 3-COLOURING in Lemma 4.4. However, we leave the high-girth problem for STAR k-COLOURING open for $k \geq 4$, as follows. We believe it represents an interesting technical challenge. At the moment, we only know that for $k \geq 4$, Star k-Colouring is NP-complete for bipartite graphs [2] and thus for graphs of girth at least 4.

Open Problem 12. *For every $g \geq 5$, determine the complexity of* Star k-Colouring *($k \geq 4$) for graphs of girth at least g.*

Naturally we also aim to settle the remaining open cases for our three problems in Table 4.2. In particular, there is one case left for Star Colouring and one case left for Injective Colouring. We note that the graph G' in the proof of Lemma 4.14 contains an induced $2P_1 + P_4$.

Open Problem 13. *Determine the complexity of* Injective Colouring *for $(2P_1 + P_4)$-free graphs.*

Open Problem 14. *Determine the complexity of* Star Colouring *for $2P_2$-free graphs.*

Recall that Acyclic Colouring and Injective Colouring, and also Colouring, are all NP-complete for $2P_2$-free graphs. However, none of the hardness constructions for these problems carry over to Star Colouring. In this context, the next open problem from Lyons [71] for a subclass of $2P_2$-free graphs is also interesting. A graph $G = (V, E)$ is *split* if $V = I \cup K$, where I is an independent set, K is a clique and $I \cap K = \emptyset$. The class of split graphs coincides with the class of $(2P_2, C_4, C_5)$-free graphs [37].

Open Problem 15. *Determine the complexity of* Star Colouring *for split graphs, or equivalently, $(2P_2, C_4, C_5)$-free graphs.*

Let $\omega(G)$ denote the clique number of G (size of a largest clique of G). Let $\chi_s(G)$ denote the the star chromatic number of G. It is easily observed (see also [70]) that if G is a split graph, then either $\chi_s(G) = \omega(G)$ or $\chi_s(G) = \omega(G) + 1$.

Chapter 5

Variants of Colouring for Graphs of Bounded Diameter

In Section 5.2 we present a result on chair-free graphs of bounded diameter. We then demonstrate that this result allows us to obtain polynomial-time algorithms for a number of problems related to 3-Colouring for chair-free graphs of bounded diameter. Next, we present an NP-completeness result which exhibits a limit on how far this result can be extended. Finally, in Section 5.3, we prove an NP-completeness result for the $L(1,2)$-labelling problem for graphs of diameter at most 2. In [19] and [20] a number of related results are proved.

In [20] we study the complexity of a further decision problem, Independent Set for H-free graphs of bounded diameter. In [19] we show that Acyclic 3-Colouring is NP-complete for graphs of diameter at most 4 but polynomial-time solvable for graphs of diameter at most 2. Additionally, we show that Star 3-Colouring is polynomial-time solvable for graphs of diameter at most 3 and NP-complete for graphs of diameter at most 8.

5.1 Known Results

We refer the reader to Chapter 3 for results on 3-Colouring for polyad-free graphs of bounded diameter and to Chapter 4 for results on Acyclic 3-Colouring and Star 3-Colouring for polyad-free graphs. Here, we consider three further problems; Independent Feedback Vertex Set, Independent Odd cycle Transversal and Near-Bipartiteness. Each of these problems are known to be NP-complete for H-free graphs when H contains a claw or a cycle [15]. The complexity of Near-bipartiteness and Independent Feedback Vertex Set for graphs of bounded diameter is classified in [14]. Both problems are polynomial-time solvable for graphs of diameter at most 2 and NP-complete for graphs of diameter at most 3. By reduction from 3-colouring for graphs of diameter 3 [78], Independent Odd Cycle Transversal is NP-complete for graphs of diameter at most 3. Like 3-colouring, its complexity remains open for graphs of diameter 2.

5.2 Chair-free Graphs of Bounded Diameter

We first prove our result for chair-free graphs.

Theorem 5.1. *Let $d \geq 1$ be an integer and G be a chair-free non-bipartite graph of diameter d with n vertices and m edges.*

1. *We can decide whether G is 3-colourable in $O(n+m)$ time.*
2. *If G is 3-colourable, then we find in $O(n+m)$ time either all 3-colourings of G, or a triangle xyz in G with exactly one vertex, say x, that has a set of private neighbours $P(x)$, and all 3-colourings of $G - P(x)$ that can be extended to 3-colourings of G. In both cases, we find at most $3^{9 \cdot 2^d + 8}$ 3-colourings.*

Proof. We first check in constant time whether G has at most $2d+1$ vertices. If so, we can determine in constant time all 3-colourings of G and there are at most 3^{2d+1}. Note that $3^{2d+1} < 3^{9 \cdot 2^d + 8}$. We proceed by assuming that G has at least $2d+2$ vertices and claim that G contains a triangle. We prove this claim by contradiction: assume that G is triangle-free. As G is not bipartite, there is an odd cycle in G. Let $x_1 x_2 \ldots x_p$ be a shortest odd cycle. As G is triangle-free and of diameter d, we find $5 \leq p \leq 2d+1$.

Moreover, as G has size at least $2d+2$, there is some vertex outside this cycle that has a neighbour on this cycle. Without loss of generality let us assume y with $y \notin \{x_1, x_2, \ldots, x_p\}$ is adjacent to x_1. As G is triangle-free, y does not have two consecutive neighbours on $x_1x_2\ldots x_p$. As G is chair-free and y is neither adjacent to x_2 nor to x_p, we find that y must be adjacent to x_3. We repeat this argument and obtain that y is adjacent to x_{2q+1} for every $0 \leq q \leq \lfloor \frac{p}{2} \rfloor$. In particular y is adjacent to the two consecutive vertices x_1 and x_p, a contradiction. We conclude that our assumption is false and that G contains a triangle.

We continue and show that we can compute a triangle, say T, of G in $O(n+m)$ time. Let u be a vertex of G. We partition $V(G)$ from $\{u\}$ and note that breadth-first search computes a breadth-first tree F, that is, F is a spanning tree of G such that each vertex of N_i has distance i to u in F for any i. As G is not bipartite, there has to be an edge e and an integer i such that e is incident to two vertices of N_i. We can compute such an edge that additionally minimizes i in $O(n+m)$ time. By adding this edge to F, we find an odd cycle C in G. As F is of diameter at most $2d$, we find that C has at most $2d+1$ vertices. Hence, we can determine in constant time a shortest induced odd cycle, say C', in $G[V(C)]$. We check in constant time whether C' is a triangle. If not, then C' has size at least 5. As G is of order at least $2d+2$, there is a vertex outside C' that has a neighbour on C'. We compute such a vertex, say y, in $O(n+m)$ time. As shown above, y has two consecutive neighbours on C'. As C' has at most $2d+1$ vertices, we can find such two vertices, and thus a triangle in G, in constant time.

Let $\{x, y, z\}$ be the vertex set of the triangle T. We partition $V(G)$ from $V(T)$ in $O(n+m)$ time. We additionally determine all private neighbours of the vertices of T and all vertices of N_1 that are adjacent to all vertices of T in linear time. If there is a vertex of the latter type, then G is not 3-colourable. Thus, we focus on the case where each vertex of N_1 is adjacent to at most two vertices of T. We compute in linear time the set N_1^* of all vertices of N_1

that have two neighbours of T. Clearly, $S = N_1 \setminus N_1^*$ consists of all private neighbours of the vertices of T, and its computation takes linear time. We proceed by considering $G - N_1$. We check in linear time if this graph has at most $9 \cdot 2^d + 2$ vertices.

Let us consider the subcase where $G - N_1$ has at least $9 \cdot 2^d + 3$ vertices. We claim that G is not 3-colourable and prove this claim by contradiction: assume that G is 3-colourable. Hence, G is K_4-free. Recall that every vertex of N_1 has at most two neighbours on T. Let $i \geq 1$ and u be a vertex of N_i. As G is *chair*-free, the neighbours of u in N_{i+1} form a clique. As G is K_4-free, we obtain that u has at most 2 neighbours in N_{i+1}. It follows that

$$3 + 9 \cdot 2^d \leq |N_0| + |N_2| + |N_3| + \ldots + |N_d| \leq 3 + |N_2| \cdot \sum_{i=2}^{d} 2^{i-2} < 3 + |N_2| \cdot 2^{d-1}.$$

Hence, $|N_2| > 18$. We let N_2^* be the neighbours of N_1^* in N_2. Consider the set N_{xy} of common neighbours of x and y in N_1^*. The set N_{xy} is an independent set as G is K_4-free. Every vertex $u \in N_2^*$ with a neighbour v in N_{xy} must be adjacent to every vertex in N_{xy}, as G is chair-free. For the same reason, no vertex of N_1^* has two non-adjacent neighbours in N_2^*. As G is K_4-free, this means that there are at most two vertices in N_2^* that are adjacent to the vertices of N_{xy}. By applying the same reasoning for every other pair of vertices of T, we find that N_2^* has size at most 6. Thus, $|N_2 \setminus N_2^*| > 12$. As every vertex of N_1 has at most two neighbours in N_2, it follows that $|S| > 6$.

We consider the subcase where at least two vertices, say x and y, of T have a private neighbour. Assume that x has two non-adjacent private neighbours u and v in S. Then these three vertices, together with y and a private neighbour $w \in S$ of y induce a chair unless w is adjacent to at least one of u and v. If w is adjacent to u but not to v, then $\{u, v, w, x, z\}$ induces a chair. Hence, w is adjacent to both u and v, but then $\{u, v, w, y, z\}$ induces a chair. Therefore, the private neighbours of every vertex of T form a clique. As G is 3-colourable, we find $|S| \leq 6$ if at least two vertices of T have private neighbours. Thus, we obtain that all vertices of S are adjacent to a common vertex, say x, of T. As G is 3-colourable, we find that $G[S]$ is bipartite.

Due to the above, we partition S into two independent sets A and B (one of these two sets might be empty). As G is chair-free and as A is independent, the vertices of A share the same set of neighbours in N_2. Similarly, the vertices of B share the same set of neighbours in N_2. As G is chair-free and K_4-free, the neighbourhood of every vertex of $A \cup B$ is a clique of size at most 2. We conclude that the total number of vertices in N_2 with a neighbour in S is at most 4, a contradiction as $|N_2 \setminus N_2^*| > 12$. We find that G is not 3-colourable and proceed by assuming that $G - N_1$ has at most $9 \cdot 2^d + 2$ vertices.

We consider every vertex labelling of $G - N_1$ with labels $1, 2, 3$ and determine in $O(n+m)$ time which ones lead to a 3-colouring of G. We discard those labellings which are not a 3-colouring of $G - N_1$. Given a 3-colouring of $G - N_1$, each vertex of N_1^* receives the remaining available label that is not used for its neighbours of T. Note that this assignment takes linear time. We discard in $O(n+m)$ time those labellings which do not lead to a 3-colouring of $G - S$.

Let us take an arbitrary 3-colouring of $G - S$. We assign lists to the vertices of G as follows: we set $L(u) = \{i\}$, where i is the label of u, if $u \notin S$ and we set $L(u) = \{1, 2, 3\} \setminus \{i\}$, where i is the label of the unique neighbour of u on T, if $u \in S$. Thus, checking whether a given 3-colouring of $G - S$ leads to a 3-colouring of G takes $O(n + m)$ time by Theorem 3.4 as (G, L) is an instance of 2-List Colouring. We discard those 3-colourings of $G - S$ which do not lead to a 3-colouring of G. If no 3-colouring of $G - S$ lead to a 3-colouring of G, then G is not 3-colourable. Hence, we proceed by assuming that at least one does, and so we find that G is 3-colourable. As there are at most $3^{9 \cdot 2^d + 2}$ vertex labellings of $G - N_1$, we can determine all 3-colourings of $G - S$ that can be extended to 3-colourings of G in $O(n + m)$ time, and there are at most $3^{9 \cdot 2^d + 2}$ such colourings of $G - S$.

If $S = \emptyset$, then $G - S$ equals G. We consider the subcase where at least two vertices of T have a private neighbour. As shown above, the private neighbours of every vertex of T form a clique. If $|S| > 6$, which we check in constant time, then G is not 3-colourable. Otherwise, as we have at most $3^{9 \cdot 2^d + 2}$ 3-colourings of $G - S$, we find at most $3^{9 \cdot 2^d + 8}$ 3-colourings of G and their computation takes $O(n + m)$ time. We finally consider the subcase where all vertices of S are adjacent to a single vertex, say x, of T. We conclude that $S = P(x)$, which completes our proof. $\square$

We next use this result to show that each of the problems Independent Feedback Vertex Set, Independent Odd Cycle Transversal and Near Bipartiteness are polynomial-time solvable for chair-free graphs of bounded diameter. Our proof requires the following theorem. Note that a complex is a complete bipartite graph minus the edges of some (possibly empty) matching.

Theorem 5.2 ([3])**.** *If G is a connected bipartite chair-free graph, then G is a cycle or a path or a complex.*

Theorem 5.3. *If $d \geq 1$, then* 3-Colouring, Acyclic 3-Colouring, Star 3-Colouring, Independent Odd Cycle Transversal, Independent Feedback Vertex Set, *and* Near-Bipartiteness *can be solved in $O(n + m)$ time for chair-free graphs of diameter at most d.*

Proof. Let G be a chair-free graph of diameter at most d with n vertices and m edges. Note that G is acyclic 3-colourable or star 3-colourable only if G is 3-colourable. Moreover, if I is an independent set of G for which $G - I$ is a bipartite graph, then G is 3-colourable. Hence, our problems require all the yes-instances to be 3-colourable. If $d = 1$, then G is 3-colourable if and only if G has at most 3 vertices, and so each of our problems can be solved in constant time. We proceed by assuming $d \geq 2$ and check in $O(n + m)$ time whether G is bipartite.

Case 1: G is bipartite.

Note that G is 3-colourable, near-bipartite, and has an independent odd cycle transversal of size at most k for any integer k. We can determine the parts, say S_1 and S_2, of G in $O(n+m)$ time. We may assume without loss of generality that $|S_1| \geq |S_2|$. We check in constant time whether $|S_1| + |S_2| \leq \max\{8, 2d\}$ and if so, then we can solve each of our problems in constant time. Otherwise, we find that $|S_1| \geq 5$. As bipartite graphs of maximum degree at most 2 and diameter at most d are paths or cycles of at most $2d$ vertices, we find that G has a vertex of degree at least 3, and so G is a complex by Theorem 5.2.

We first claim that in the case where G is a complex with $|S_1| \geq 5$, G is star 3-colourable if $|S_2| \leq 2$ and acyclic 3-colourable only if $|S_2| \leq 2$. Note that this claim completes the bipartite case for ACYCLIC 3-COLOURING and STAR 3-COLOURING as we can decide whether $|S_2| \leq 2$ or not in constant time and as every star 3-colouring of a graph is acyclic. We prove our claim as follows: If $|S_2| \leq 2$, then, for any $s \in S_2$, $G - s$ is a forest each component of which is of diameter at most 2, and thus G is star 3-colourable with colour classes $S_1, S_2 \setminus \{s\}$, and $\{s\}$. If $|S_2| \geq 3$, then let c be an arbitrary 3-colouring of G. By the pigeonhole principle there exists a colour class X of c that contains at least two vertices of S_1, and so $X \cap S_2 = \emptyset$. As $|S_2| \geq 3$, there are two vertices $s_2, s_2' \in S_2$ that are coloured alike. As $|S_1| \geq 5$, and as s_2 and s_2' are of degree at least $|S_1| - 1$, we find that s_2 and s_2' have at least three common neighbours in S_1 two of which, say s_1 and s_1', are coloured alike. Hence, $s_1 s_2 s_1' s_2'$ is a bichromatic 4-cycle. We conclude that every 3-colouring of G is not acyclic, which completes the proof of our claim.

It remains to consider INDEPENDENT FEEDBACK VERTEX SET for complexes with at least 9 vertices. Let k be an arbitrary integer. We claim that in the case where G is a complex with $|S_1| \geq 5$, G has an independent feedback vertex set of size at most k if and only if $k \geq |S_2| - 1$. Note that the latter can be decided in linear time. We prove our claim as follows: If $|S_2| \leq 2$, then $G - s$ is a forest for any $s \in S_2$ and G has an independent feedback vertex set of size at most k. Hence, we may assume $|S_2| \geq 3$. Let I be a minimum independent feedback vertex set in G. Such a set exists as G is bipartite. As $S_2 \setminus \{s\}$ is independent and as $G[S_1 \cup \{s\}]$ is a forest for each vertex $s \in S_2$, we find $|I| \leq |S_2| - 1$. For the sake of a contradiction, let us assume $|I| \leq |S_2| - 2$. Hence, any two vertices of $S_2 \setminus I$ have at least $|S_1| - 2$ common neighbours in S_1, and so $|I \cap S_1| \geq |S_1| - 3 \geq 2$. Moreover, $I = I \cap S_1$ as every vertex of S_2 has a neighbour in $I \cap S_1$ and as I is independent. As I is an independent feedback vertex set with $|I| \leq |S_1| - 2$, any two vertices of $S_1 \setminus I$ do not have two common neighbours in S_2 and so $|S_2| \leq 3$. Hence,

$$5 \leq |S_1| \leq |I| + 3 \leq |S_2| + 1 \leq 4,$$

a contradiction. As $|I| = |S_2| - 1$, the proof of our claim is complete.

Case 2: G is not bipartite.

Outline. As our problems require all the yes-instances to be 3-colourable, we check first whether G is 3-colourable. If so, then we compute an induced subgraph H of G and determine the set $\mathcal{C}$ of all its 3-colourings that can be extended to 3-colourings of G. As we compute H by applying Theorem 5.1, we find that $|\mathcal{C}| \leq 3^{9 \cdot 2^d + 8}$. We then distinguish some subcases. In some of them we further branch by extending our 3-colourings. However, in some of them we find that H equals G, and so our six problems are solvable in $O(n + m)$ time as $\mathcal{C}$ is of constant size. As an implicit step, we apply this finding whenever H is the whole graph G.

Full Proof. We first apply Theorem 5.1. We continue by assuming that G is 3-colourable. In fact, the only remaining case is that where the lemma provides a triangle T on vertex set $\{x, y, z\}$, a vertex x of T that has private neighbours, and the set of all 3-colourings of $G - P(x)$ that can be extended to 3-colourings of G. Note that we have at most $3^{9 \cdot 2^d + 8}$ such 3-colourings. We partition $V(G)$ from $V(T)$.

We find that $G[P(x)]$ is bipartite, as G is 3-colourable, but not necessarily connected. We extend each 3-colouring of $G - P(x)$ that can be extended to a 3-colouring of G to some vertices of $P(x)$. Let c be an arbitrary 3-colouring of $G - P(x)$ that can be extended to a 3-colouring of G. For $i \in \{0, 1, 2\}$, we compute in $O(n + m)$ time the set S_i of all vertices of $P(x)$ which have i available colours with respect to c, that is, S_i is the set of all vertices of $P(x)$ which have neighbours in $3 - i$ colours. As c can be extended to a 3-colouring of G, we find that S_0 is empty. It takes $O(n+m)$ time to determine the available colour of each vertex in S_1. Furthermore, we can extend c by breadth-first search in the same time to the vertices of those components of $G[P(x)]$ that contain at least one vertex of S_1.

Let S_c be the set of vertices that induce those components of $G[P(x)]$ that do not contain a vertex of S_1. Note that S_c can be computed in $O(n+m)$ time and that all neighbours of all vertices of S_c in $V(G) \setminus S_c$ are coloured alike. Moreover, every vertex of S_c has its neighbours in $[N(y) \cap N(z)] \cup N_2 \cup S_c \cup \{x\}$ by definition. As c can be extended to a 3-colouring of G, we find that our approach leads to a 3-colouring, say c', of $G - S_c$. As there are at most $3^{9 \cdot 2^6 + 2}$ 3-colourings of $G - P(x)$, we find at most $3^{9 \cdot 2^6 + 2}$ such triples (c, c', S_c). Furthermore, for each 3-colouring c_s of $G - P(x)$, there exists a triple $(c_s, {c_s}', S_{c_s})$ if c_s can be extended to a 3-colouring of G. We proceed by considering the case where $S_c \neq \emptyset$ as otherwise $G = G - S_c$. We continue by distinguishing on the problems we are considering. Recall that G is 3-colourable.

Subcase 2.1: Acyclic 3-Colouring and Star 3-Colouring

We check whether for some triple (c, c', S_c), the 3-colouring c' of $G - S_c$ that can be extended to an acyclic 3-colouring or star 3-colouring of G. By this approach, we clearly solve Acyclic 3-Colouring and Star 3-Colouring.

Let (c, c', S_c) be an arbitrary triple as defined above. Recall that a star 3-colouring of a graph is acyclic. In time $O(n + m)$, we can determine the components of $G[S_c]$ and check whether $G[S_c]$ is a forest. If not, then $G[S_c \cup \{x\}]$, and thus G, is not acyclic 3-colourable. We continue and assume that

$G[S_c]$ is a forest. We check in $O(n+m)$ time if a vertex of S_c has a neighbour in $N(y) \cap N(z)$. If so, say $s \in S_c$ is adjacent to $v \in N(y) \cap N(z)$, then c' cannot be extended to an acyclic 3-colouring of c as either s and x are coloured alike or one of $\{svyx, svzx\}$ is a bichromatic 4-cycle.

We proceed by assuming that S_c has its neighbours in $N_2 \cup S_c \cup \{x\}$. As G is chair-free, every two non-adjacent vertices of S_c share the same neighbours in N_2 and, if there exists such a neighbour, then these two vertices have to be coloured differently to avoid a bichromatic 4-cycle. Therefore, in any acyclic extension of c' to G, each of the two colour classes in S_c either has size at most 1 or has no neighbour in N_2. We check in constant time if S_c is of size at most 2. If so, then there are at most 4 possibilities to extend c' to a 3-colouring of G. Hence, we may assume $|S_c| \geq 3$. We check in $O(n+m)$ time if a vertex of S_c has a neighbour in N_2.

Let us consider the subcase where $s \in S_c$ has a neighbour, say v, in N_2. Let G_s be the component of $G[P(x)]$ that contains s. Note that there are at most two possibilities to extend c' to the vertices of G_s. We check in linear time if $S_c \setminus V(G_s)$ is of size at least 2. If so, say $s_1, s_2 \in S_c \setminus V(G_s)$, then v is a neighbour of s, s_1, and s_2. Thus, $xs'_1vs'_2$ is a bichromatic 4-cycle for two vertices s'_1 and s'_2 of $\{s, s_1, s_2\}$. We conclude that c' cannot be extended to an acyclic 3-colouring of G. Hence, we may assume $|S_c \setminus V(G_s)| \leq 1$, and so there are at most four possibilities to extend c' to a 3-colouring of G each of which can be obtained in $O(n+m)$ time.

We proceed by assuming that no vertex of S_c has a neighbour in N_2. In other words, each vertex of S_c has its neighbours in $S_c \cup \{x\}$. As x is a cut-vertex of G, any extension of c' to a 3-colouring of G is acyclic if and only if c' is acyclic. Hence, we can solve Acyclic 3-Colouring.

We now check in $O(n+m)$ time if each component of $G[S_c]$ is of diameter at most 2. If not, then $G[S_c \cup \{x\}]$, and thus G is not star 3-colourable. Let us proceed by assuming that each component of $G[S_c]$ is of diameter at most 2. We find that every 3-colouring of $G[S_c \cup \{x\}]$ is a star 3-colouring.

In other words, we can restrict ourselves to those 3-colouring extensions of c' to G that assign one colour to all vertices of S_c if S_c is independent, and an arbitrary 3-colouring extensions of c' to G if S_c is not independent. Note that we can check in $O(n + m)$ time whether S_c is independent. We find in both subcases at most two extensions of c' to G.

Subcase 2.2: Independent Odd Cycle Transversal
Let k be an arbitrary integer. We check whether some triple (c, c', S_c) consists of a 3-colouring c' of $G - S_c$ that can be extended to a 3-colouring of G where one colour class is an independent odd cycle transversal of size at most k. As all the yes-instances require G to be 3-colourable, this approach clearly solves Independent Odd Cycle Transversal.

Let (c, c', S_c) be an arbitrary triple as defined above. Moreover, let X, Y, Z be the colour classes of c' with $x \in X$, $y \in Y$, and $z \in Z$. Clearly, X, Y, and Z can be computed in linear time. We decide in linear time which of $\{Y, Z\}$ is of smaller size, say $|Y| \leq |Z|$.

Recall that all vertices of S_c have their neighbours in $S_c \cup X$. Note that c' can be extended to a 3-colouring of G by 2-colourings of $G[S_c]$ on the colours that c' assigns to y and z, and these are the only possibilities. We find that the smallest possible colour class of a 3-colouring of G that extends c' consists of the vertices either in X or in $Y \cup W$, where W is the smallest possible colour class of a 2-colouring of $G[S_c]$. As we can compute the components of $G[S_c]$ and its parts in $O(n+m)$ time, we can find W in the same time. Hence, the smallest possible independent odd cycle transversal of G that is a colour class of an extension of c' to a 3-colouring of G is of size $\min\{|X|, |Y \cup W|\}$. We can compare the sizes of X and $Y \cup W$ with k in linear time.

Subcase 2.3: The problems Independent Feedback Vertex Set and Near-Bipartiteness
Let k be an arbitrary integer. We check whether some triple (c, c', S_c) consists of a 3-colouring c' of $G - S_c$ that can be extended to a 3-colouring of G where one colour class is an independent feedback vertex set (of size at most k).

As all the yes-instances require G to be 3-colourable, this approach clearly solves Independent Feedback Vertex Set and Near-Bipartiteness.

Let (c, c', S_c) be an arbitrary triple as defined above. Moreover, let X, Y, Z be the colour classes of c' with $x \in X$, $y \in Y$, and $z \in Z$. Clearly, X, Y, and Z can be computed in linear time. We check first whether $G - X$ is a forest in $O(n + m)$ time. If so, then we find that X is an independent feedback vertex set of G and we can determine its size in linear time. Hence, we proceed by assuming that $G - X$ contains a cycle or $|X| > k$. As we aim to find an extension of c' to a 3-colouring of G whose one colour class is an independent feedback vertex set (of size at most k), we find that such a set consists of the vertices of Y or of Z, and the vertices of some set $A \subseteq S_c$.

Recall that all vertices of S_c have their neighbours in $[N(y) \cap N(z)] \cup N_2 \cup S_c \cup \{x\}$ and their neighbours in $[N(y) \cap N(z)] \cup N_2 \cup \{x\}$ form an independent set. Note that c' can be extended to a 3-colouring of G by 2-colourings of $G[S_c]$ on the colours that c' assigns to y and z, and these are the only possibilities. If $G[S_c]$ is connected, which can be tested in $O(n+m)$ time, then there are at most two such possibilities. We proceed by assuming that $G[S_c]$ is disconnected, and so $|S_c| \geq 2$.

We claim that all vertices of S_c have the same neighbours in N_2. Let us assume that v is an arbitrary vertex of N_2 that is adjacent to some vertex of S_c. Let S_v be the set of neighbours of v in S_c. By definition, we find that S_v is non-empty. As G is chair-free, we obtain that every vertex of S_v is adjacent to every vertex of $S_c \setminus S_v$ as otherwise $\{s_1, s_2, v, x, y\}$ would induce a chair for some possible vertices $s_1 \in S_v$ and $s_2 \in S_c \setminus S_v$. As $G[S_c]$ is disconnected, we find that $S_c \setminus S_v = \emptyset$, which completes the proof of our claim as v is arbitrarily chosen.

We can check if there is a vertex in $N(y) \cap N(z)$ in $O(n+m)$ time. First assume there is such a vertex, say w. As $\{s_1, s_2, w, x, y\}$ does not induce a chair for each two vertices s_1, s_2 of an independent set I of $G[S_c]$, we find that w is adjacent to all but at most one vertex of I. As $G[S_c]$ is bipartite, it follows that w has at least $|S_c|-2$ neighbours in S_c. For each $s \in N(w) \cap S_c$, we find $s \in A$ as $sxyw$ and $sxzw$ are 4-cycles. Note that $N(w) \cap S_c$ can be computed in $O(n+m)$ time. As $|N(w) \cap S_c| \geq |S_c|-2$, we find at most eight possibilities to extend c' to a 3-colouring of G by a 2-colouring of $G[S_c]$ in which one colour class contains all the vertices of $N(w) \cap S_c$. Hence, we may assume that $N(y) \cap N(z) = \emptyset$, and so every two vertices of S_c share the same neighbours in $V(G) \setminus S_c$.

If no vertex of N_2 has a neighbour in S_c, then x is a cut-vertex. In this case we find that G has an independent feedback vertex set of size at most k if and only if $G - S_c$ has an independent feedback vertex set (of size at most $k - |W|$, where W is the smallest possible colour class of a 2-colouring of $G[S_c]$.

We proceed by considering the situation where $v \in N_2$ has a neighbour in S_c. Recall that all vertices of S_c are adjacent to v. As xs_1vs_2 is a 4-cycle for any two vertices $s_1, s_2 \in S_c$, we find that A has size at least $|S_c|-1$. In other words, we aim for such a 2-colouring of $G[S_c]$ whose one colour class is of size at most 1. If S_c is not independent, we have at most two such possibilities, and each leads to a 3-colouring of G.

Now suppose that S_c is independent. We find that any two vertices of S_c have the same neighbours in G. Let us fix one vertex, say, s of S_c. As there is at most one vertex of S_c that is not in the independent feedback vertex set, we may assume that s is that vertex. We have four ways of colouring the vertices of S_c such that all vertices of $S_c \setminus \{s\}$ receive the same colour. □

Next we present an NP-completeness result for polyad-free graphs.

Theorem 5.4. *For the problems* Star 3-Colouring, Independent Odd Cycle Transversal *and* Acyclic 3-Colouring, *there exists a polyad H and integer d so that the problem remains* NP*-complete on H-free graphs of diameter d.*

Proof. We deploy an argument previously used in Chapter 2. There we recall the standard reduction from Not-All-Equal-3-Sat to 3-colouring.

- Add a vertex v_{x_i} for each literal x_i.
- Add an edge between each literal and its negation.
- Add a vertex z adjacent to every literal vertex.
- For each clause C_i add a triangle T_i with vertices $c_{i_1}, c_{i_2}, c_{i_3}$.
- Fix an arbitrary order of the literals of C_i, $x_{i_1}, x_{i_2}, x_{i_3}$ and add an edge $v_{x_{i_j}} c_{i_j}$.

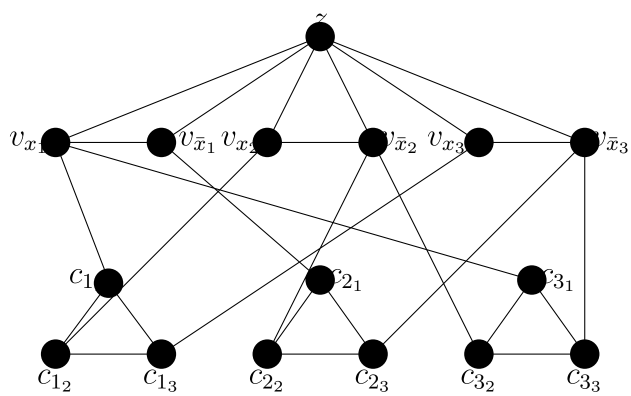

Figure 5.1: The standard reduction from Not-all-Equal 3-Sat to 3-Colouring on the instance $\phi = (x_1, x_2, x_3), (\bar{x}_1, \bar{x}_2, \bar{x}_3), (x_1, \bar{x}_2, \bar{x}_3)$.

In Chapter 2 we reduced from 3-colouring for $K^3_{1,4}$-free graphs of diameter at most 4. This result implies the NP-completeness of Independent Odd Cycle Transversal for the same class of graphs.

Suppose now we attempt to frame a similar argument for Acyclic 3-Colouring (respectively, Star 3-Colouring) by putting this construction through a process by which we map edges to bipartite graphs $K_{2,3}$ (respectively, $K_{2,2}$)

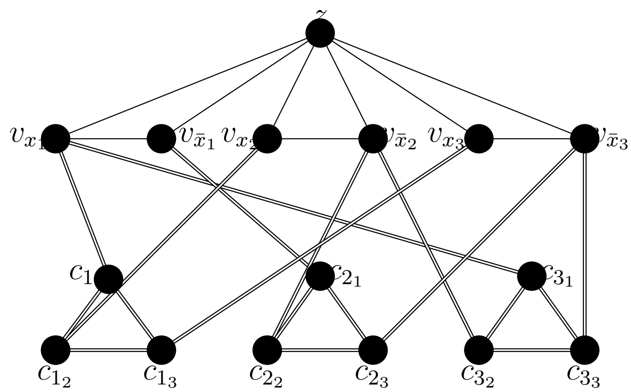

Figure 5.2: The reduction from NOT-ALL-EQUAL 3-SAT to ACYCLIC 3-COLOURING on the instance $\phi = (x_1, x_2, x_3), (\bar{x}_1, \bar{x}_2, \bar{x}_3), (x_1, \bar{x}_2, \bar{x}_3)$. Double lines are edges that are substitution instance of $K_{2,3}$ where the endpoints come from the partition of size 2.

which is the standard reduction from 3-COLOURING to these respective problems. That is, for ACYCLIC 3-COLOURING, we replace each edge u_1u_2 by three new vertices w_1, w_2, w_3 and edges u_iw_j for $i \in \{1, 2\}$ and $j \in \{1, 2, 3\}$, and for STAR 3-COLOURING, we replace each edge u_1u_2 by two new vertices w_1 and w_2 and edges u_iw_j for $i \in \{1, 2\}$ and $j \in \{1, 2\}$. Alas, using the top vertex z, which is the only vertex of unbounded degree, we can find arbitrarily large polyads in both cases.

However, for ACYCLIC 3-COLOURING, consider that we only substitute edges outside of the set induced by z and the vertices v_{x_i} by instances of $K_{2,3}$. This is drawn in Figure 5.2. We claim that ϕ is not-all equal satisfiable if and only if G is acyclically 3-colourable.

Given an acyclic 3-colouring of G, assume z is assigned colour 1. Then each vertex v_{x_i} is assigned either colour 2 or colour 3. The argument here concludes as it does in the reduction to 3-COLOURING, since in any acyclic 3-colouring of the gadget $K_{2,3}$ the vertices in the partition of size 2 must receive distinct colours.

If ϕ is satisfiable, then we can colour vertex z with colour 1, each true literal v_{x_i} with colour 2 and each false literal v_{x_i} with colour 3. The proof concludes as in the reduction to 3-Colouring, bearing in mind the reduction from 3-Colouring to Acyclic 3-Colouring except that we must argue there are no bichromatic cycles through z. This follows since all 4-paths emanating from z are coloured with three colours. Indeed, the only non-trivial case arises when we move from z to v_{x_i} to a vertex q in a clause triangle via a vertex p in the $K_{2,3}$ edge gadget. Thus there is a path $zv_{x_i}pq$ on 4 distinct vertices. Suppose the colour of p is the same as z, then the colour of q will be different from this and from v_{x_i}.

We claim that G has diameter at most 8. Indeed, there is path of length at most 4 from any vertex to z. We further claim that G is $K_{1,6}^8$-free, which follows from the fact that any induced $K_{1,6}^8$ in the graph would have to involve the vertex z and all paths of order 10 from z must involve two vertices adjacent to z. This concludes the argument for Acyclic 3-Colouring.

Our construction for Acyclic 3-Colouring fails for Star 3-Colouring. However, what we propose to do is to keep some of the construction (the bottom half of Figure 5.1) while we substitute the remainder (the top half of Figure 5.1). We reduce from a slightly different problem: Not-All-Equal($\leq 3, 2/3$)-Sat with positive literals asks the same question as Not-all-Equal 3-Sat but takes as input an instance ϕ that has a set of variables $X = \{x_1, \ldots, x_n\}$ and a set of literal clauses $\{C_1, \ldots, C_m\}$ over X with the following properties. Each C_i has either two or three literals, and these literals are all positive. Moreover, each literal occurs in at most three different clauses. This problem is well-known to be NP-complete, which follows from [87] and a folklore trick (see, for example, [42]). Given a CNF formula ϕ with the above properties, we construct a graph G as follows:

- Add a vertex v_{x_i} for each variable x_i.
- Add a vertex z adjacent to each vertex v_{x_i}.

- Add two new vertices z', z'' in a triangle with z.
- Add vertices $p^1_{x_i}, p^2_{x_i}, p^3_{x_i}$ for each instance of a variable x_i with edges from each of these to v_{x_i}.
- Add vertices $q^1_{x_i}, q^2_{x_i}, q^3_{x_i}$ for each instance of a variable x_i with edges from each $q^j_{x_i}$ to $p^j_{x_i}$ which are substitution instance of $K_{2,2}$ and the endpoints come from the same part.
- For each clause C_i add a triangle T_i with vertices $c_{i_1}, c_{i_2}, c_{i_3}$ where edges are substitution instance of $K_{2,2}$ and the endpoints come from the same part.
- Fix an arbitrary order of the literals of every C_i, $x_{i_1}, x_{i_2}, x_{i_3}$. Assign every pair (i, j) a vertex of $q^1_{x_{i_j}}, q^2_{x_{i_j}}, q^3_{x_{i_j}}$ and make this vertex adjacent to c_{i_j}, such that this assignment is injective. Let each of the new edges be a substitution instance of $K_{2,2}$, where the endpoints come from the same part.

We draw the special edges that are in fact built from instances $K_{2,2}$ with double lines in Figure 5.3. We claim ϕ is not-all-equal satisfiable if and only if G is star 3-colourable.

Given a star 3-colouring of G, assume z is assigned colour 1. Then each vertex v_{x_i} is assigned either colour 2 or colour 3. Now each of the vertices $p^1_{x_i}, p^2_{x_i}, p^3_{x_i}$ is assigned the same colour from $\{2, 3\}$ (this is enforced by the fact that $\{z', z''\}$ must be coloured $\{2, 3\}$ which forbids any possibility that colour 1 is used). Furthermore each of the vertices $q^1_{x_i}, q^2_{x_i}, q^3_{x_i}$ is assigned precisely the colour from $\{2, 3\}$ that $p^1_{x_i}, p^2_{x_i}, p^3_{x_i}$ was not assigned. The argument here concludes as it does in the reduction to 3-Colouring.

If ϕ is satisfiable, then we can colour vertex z with colour 1, the vertices $q^1_{x_i}, q^2_{x_i}, q^3_{x_i}$ of each true literal with colour 2 and the vertices $q^1_{x_i}, q^2_{x_i}, q^3_{x_i}$ of each false literal with colour 3. Then, since each clause has at least one true literal and at least one false literal, each triangle has neighbours in two different

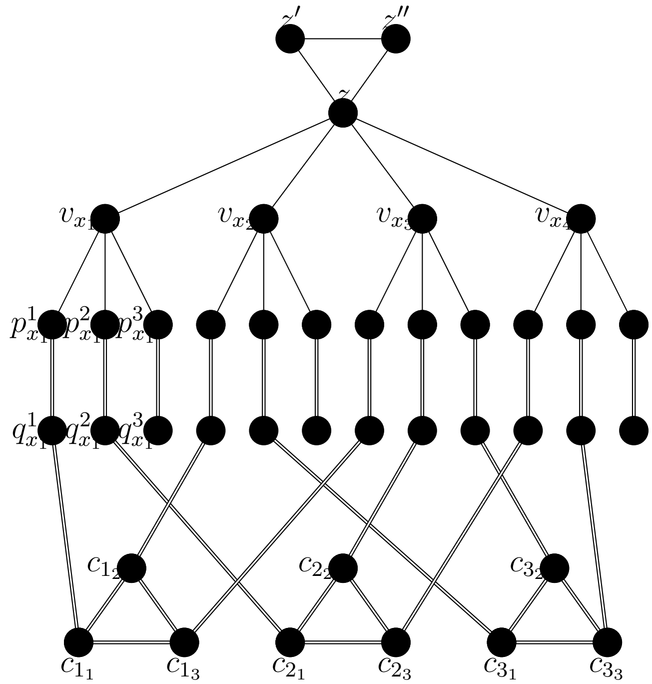

Figure 5.3: The reduction from NOT-ALL-EQUAL($\leq 3, 2/3$)-SAT to STAR 3-COLOURING on the instance $\phi = (x_1, x_2, x_3), (x_1, x_3, x_4), (x_2, x_3, x_4)$. Double lines are edges that are substitution instance of $K_{2,2}$ where the endpoints come from same partition.

colours. This implies that each triangle is 3-colourable. The argument here concludes as it does in the reduction to 3-COLOURING.

We claim that G has diameter at most 14. Indeed, every vertex has a path of length at most 7 from it to z. Moreover, all paths of order 16 from z must involve two vertices adjacent to z. Therefore G is $K^{16}_{1,6}$-free. This concludes the argument for STAR 3-COLOURING. $\square$

5.3 L(1,2)-labelling for Graphs of Bounded Diameter

Here we prove that $L(1,2)$-LABELLING is NP-complete for graphs of diameter at most 2. To do this we prove that it is NP-complete to decide whether a graph of diameter 2 contains a Hamiltonian Path, no edge of which is contained in a triangle.

We first present, as Lemmas 5.1 and 5.2, two hardness results for HAMILTONIAN CYCLE. We use Lemma 5.1 to prove Lemmas 5.2, and the latter to prove Lemma 5.3.

The *eccentricity* of a vertex u in a graph is the maximum distance of u to some other vertex of G. The *radius* of G is the minimum eccentricity of G.

Lemma 5.1. HAMILTONIAN CYCLE *is* NP*-complete even for connected bipartite graphs of minimum degree* 2 *and maximum degree* 5 *that have the following three additional properties:*

1. *for every two vertices x, y that belong to the same partition class and that have no common neighbour, there exists a vertex in the same partition class as x, y that is of distance greater than* 2 *from both x and y;*
2. *for every two non-adjacent vertices x, y that belong to different partition classes, either x has a neighbour of distance greater than* 2 *from y, or y has a neighbour of distance greater than* 2 *from x, and*
3. *no two vertices of degree* 2 *have the same neighbourhood.*

Proof. We reduce from HAMILTONIAN CYCLE, which is NP-complete even for graphs of maximum degree 3 [40]. As graphs of bounded maximum degree and bounded radius have constant size, the problem remains NP-complete if in addition we assume that the input graph $G = (V, E)$ of maximum degree 3 has radius at least 10.

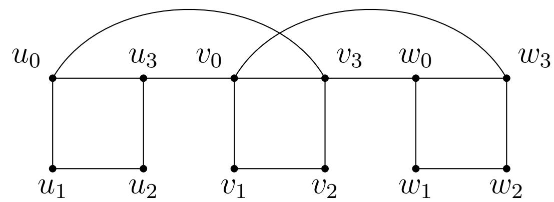

Figure 5.4: The graph G' from the proof of Lemma 5.1, when G is the 3-vertex path uvw.

We follow the construction used in [63]. That is, from G we construct a graph $G' = (V', E')$ as follows. We replace each $v \in V$ by a 4-cycle $v_0v_1v_2v_3$. Moreover, for each $uv \in E$, we do as follows. Let $u_0u_1u_2u_3$ and $v_0v_1v_2v_3$ be the 4-cycles that are associated with u and v, respectively. We add the two edges u_0v_3 and u_3v_0. This gives us the graph G'. See also Figure 5.4. It is readily seen that G has a Hamiltonian cycle if and only if G' has a Hamiltonian cycle. Moreover, G' is bipartite with one part
$A = \{v_i : i = 0, 2\}$ and the other $B = \{v_i : i = 1, 3\}$, and G' has minimum degree 2 and maximum degree 5; the latter holds as every vertex v_i has two more neighbours than v and v has degree at most 3 (as G has maximum degree 3). We now prove properties 1–3.

We first show Property 1. Let x and y be in the same partition class, say A, and assume that x and y have no common neighbour. If every vertex of A is of distance 2 from either x or y then, as G is connected, x and y are of distance at most 6 from each other. Consequently, the distance from x to any other vertex is at most $6 + 2 + 1 = 9$. Hence, G' has radius at most 9. As the distance between any two vertices u_i and v_i in G' is at least the distance between u and v in G, we find that G also has radius at most 9, a contradiction.

We now show Property 2. Let $x \in A$ and $y \in B$ be non-adjacent. Then $x = u_i$ for some $i \in \{0, 2\}$ and $y = v_j$ for some $j \in \{1, 3\}$ for vertices $u, v \in V$

with $u \neq v$. First suppose that $x = u_0$. If $y = v_1$, then u_1 is adjacent to u_0 and shares no neighbour with v_1, since $u \neq v$. If $y = v_3$ then v_2 is adjacent to v_3 and shares no neighbour with u_0, since $x = u_0$ and $y = v_3$ are non-adjacent. Now suppose that $x = u_2$. If $y = v_1$, then u_1 is adjacent to u_2 and shares no neighbour with v_1. Finally, if $x = u_2$ and $y = v_3$ then v_2 is adjacent to v_3 and shares no neighbour with u_2.

Finally, Property 3 holds since the set of vertices of degree 2 is $\{v_1, v_2 : v \in V\}$, and no pair of vertices from this set has the same neighbours. □

Lemma 5.2. Hamiltonian Path *is* NP*-complete even for connected bipartite graphs that satisfy Properties 1 and 2 of Lemma 5.1.*

Proof. We reduce from Hamiltonian Cycle, which is NP-complete even for the graphs $G' = (V', E')$ constructed in the proof of Lemma 5.1. We modify a given graph G' into a graph G'' as follows. We take some vertex x of degree 2 and add a new vertex x' with the same neighbourhood as x. We then add two further new vertices, x_1 and x_1' such that x_1 is pendant on x and x_1' is pendant on x'. See also Figure 5.5. We observe that G' has a Hamiltonian cycle if and only if G'' has a Hamiltonian path, which must start in u_1 and end in u_1'. As G' is bipartite, G'' is also bipartite. Hence, it remains to prove Properties 1 and 2.

We first show that Property 1 holds for G''. As Property 1 holds for G' by Lemma 5.1 and the three new vertices x', x_1, x_1' do not decrease the distance between any two vertices of G', we only need to consider pairs of vertices involving at least one of $\{x', x_1, x_1'\}$. Vertices x_1 and x_1' belong to the same partition class of G'' and have no common neighbour. Any non-neighbour z of x in G' is of distance greater than 2 from both x_1 and x_1', and we can choose z such that z belongs to the same partition class of G'' as x_1 and x_1'. Now consider one of x_1, x_1', say x_1, and a vertex y of G' that belongs to the same partition class as x_1 in G'', such that x_1 and y do not have a common neighbour. Then x'_1 is of distance greater than 2 from y in G'', and we can take x'_1. Vertices x and x' also belong to the same partition class of G'', but their neighbourhood is the same. Therefore, as Property 1 holds with respect to x in G', Property 1 also holds with respect to x' in G''.

We now show that Property 2 holds for G''. Again we need only to verify pairs involving at least one of $\{x', x_1, x'_1\}$. We first consider the pair (x', x_1); note that x' and x_1 are non-adjacent and belong to different partition classes of G''. We can take x'_1 as the desired vertex, as x'_1 is adjacent to x' but of distance greater than 2 from x_1 in G''. By symmetry, Property 2 holds for the pair (x, x'_1).

We now consider a pair (x', y) where $y \notin \{x_1, x'_1\}$ belongs to a different partition class of G'' than x' and is not adjacent to x'. As x and x' have the same neighbourhood in G'', we find that y and x are non-adjacent vertices in different partition classes as well. As Property 2 holds for G', there exists a vertex z that is a neighbour of one of $\{x, y\}$ but that is of distance greater than 2 from the other vertex of $\{x, y\}$. As the distance between two vertices of G' is the same in G'', we can take z as the desired vertex for the pair (x', y).

Finally, we consider a pair (x_1, y) or (x'_1, y), say (x_1, y) (by symmetry), where y is a non-neighbour of x_1 in G'' such that x_1 and y belong to different partition classes of G''. Note that y must be a vertex of G'. For contradiction, assume that every neighbour of y is of distance 2 from x_1 in G''. Then every neighbour of y in G'' is a neighbour of x. As y belongs to G', we find that y has degree at least 2 in G'. As x has degree 2 in G', this means that in G', both x and y have the same neighbourhood. The latter is a contradiction, as G' satisfies Property 3 of Lemma 5.2. We conclude that G'' has Property 2. $\square$

Lemma 5.3. *It is* NP*-complete to decide if a graph has a Hamiltonian path, no edge of which is contained in a triangle, even for graphs of diameter* 2.

Proof. We reduce from Hamiltonian Path, which is NP-complete even for

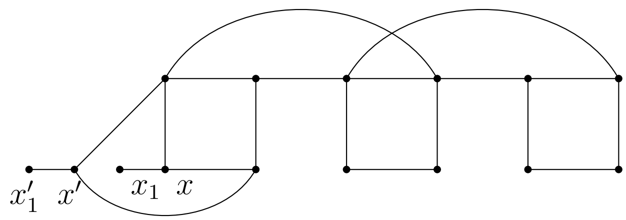

Figure 5.5: The graph G'' from the proof of Lemma 5.2.

the graphs $G'' = (V'', E''')$ constructed in the proof of Lemma 5.2. We modify a given graph G'' into a graph G^* by adding an edge between any two vertices u, v that belong to the same partition class and that are of distance greater than 2 from each other in G''. By our construction, the distance between any two vertices that belong to the same partition class of G'' is at most 2 in G^*. As G'' has Property 2, the distance between any two vertices in different partition classes of G'' is at most 2 in G^* as well. Hence, G^* has diameter at most 2.

It remains to prove that G'' has a Hamiltonian path if and only if G^* has a Hamiltonian path, no edge of which is contained in a triangle. For showing this it suffices to prove that for every edge e of G^*, it holds that e does not belong to a triangle in G^* if and only if e is an edge of G''.

First suppose that e is not an edge of G''. Say e is an edge between x and y, where x and y are two vertices of distance greater than 2 that belong to the same partition class of G''. As G'' has Property 1, there exists a vertex z that also belongs to the same partition class as x and y and that is of distance greater than 2 from both x and y. Hence, we have added the edges xz and yz as well, thus e belongs to a triangle in G^*.

Now suppose that e is an edge of G''. Let $e = xy$ for two vertices x and y (which belong to different bipartition classes of G''). For contradiction, assume that x and y are contained in a triangle xyz where z belongs to the same partition class as x, so we added the edge xz. Note that x and z have a common neighbour in G'', namely y. This means that their distance is not greater than 2 in G''. Hence, we would not have added the edge xz, a contradiction. $\square$

We can now prove our main result. For doing this, we show that an n-vertex graph G of diameter 2 has an $L(1,2)$-n-labelling if and only if G has a Hamiltonian path, no edge of which is contained in a triangle.

Theorem 5.5. *The* $L(1,2)$-LABELLING *problem is* NP*-complete even for graphs of diameter at most* 2*.*

Proof. Let G be an n-vertex graph of diameter 2. It suffices to prove that G has an $L(1,2)$-n-labelling if and only if G has a Hamiltonian path, no edge of which is contained in a triangle. Then, afterwards, we can apply Lemma 5.3.

First suppose that G has an $L(1,2)$-n-labelling c. Since G has diameter 2, any two non-adjacent vertices have a common neighbour. Hence, colours of non-adjacent vertices must differ by at least 2. Consequently, two vertices with consecutive colours must be adjacent. As colours of adjacent vertices differ by at least 1, we also find that no two vertices have the same colour. Consequently, every colour i with $1 \leq i \leq n$ is used. Therefore we have a Hamiltonian path $P = v_1 \dots v_n$ where v_i is the vertex with colour $c(v_i) = i$. No edge $v_i v_{i+1}$ is contained in a triangle since there can be no path of length 2 between v_i and v_{i+1}.

Now suppose that G contains a Hamiltonian path $P = v_1 \dots v_n$, no edge of which is contained in a triangle. The latter means that there is no path of length 2 between v_i and v_{i+1} for $i \in \{1, \dots, n-1\}$. Then we obtain an $L(1,2)$-n-labelling c by defining $c(v_i) = i$. $\square$

5.4 Conclusions

We now suggest two open problems based on the results in this chapter.

Open Problem 16. *Does there exist a polyad H and an integer d such that* Independent Feedback Vertex Set *and* Near-Bipartiteness *are* NP*-complete for H-free graphs of diameter at most d?*

Open Problem 17. *For the remaining four problems,* 3-Colouring, Independent Odd Cycle Transversal*,* Acyclic 3-Colouring *and* Star 3-Colouring*, can we narrow the gap between our* NP*-completeness and polynomial-time results for chair-free graphs?*

Chapter 6

Disjoint Paths and Connected Subgraphs

In this chapter we study two further problems, Disjoint Paths and Disjoint Connected Subgraphs. We first introduce the necessary definitions and terminology. In Section 6.3 we prove a complexity dichotomy for k-Disjoint Connected Subgraphs. In Section 6.4 we give an almost complete complexity classification for Disjoint Connected Subgraphs.

6.1 Terminology

A path from s to t in a graph G is an *s-t-path* of G, and s and t are called its *terminals*. Here we assume that s and t are distinct. Two pairs (s_1, t_1) and (s_2, t_2) are *disjoint* if $\{s_1, t_1\} \cap \{s_2, t_2\} = \emptyset$. In 1980, Shiloach [90] gave a polynomial-time algorithm for testing if a graph with disjoint terminal pairs (s_1, t_1) and (s_2, t_2) has vertex-disjoint paths P^1 and P^2 such that each P^i is an s_i-t_i path. This problem can be generalized as follows.

Disjoint Paths	
Instance:	A graph G and pairwise disjoint terminal pairs $(s_1, t_1) \dots, (s_k, t_k)$.
Question:	Does G have pairwise vertex-disjoint paths P^1,...,P^k such that P^i is an s_i-t_i path for $i \in \{1, \dots, k\}$?

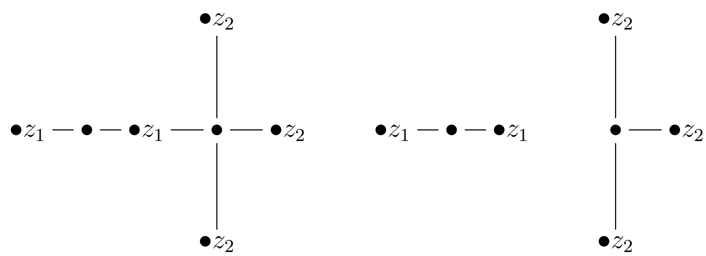

Figure 6.1: An example of a yes-instance (G, Z_1, Z_2) of (2-)Disjoint Connected Subgraphs (left) together with a solution (right).

Karp [57] proved that Disjoint Paths is NP-complete. If k is fixed, that is, not part of the input, then we denote the problem as k-Disjoint Paths. For every $k \geq 1$, Robertson and Seymour proved the following celebrated result.

Theorem 6.1 ([85])**.** *For all $k \geq 1$, k-*Disjoint Paths *is polynomial-time solvable.*

The running time in Theorem 6.1 is cubic. This was later improved to quadratic time by Kawarabayashi, Kobayashi and Reed [54].

As Disjoint Paths is NP-complete, it is natural to consider special graph classes. The Disjoint Paths problem is known to be NP-complete even for graph of clique-width at most 6 [46], split graphs [47], interval graphs [80] and line graphs. The latter result can be obtained by a straightforward reduction (see, for example, [46, 47]) from its edge variant, Edge Disjoint Paths, proven to be NP-complete by Even, Itai and Shamir [36]. On the positive side, Disjoint Paths is polynomial-time solvable for cographs, or equivalently, P_4-free graphs [46].

We can generalize the Disjoint Paths problem by considering terminal sets Z_i instead of terminal pairs (s_i, t_i). We write $G[S]$ for the subgraph of a graph $G = (V, E)$ induced by $S \subseteq V$, where S is *connected* if $G[S]$ is connected.

DISJOINT CONNECTED SUBGRAPHS	
Instance:	A graph G and pairwise disjoint terminal sets $Z_1, \ldots, Z_k$.
Question:	Does G have pairwise disjoint connected sets $S_1, \ldots, S_k$ such that $Z_i \subseteq S_i$ for $i \in \{1, \ldots, k\}$?

If k is fixed, then we write k-DISJOINT CONNECTED SUBGRAPHS. We refer to Figure 6.1 for a simple example of an instance (G, Z_1, Z_2) of 2-DISJOINT CONNECTED SUBGRAPHS. Robertson and Seymour [85] proved in fact that k-DISJOINT CONNECTED SUBGRAPHS is cubic-time solvable as long as $|Z_1| + \ldots + |Z_k|$ is fixed (this result implies Theorem 6.1). Otherwise, van 't Hof et al. [94] proved that already 2-DISJOINT CONNECTED SUBGRAPHS is NP-complete even if $|Z_1| = 2$ (and $|Z_2|$ may have arbitrarily large size). The same authors also proved that 2-DISJOINT CONNECTED SUBGRAPHS is NP-complete for split graphs. Afterwards, Gray et al. [43] proved that 2-DISJOINT CONNECTED SUBGRAPHS is NP-complete for planar graphs. Hence, Theorem 6.1 cannot be extended to hold for k-DISJOINT CONNECTED SUBGRAPHS.

We note that in recent years a number of exact algorithms were designed for k-DISJOINT CONNECTED SUBGRAPHS. Cygan et al. [30] gave an $O^*(1.933^n)$-time algorithm for the case $k = 2$ (see [84, 94] for faster exact algorithms for special graph classes). Telle and Villanger [92] improved this to time $O^*(1.7804^n)$. Recently, Agrawal et al. [1] gave an $O^*(1.88^n)$-time algorithm for the case $k = 3$. Moreover, the 2-DISJOINT CONNECTED SUBGRAPHS problem plays a crucial role in graph contractibility: a connected graph can be contracted to the 4-vertex path if and only if there exist two vertices u and v such that $(G - \{u, v\}, N(u), N(v))$ is a yes-instance of 2-DISJOINT CONNECTED SUBGRAPHS (see, e.g. [59, 94]).

6.2 Our Results

By combining some of the aforementioned known results with a number of new results, we prove the following two theorems in Sections 6.3 and 6.4, respectively. In particular, we generalize the polynomial-time result for DISJOINT PATHS on P_4-free graphs to hold even for DISJOINT CONNECTED SUBGRAPHS.

Theorem 6.2. *Let H be a graph. If $H \subseteq_i sP_1 + P_4$, then for every $k \geq 2$, k-*Disjoint Connected Subgraphs *on H-free graphs is polynomial-time solvable; otherwise even* 2-Disjoint Connected Subgraphs *is* NP-*complete.*

Theorem 6.3. *Let H be a graph not in $\{3P_1, 2P_1 + P_2, P_1 + P_3\}$. If $H \subseteq_i P_4$, then* Disjoint Connected Subgraphs *is polynomial-time solvable for H-free graphs; otherwise even* Disjoint Paths *is* NP-*complete.*

Theorem 6.2 completely classifies, for every $k \geq 2$, the complexity of k-Disjoint Connected Subgraphs on H-free graphs. Theorem 3.6 determines the complexity of Disjoint Paths and Disjoint Connected Subgraphs on H-free graphs for every graph H except if $H \in \{3P_1, 2P_1 + P_2, P_1 + P_3\}$. In Section 6.5 we reduce the number of open cases from six to *three* by showing some equivalencies.

In Section 6.6 we give some directions for future work. In particular we prove that both problems are polynomial-time solvable for co-bipartite graphs, which form a subclass of the class of $3P_1$-free graphs.

6.3 The Proof of Theorem 6.2

We consider k-Disjoint Connected Subgraphs for fixed k. First, we show a polynomial-time algorithm on H-free graphs when $H \subseteq_i sP_1 + P_4$ for some fixed $s \geq 0$. Then, we prove the hardness result.

For the algorithm, we need the following lemma for P_4-free graphs, or equivalently, cographs. This lemma is well known and follows immediately from the definition of a cograph [28]: in the construction of a connected cograph G, the last operation must be a join, so there exists cographs G_1 and G_2, such that G is obtained from adding an edge between every vertex of G_1 and every vertex of G_2. Hence, the spanning complete bipartite graph of G has non-empty partition classes $V(G_1)$ and $V(G_2)$.

Lemma 6.1. *Every connected P_4-free graph on at least two vertices has a spanning complete bipartite subgraph.*

We note that if two adjacent vertices will always appear in the same set of every solution $(S_1, \ldots, S_k)$ for an instance $(G, Z_1, \ldots, Z_k)$, then we may contract the edge between them at the start of any algorithm. In particular, we may contract any edge inside a terminal set. This takes linear time. Moreover, H-free graphs are readily seen (see e.g. [59]) to be closed under edge contraction if H is a linear forest. Hence, we can make the following observation.

Lemma 6.2. *For $k \geq 2$, from every instance of $(G, Z_1, \ldots, Z_k)$ of k-*Disjoint Connected Subgraphs *we can obtain in polynomial time an equivalent instance $(G', Z'_1, \ldots, Z'_k)$ such that every Z'_i is an independent set. Moreover, if G is H-free for some linear forest H, then G' is also H-free.*

We can now prove the following lemma.

Lemma 6.3. *Let H be a graph. If $H \subseteq_i sP_1 + P_4$, then for every $k \geq 1$, k-*Disjoint Connected Subgraphs *on H-free graphs is polynomial-time solvable.*

Proof. Let $H \subseteq_i sP_1 + P_4$ for some $s \geq 0$. Let $(G, Z_1, \ldots, Z_k)$ be an instance of k-Disjoint Connected Subgraphs, where G is an H-free graph. By Lemma 6.2, we may assume without loss of generality that G is connected and moreover that $Z_1, \ldots, Z_k$ are all independent sets.

We first analyze the structure of a solution $(S_1, \ldots, S_k)$ (if it exists). For $i \in \{1, \ldots, k\}$, we may assume that S_i is inclusion-wise minimal, meaning there is no $S'_i \subset S_i$ that contains Z_i and is connected. Consider a graph $G[S_i]$. Either $G[S_i]$ is P_4-free or $G[S_i]$ contains an induced $rP_1 + P_4$ for some $0 \leq r \leq s-1$. We will now show that in both cases, S_i is the (not necessarily disjoint) union of Z_i and a connected dominating set of $G[S_i]$ of constant size.

First suppose that $G[S_i]$ is P_4-free. As $G[S_i]$ is connected and Z_i is independent, we apply Lemma 6.1 to find that $S_i \setminus Z_i$ contains a vertex u that is adjacent to every vertex of Z_i. Hence, by minimality, $S_i = Z_i \cup \{u\}$ and $\{u\}$ is a connected dominating set of $G[S_i]$ of size 1.

Now suppose that $G[S_i]$ has an induced $rP_1 + P_4$ for some $r \geq 0$, where we choose r to be maximum. Note that $r \leq s-1$. Let W be the vertex set of the induced rP_1+P_4. Then, as r is maximum, W dominates $G[S_i]$. Note that $G[W]$ has $r+1 \leq s$ connected components. Then, as $G[S_i]$ is connected and W is a dominating set of $G[S_i]$ of size $r+4 \leq s+3$, it follows from folklore arguments (see e.g. [93, Prop. 6.3.24]) that $G[S_i]$ has a connected dominating set W' of size at most $3s+1$. Moreover, by minimality, $S_i = Z_i \cup W'$.

Hence, in both cases we find that S_i is the union of Z_i and a connected dominating set of $G[S_i]$ of size at most $t = 3s+1$; note that t is a constant, as s is a constant.

Our algorithm now does as follows. We consider all options of choosing a connected dominating set of each $G[S_i]$, which from the above has size at most t. As soon as one of the guesses makes every Z_i connected, we stop and return the solution. The total number of options is $O(n^{tk})$, which is polynomial as k and t are fixed. Moreover, checking the connectivity condition can be done in polynomial time. Hence, the total running time of the algorithm is polynomial. □

The proof our next result is inspired by the aforementioned NP-completeness result of [94] for instances (G, Z_1, Z_2) where $|Z_1| = 2$ but G is a general graph.

Lemma 6.4. *The* 2-Disjoint Connected Subgraphs *problem is* NP-*complete even on instances* (G, Z_1, Z_2) *where* $|Z_1| = 2$ *and* G *is a line graph.*

Proof. Note that the problem is in NP. We reduce from 3-SAT.
Let $\phi = \phi(x_1, \ldots, x_n)$ be an instance of 3-SAT with clauses $C_1, \ldots, C_m$. We construct a corresponding graph $G = (V, E)$ as follows. We start with two disjoint paths P and $\bar{P}$ on vertices p_i, x_i, q_i and $\bar{p}_i, \bar{x}_i, \bar{q}_i$, respectively, where $x_i, \bar{x}_i$ correspond to the positive and negative literals in ϕ, respectively. To be more precise, we define:

$$P = p_1x_1q_1p_2x_2q_2, \ldots, p_nx_nq_n, \text{ and } \overline{P} = \bar{p}_1\bar{x}_1\bar{q}_1 \ldots \bar{p}_n\bar{x}_n\bar{q}_n$$

We add the two edges $e = p_1\bar{p}_1$, and $f = q_n\bar{q}_n$. For $i = 1, \ldots, n-1$, we also add edges $q_i\bar{p}_{i+1}$ and $\bar{q}_ip_{i+1}$. We now replace each x_i by vertices

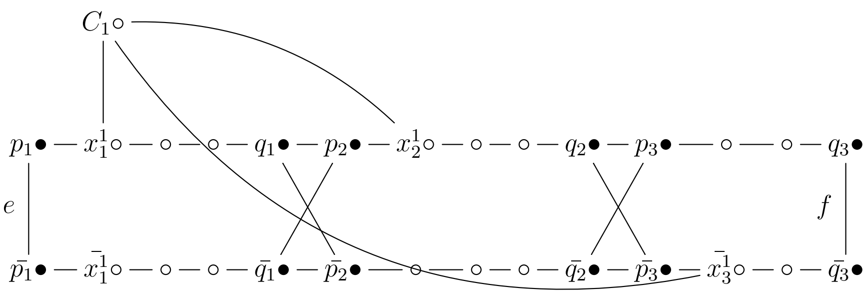

Figure 6.2: The construction described with edges added for the clause $C_1 = (x_1 \vee x_2 \vee \bar{x_3})$.

$x_i^{j_1}, x_i^{j_2}, \ldots x_i^{j_r}$, where $j_1, \ldots, j_r$ are the indices of the clauses C_j that contain x_i. That is, we replace the subpath $p_i x_i q_i$ of P by the path $p_i x_i^{j_1} x_i^{j_2}, \ldots x_i^{j_r}, q_i$. We do the same path replacement operation on $\bar{P}$ with respect to every $\bar{x}_i$. Finally, we add every clause C_j as a vertex and add an edge between C_j and x_i^j if and only if $x_i \in C_j$, and between C_j and $\bar{x}_i^j$ if and only if $\bar{x}_j \in C_j$. This completes the description of $G = (V, E)$. We refer to Figure 6.2 for an illustration of our construction.

We now focus on the line graph $L = L(G)$ of G.
Let $Z_1 = \{e, f\} \subseteq E = V(L)$ and let Z_2 consist of all vertices of L that correspond to edges in G that are incident to some C_j. Note that Z_1 and Z_2 are disjoint. Moreover, each clause C_j corresponds to a clique of size at most 3 in L, which we call the clause clique of C_j. We claim that ϕ is satisfiable if and only if the instance (L, Z_1, Z_2) of 2-Disjoint Connected Subgraphs is a yes-instance.

First suppose that ϕ is satisfiable. Let τ be a satisfying truth assignment for ϕ. In G, we let P^1 denote the unique path whose first edge is e and whose last edge is f and that passes through all $x_i^j \in V$ if $x_i = 0$ and through all $\bar{x}_i^j$ if $x_i = 1$. In L we let S_1 consist of all vertices of $L(P^1)$; note that $Z_1 = \{e, f\}$ is contained in S_1 and that S_1 is connected. We let P^2 denote the "complementary" path in G whose first edge is e and whose last edge is

f but that passes through all x_i^j if and only if P^1 passes through all $\bar{x}_i^j$, and conversely ($i = 1, \ldots, n$). In L, we put all vertices of $L(P^2)$, except e and f, together with all vertices of Z_2 in S_2. As τ satisfies ϕ, some vertex of each clause clique is adjacent to a vertex of P^2. Hence, as P^2 is a path, S_2 is connected and we found a solution for (L, Z_1, Z_2).

Now suppose that (L, Z_1, Z_2) is a yes-instance of 2-Disjoint Connected Subgraphs. Then $V(L)$ can be partitioned into two vertex-disjoint connected sets S_1 and S_2 such that $Z_1 \subseteq S_1$ and $Z_2 \subseteq S_2$. In particular, $L[S_1]$ contains a path P^1 from e to f. In fact, we may assume that $S_1 = V(P^1)$, as we can move every other vertex of S_1 (if they exist) to S_2 without disconnecting S_2.

Note that P^1 corresponds to a connected subgraph that contains the adjacent vertices p_1 and $\bar{p}_1$ as well as the adjacent vertices q_n and $\bar{q}_n$. Hence, we can modify P^1 into a path Q in G that starts in p_1 or $\bar{p}_1$ and that ends in q_n or $\bar{q}_n$. Note that Q contains no edge incident to a clause vertex C_j, as those edges correspond to vertices in L that belong to Z_2. Hence, by construction, Q "moves from left to right", that is, Q cannot pass through both some x_i^j and $\bar{x}_i^j$ (as then Q needs to pass through either x_i^j or $\bar{x}_i^j$ again implying that Q is not a path).

Moreover, if Q passes through some x_i^j, then Q must pass through all vertices $x_i^{j_h}$. Similarly, if Q passes through some $\bar{x}_i^j$, then Q must pass through all vertices $\bar{x}_i^{j_h}$. As Q connects the edges $p_1\bar{p}_1$ and $q_n\bar{q}_n$, we conclude that Q must pass, for $i = 1, \ldots, n$, through either every $x_i^{j_h}$ or through every $\bar{x}_i^{j_h}$. Thus we may define a truth assignment τ by setting

$$x_i = \begin{cases} 1 \text{ if } Q \text{ passes through all } \bar{x}_i^j \\ 0 \text{ if } Q \text{ passes through all } x_i^j. \end{cases}$$

We claim that τ satisfies ϕ. For contradiction, assume some clause C_j is not satisfied. Then Q passes through all its literals. However, then in S_2, the vertices of Z_2 that correspond to edges incident to C_j are not connected to other vertices of Z_2, a contradiction. This completes the proof of the lemma. $\square$

A straightforward modification of the reduction of Lemma 6.5 gives us Lemma 6.6. We can also obtain Lemma 6.6 by subdividing the graph G in the proof of Lemma 6.4 twice (to get a bipartite graph) or p times (to get a graph of girth at least p).

Lemma 6.5 ([94]). 2-Disjoint Connected Subgraphs *is* NP*-complete for split graphs, or equivalently,* $(2P_2, C_4, C_5)$*-free graphs.*

Lemma 6.6. 2-Disjoint Connected Subgraphs *is* NP*-complete for bipartite graphs and for graphs of girth at least* p*, for every integer* $p \geq 3$.

We are now ready to prove Theorem 6.2.

Theorem 6.2 (restated) *Let* H *be a graph. If* $H \subseteq_i sP_1 + P_4$*, then for every* $k \geq 1$, k-Disjoint Connected Subgraphs *on* H*-free graphs is polynomial-time solvable; otherwise even* 2-Disjoint Connected Subgraphs *is* NP*-complete.*

Proof. If H contains an induced cycle C_s for some $s \geq 3$, then we apply Lemma 6.6 by setting $p = s+1$. Now assume that H contains no cycle, that is, H is a forest. If H has a vertex of degree at least 3, then H is a superclass of the class of claw-free graphs, which in turn contains all line graphs. Hence, we can apply Lemma 6.4. In the remaining case H is a linear forest. If H contains an induced $2P_2$, we apply Lemma 6.5. Otherwise H is an induced subgraph of $sP_1 + P_4$ for some $s \geq 0$ and we apply Lemma 6.3. $\square$

6.4 The Proof of Theorem 6.3

We first prove the following result, which generalizes the corresponding result of Disjoint Paths for P_4-free graphs due to Gurski and Wanke [46]. We show that we can use the same modification to a matching problem in a bipartite graph.

Lemma 6.7. Disjoint Connected Subgraphs *is polynomial-time solvable for P_4-free graphs.*

Proof. For some integer $k \geq 2$, let $(G, Z_1, \ldots, Z_k)$ be an instance of Disjoint Connected Subgraphs where G is a P_4-free graph. By Lemma 6.2 we may assume that every Z_i is an independent set. Now suppose that $(G, Z_1, \ldots, Z_k)$ has a solution $(S_1, \ldots, S_k)$. Then $G[S_i]$ is a connected P_4-free graph. Hence, by Lemma 6.1, $G[S_i]$ has a spanning complete bipartite graph on non-empty partition classes A_i and B_i. As every Z_i is an independent set, it follows that either $Z_i \subseteq A_i$ or $Z_i \subseteq B_i$. If $Z_i \subseteq A_i$, then every vertex of B_i is adjacent to every vertex of Z_i. Similarly, if $Z_i \subseteq B_i$, then every vertex of A_i is adjacent to every vertex of Z_i. We conclude that in every set S_i, there exists a vertex y_i such that $Z_i \cup \{y_i\}$ is connected.

The latter enables us to construct a bipartite graph $G' = (X \cup Y, E')$ where X contains vertices $x_1, \ldots, x_k$ corresponding to the set $Z_1, \ldots, Z_k$ and Y is the set of non-terminal vertices of G. We add an edge between $x_i \in X$ and $y \in Y$ if and only if y is adjacent to every vertex of Z_i. Then $(G, Z_1 \ldots Z_k)$ is a yes-instance of Disjoint Connected Subgraphs if and only if G' contains a matching of size k. It remains to observe that we can find a maximum matching in polynomial time, for example, by using the Hopcroft-Karp algorithm for bipartite graphs [51].

□

The first lemma of a series of four is obtained by a straightforward reduction from the Edge Disjoint Paths problem (see, e.g. [46, 47]), which was proven to be NP-complete by Even, Itai and Shamir [36]. The second lemma follows from the observation that an edge subdivision of the graph G in an instance of Disjoint Paths results in an equivalent instance of Disjoint Paths; we apply this operation a sufficiently large number of times to obtain a graph of large girth. The third lemma is due to Heggernes et al. [47]. We modify their construction to prove the fourth lemma.

Lemma 6.8. Disjoint Paths *is* NP*-complete for line graphs.*

Lemma 6.9. *For every $g \geq 3$,* Disjoint Paths *is* NP*-complete for graphs of girth at least g.*

Lemma 6.10 ([47])**.** Disjoint Paths *is* NP*-complete for split graphs, or equivalently,* $(C_4, C_5, 2P_2)$*-free graphs.*

Lemma 6.11. Disjoint Paths *is* NP*-complete for* $(4P_1, P_1+P_4)$*-free graphs.*

Proof. We reduce from Disjoint Paths on split graphs, which is NP-complete by Lemma 6.10. By inspection of this result (see [47, Theorem 3]), we note that the instances $(G, \{(s_1, t_1), \ldots, (s_k, t_k)\})$ have the following property: the split graph G has a split decomposition (C, I), where C is a clique, I an independent set, C and I are disjoint, and $C \cup I = V(G)$, such that $I = \{s_1, \ldots, s_k, t_1, \ldots, t_k\}$. Now let G' be obtained from G by, for each terminal s_i, adding edges to s_j and t_j for all $j \neq i$. Then consider the instance $(G', \{(s_1, t_1), \ldots, (s_k, t_k)\})$.

We note that $G'[C]$ is still a complete graph, while $G'[I]$ is a complete graph minus a matching. It is immediate that G' is $4P_1$-free. Moreover, any induced subgraph H of G' that is isomorphic to P_4 must contain at least two vertices of I and at least one vertex of C. If H contains two vertices of C, then as $G'[C]$ is a clique, H contains two non-adjacent vertices in I. Similarly, if H contains one vertex of C (and thus three vertices of I), then H contains two non-adjacent vertices in I. Since C is a clique in G' and every (other) vertex of I is adjacent in G' to any pair of non-adjacent vertices of I, it follows that G' is $P_1 + P_4$-free as well.

We claim that $(G, \{(s_1, t_1), \ldots, (s_k, t_k)\})$ is a yes-instance if and only if $(G', \{(s_1, t_1), \ldots, (s_k, t_k)\})$ is a yes-instance. This is because the edges that were added to G to obtain G' are only between terminal vertices of different pairs. These edges cannot be used by any solution of Disjoint Paths for $(G', \{(s_1, t_1), \ldots, (s_k, t_k)\})$, and thus the feasibility of the instance is not affected by the addition of these edges. $\square$

We are now ready to prove Theorem 6.3.

Theorem 6.3 (restated) *Let H be a graph not in $\{3P_1, 2P_1 + P_2, P_1 + P_3\}$. If $H \subseteq_i P_4$, then* Disjoint Connected Subgraphs *is polynomial-time solvable for H-free graphs; otherwise even* Disjoint Paths *is* NP*-complete.*

Proof. First suppose that H contains a cycle C_r for some $r \geq 3$. Then Disjoint Paths is NP-complete for the class of H-free graphs, as Disjoint Paths is NP-complete on the subclass consisting of graphs of girth $r + 1$ by Lemma 6.9. Now suppose that H contains no cycle, that is, H is a forest. If H contains a vertex of degree at least 3, then the class of H-free graphs contains the class of claw-free graphs, which in turn contains the class of line graphs. Hence, we can apply Lemma 6.8. It remains to consider the case where H is a forest with no vertices of degree at least 3, that is, when H is a linear forest.

If H contains four connected components, then the class of H-free graphs contains the class of $4P_1$-free graphs, and we can use Lemma 6.11. If H contains an induced P_5 or two connected components that each have at least one edge, then H contains the class of $2P_2$-free graphs, and we can use Lemma 6.10. If H contains two connected components, one of which has at least four vertices, then H contains the class of $(P_1 + P_4)$-free graphs, and we can use Lemma 6.11 again. As $H \notin \{3P_1, 2P_1 + P_2, P_1 + P_3\}$, this means that in the remaining case H is an induced subgraph of P_4. In that case even Disjoint Connected Subgraphs is polynomial-time solvable on H-free graphs, due to Lemma 6.7. $\square$

6.5 Reducing the Number of Open Cases to Three

Theorem 6.3 shows that we have the same three open cases for Disjoint Paths and Disjoint Connected Subgraphs, namely when $H \in \{3P_1, P_1 + P_3, 2P_1 + P_2\}$. We show that instead of six open cases, we have in fact only three.

Proposition 1. Disjoint Paths *and* Disjoint Connected Subgraphs *are equivalent for* $3P_1$*-free graphs.*

Proof. Every instance of Disjoint Paths is an instance of Disjoint Connected Subgraphs. Let $(G, Z_1, \ldots, Z_k)$ be an instance of Disjoint Connected Subgraphs where G is a $3P_1$-free graph. By Lemma 6.2 we may assume that each Z_i is an independent set. Then, as G is $3P_1$-free, each Z_i has size at most 2. So we obtained an instance of Disjoint Paths. □

Proposition 2. Disjoint Paths *on* $(P_1 + P_3)$*-free graphs and* Disjoint Connected Subgraphs *on* $(P_1 + P_3)$*-free graphs are polynomially equivalent to* Disjoint Paths *on* $3P_1$*-free graphs.*

Proof. We prove that we can solve Disjoint Connected Subgraphs in polynomial time on $(P_1 + P_3)$-free graphs if we have a polynomial-time algorithm for Disjoint Paths on $3P_1$-free graphs. Showing this suffices to prove the theorem, as Disjoint Paths is a special case of Disjoint Connected Subgraphs and $3P_1$-free graphs form a subclass of $(P_1 + P_3)$-free graphs.

Let $(G, Z_1, \ldots, Z_k)$ be an instance of Disjoint Connected Subgraphs, where G is a $(P_1 + P_3)$-free graph. Olariu [82] proved that every connected $\overline{P_1 + P_3}$-free graph is either triangle-free or complete multipartite. Hence, the vertex set of G can be partitioned into sets $D_1, \ldots, D_p$ for some $p \geq 1$ such that

- every $G[D_i]$ is $3P_1$-free or the disjoint union of complete graphs, and
- for every i, j with $i \neq j$, every vertex of D_i is adjacent to every vertex of D_j.

Using this structural characterization, we first argue that we may assume that each Z_i has size 2, making the problem an instance of Disjoint Paths. Then we show that we can either solve the instance outright or can alter G to be $3P_1$-free.

First, we argue about the size of each Z_i. By Lemma 6.2 we may assume that every Z_i is an independent set and is thus contained in the same set D_j. If $G[D_j]$ is $3P_1$-free, then this implies that any Z_i that is contained in D_j has size 2. If $G[D_j]$ is a disjoint union of complete graphs, then each vertex of a Z_i that is contained in D_j belongs to a different connected component of D_j and $Z_i \cup \{v\}$ is connected for every vertex $v \notin D_j$. As at least one vertex $v \notin D_j$ is needed to make such a set Z_i connected, we may therefore assume that for a solution $(S_1, \ldots, S_k)$ (if it exists), $S_i = Z_i \cup \{v\}$ for some $v \notin D_j$. The latter implies that we may assume without loss of generality that every such Z_i has size 2 as well.

If $p = 1$, then each connected component of G is $3P_1$-free, and we are done. Hence, we assume that $p \geq 2$. In fact, since any two distinct sets D_i and D_j are complete to each other, the union of any two $3P_1$-free graphs induces a $3P_1$-free graph. Therefore we may assume without loss of generality that only $G[D_1]$ might be $3P_1$-free, whereas $G[D_2], \ldots, G[D_p]$ are disjoint unions of complete graphs.

Recall that $Z_i = \{s_i, t_i\}$ for every $i \in \{1, \ldots, k\}$ and we search for a solution $(P^1, \ldots, P^k)$ where each P^i is a path from s_i to t_i. First suppose s_i and t_i belong to D_1. Then P^i has length 2 or 3 and in the latter case, $V(P^i) \subseteq D_1$. Now suppose that s_i and t_i belong to D_h for some $h \in \{2, \ldots, k\}$. Then P^i has length exactly 2, and moreover, the middle (non-terminal) vertex of P^i does not belong to D_h.

We will now check if there is a solution $(P^1, \ldots, P^k)$ such that every P^i has length exactly 2. We call such a solution to be of *type 1.* In a solution of type 1, every $P^i = s_i u t_i$ for some non-terminal vertex u of G. If s_i and t_i belong to D_h for some $h \in \{2, \ldots, p\}$, then $u \in D_j$ for some $j \neq i$. If s_i and t_i belong to D_1, then $u \in D_j$ for some $j \neq 1$ but also $u \in D_1$ is possible, namely when u is adjacent to both s_i and t_i.

Verifying the existence of a type 1 solution is equivalent to finding a perfect matching in a bipartite graph $G' = A \cup B$ that is defined as follows.

The set A consists of one vertex v_i for each pair $\{s_i, t_i\}$. The set B consists of all non-terminal vertices u of G. For $\{s_i, t_i\} \subseteq D_1$, there exists an edge between u and v_i in G' if and only if in G it holds that $u \in D_h$ for some $h \in \{2, \ldots, p\}$ or $u \in D_1$ and u is adjacent to both s_i and t_i. For $\{s_i, t_i\} \subseteq D_h$ with $h \in \{2, \ldots, p\}$, there exists an edge between u and v_i in G' if and only if in G it holds that $u \in D_j$ for some $j \in \{1, \ldots, p\}$ with $h \neq j$. We can find a perfect matching in G' in polynomial time by using the Hopcroft-Karp algorithm for bipartite graphs [51].

Suppose that we find that $(G, \{s_1, t_1\}, \ldots, \{s_k, t_k\})$ has no solution of type 1. As a solution can be assumed to be of type 1 if $G[D_1]$ is the disjoint union of complete graphs, we find that $G[D_1]$ is not of this form. Hence, $G[D_1]$ is $3P_1$-free. Recall that $G[D_j]$ is the disjoint union of complete graphs for $2 \leq i \leq p$. It remains to check if there is a solution that is of *type 2* meaning a solution $(P^1, \ldots, P^k)$ in which at least one P^i, whose vertices all belong to D_1, has length 3.

To find a type 2 solution (if it exists) we construct the following graph G^*. We let $V(G^*) = A_1 \cup A_2 \cup B_1 \cup B_2$, where

- A_1 consists of all terminal vertices from D_1;
- A_2 consists of all non-terminal vertices from D_1;
- B_1 consists of all terminal vertices from $D_2 \cup \cdots \cup D_p$; and
- B_2 consists of all non-terminal vertices from $D_2 \cup \cdots \cup D_p$.

Note that $V(G^*) = V(G)$. To obtain $E(G^*)$ from $E(G)$ we add some edges (if they do not exist in G already) and also delete some edges (if these existed in G):

(i) for each $\{s_i, t_i\} \subseteq B_1$, add all edges between s_i and vertices of B_2, and delete any edges between t_i and vertices of B_2;

(ii) add an edge between every two terminal vertices in B_1 that belong to different terminal pairs; and

(iii) add an edge between every two vertices of B_2.

We note that $G^*[D_1]$ is the same graph as $G[D_1]$ and thus $G^*[D_1]$ is $3P_1$-free. Moreover, $G^*[B_1 \cup B_2]$ is $3P_1$-free by part (i) of the construction. Hence, as there exists an edge between every vertex of $A_1 \cup A_2$ and every vertex of $B_1 \cup B_2$ in G and thus also in G^*, this means that G^* is $3P_1$-free. It remains to prove that $(G, \{s_1, t_1\}, \ldots, \{s_k, t_k\})$ and $(G^*, \{s_1, t_1\}, \ldots, \{s_k, t_k\})$ are equivalent instances.

First suppose that $(G, \{s_1, t_1\}, \ldots, \{s_k, t_k\})$ has a solution $(P^1, \ldots, P^k)$. Assume that the number of paths of length 3 in this solution is minimum over all solutions for $(G, \{s_1, t_1\}, \ldots, \{s_k, t_k\})$. We note that $(P^1, \ldots, P^k)$ is a solution for $(G^*, \{s_1, t_1\}, \ldots, \{s_k, t_k\})$ unless there exists some P^i that contains an edge of $E(G) \setminus E(G^*)$. Suppose this is indeed the case. As $G^*[D_1] = G[D_1]$ and every edge between a vertex of $A_1 \cup A_2$ and a vertex of $B_1 \cup B_2$ also exists in G^*, we find that the paths connecting terminals from pairs in D_1 are paths in G^*. Hence, s_i and t_i belong to D_h for some $h \in \{2, \ldots, p\}$ and thus $P^i = s_i u t_i$ where u is a vertex of D_j for some $j \in \{2, \ldots, p\}$ with $j \neq h$.

As we already found that $(G, \{s_1, t_1\}, \ldots, \{s_k, t_k\})$ has no type 1 solution, there is at least one $P^{i'}$ with length 3, so $P^{i'} = s_{i'} v v' t_{i'}$ is in $G[D_1]$. However, we can now obtain another solution for $(G, \{s_1, t_1\}, \ldots, \{s_k, t_k\})$ by changing P^i into $s_i v t_i$ and $P^{i'}$ into $s_{i'} u t_{i'}$, a contradiction, as the number of paths of length 3 in $(P^1, \ldots, P^k)$ was minimum. We conclude that every P^i only contains edges from $E(G) \cap E(G^*)$, and thus $(P^1, \ldots, P^k)$ is a solution for $(G^*, \{s_1, t_1\}, \ldots, \{s_k, t_k\})$.

Now suppose that $(G^*, \{s_1, t_1\}, \ldots, \{s_k, t_k\})$ has a solution $(P^1, \ldots, P^k)$. Consider a path P^i. First suppose that s_i and t_i both belong to B_1. Then we may assume without loss of generality that $P^i = s_i u t_i$ for some $u \in A_2$.

As B_1 only contains terminals from pairs in $D_2 \cup \ldots \cup D_p$, the latter implies that P^i is a path in G as well. Now suppose that s_i and t_i both belong to A_1. Then we may assume without loss of generality that $P^i = s_i u t_i$ for some non-terminal vertex of $V(G) = V(G^*)$ or $P^i = s_i u u' t_i$ for two vertices u, u' in $A_2 \subseteq D_1$. Hence, P^i is a path in G as well. We conclude that $(P^1, \ldots, P^k)$ is a solution for $(G, \{s_1, t_1\}, \ldots, \{s_k, t_k\})$. This completes our proof. □

The three open cases seem challenging. We were able to prove the following positive result for a subclass of $3P_1$-free graphs, namely cobipartite graphs, or equivalently, $(3P_1, C_5, \overline{C_7}, \overline{C_9}, \ldots)$-free graphs.

Theorem 6.4. Disjoint Paths *is polynomial-time solvable for cobipartite graphs.*

Proof. Let $G = (A \cup B, E)$, with cliques A and B, be the given cobipartite graph. If s_i and t_i are adjacent in G, then use the direct edge between them as the path P^i. We can then reduce the instance by removing s_i and t_i. We now assume the instance has thus been reduced and (by abuse of notation) all terminal pairs are nonadjacent in G.

We now construct a bipartite graph G' by removing each edge within the cliques A and B as well as any edge $s_i t_j$ both of whose endpoints are terminals. We then obtain a new graph G'' by deleting each terminal vertex and adding for each terminal pair (s_i, t_i), a new vertex x_i whose neighbourhood is the union of the neighbourhoods of s_i and t_i in G'. We claim that G contains the required k disjoint paths $P^1 \ldots P^k$ if and only if G'' contains a matching of size at least k. We can check the latter in polynomial time by using the Hopcroft-Karp algorithm for bipartite graphs [51].

We first assume that G contains the disjoint paths $P^1 \ldots P^k$. Note that, since G is $3P_1$-free, we may assume each path has length at most 3. A matching M of size k is obtained as follows. For each $i = 1 \ldots k$, if P^i has length 2 we add the edge $x_i v_i$ to M where v_i is the interior vertex of P^i. If P^i has length 3 then we add its interior edge $u_i v_i$ to M.

Next assume G'' contains a matching M of size k. For each edge of M which includes a vertex x_i corresponding to a terminal pair (s_i, t_i) we set P^i to be $s_i v_i t_i$ where v_i is the vertex matched to x_i. Note that any edge uv in G which contains no terminal vertex and has one endpoint in each of A and B lies on a path of length 3 between any two terminal vertices. Therefore, for each i such that the vertex x_i is not matched in M, we can choose a distinct edge $u_i v_i$ in M to obtain the path $s_i u_i v_i t_i$ in G. $\square$

6.6 Conclusions

We first gave a dichotomy for k-Disjoint Connected Subgraphs in Theorem 6.2: for every k, the problem is polynomial-time solvable on H-free graphs if $H \subseteq_i sP_1 + P_4$ for some $s \geq 0$ and otherwise it is NP-complete even for $k = 2$. Two vertices u and v are a *P_4-suitable pair* if $(G - \{u, v\}, N(u), N(v))$ is a yes-instance of 2-Disjoint Connected Subgraphs. Recall that a graph G can be contracted to P_4 if and only if G has a P_4-suitable pair. Deciding if a pair $\{u, v\}$ is a suitable pair is polynomial-time solvable for H-free graphs if H is an induced subgraph of P_2+P_4, $P_1+P_2+P_3$, $P_1 + P_5$ or $sP_1 + P_4$ for some $s \geq 0$; otherwise it is NP-complete [59]. Hence, we conclude from our new result that the presence of the two vertices u and v that are connected to the sets $Z_1 = N(u)$ and $Z_2 = N(v)$, respectively, yield exactly three additional polynomial-time solvable cases.

We also classified, in Theorem 6.3, the complexity of Disjoint Paths and Disjoint Connected Subgraphs for H-free graphs. Due to Propositions 1 and 2, there are three non-equivalent open cases left and we ask the following:

Open Problem 18. *Determine the computational complexity of* Disjoint Paths *on H-free graph for $H \in \{3P_1, 2P_1 + P_2\}$ and the computational complexity of* Disjoint Connected Subgraphs *on H-free graphs for $H = 2P_1 + P_2$.*

Bibliography

[1] A. Agrawal, F. V. Fomin, D. Lokshtanov, S. Saurabh, and P. Tale. Path contraction faster than 2^n. *SIAM Journal on Discrete Mathematics*, 34:1302–1325, 2020.

[2] M. O. Albertson, G. G. Chappell, H. A. Kierstead, A. Kündgen, and R. Ramamurthi. Coloring with no 2-colored P_4s. *Electronic Journal of Combinatorics*, 11, 2004.

[3] V. E. Alekseev and V. V. Lozin. Augmenting graphs for independent sets. *Discrete Applied Mathematics*, 145(1):3–10, 2004.

[4] N. Alon, J. D. Cohen, T. L. Griffiths, P. Manurangsi, D. Reichman, I. Shinkar, T. Wagner, and A. Yu. Multitasking capacity: Hardness results and improved constructions. *SIAM Journal on Discrete Mathematics*, 34:885–903, 2020.

[5] N. Alon and A. Zaks. Algorithmic aspects of acyclic edge colorings. *Algorithmica*, 32:611–614, 2002.

[6] P. Angelini and F. Frati. Acyclically 3-colorable planar graphs. *Journal of Combinatorial Optimization*, 24:116–130, 2012.

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

[8] A. Atminas, V. V. Lozin, and I. Razgon. Linear time algorithm for computing a small biclique in graphs without long induced paths. *Proceedings of SWAT 2012, LNCS*, 7357:142–152, 2012.

[9] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25:1305–1317, 1996.

[10] H. L. Bodlaender, T. Kloks, R. B. Tan, and J. van Leeuwen. Approximations for lambda-colorings of graphs. *Computer Journal*, 47:193–204, 2004.

[11] J. Bok, N. Jedlicková, B. Martin, P. Ochem, D. Paulusma, and S. Smith. Acyclic, star and injective colouring: A complexity picture for H-free graphs. *CoRR*, abs/2008.09415, 2020.

[12] J. Bok, N. Jedlicková, B. Martin, D. Paulusma, and S. Smith. Acyclic, star and injective colouring: a complexity picture for H-free graphs. In *ESA 2020*, Leibniz International Proceedings in Informatics. Schloss Dagstuhl, 2020.

[13] J. Bok, N. Jedlicková, B. Martin, D. Paulusma, and S. Smith. Injective colouring for H-free graphs. *CSR 2021*, June 2021.

[14] M. Bonamy, K. K. Dabrowski, C. Feghali, M. Johnson, and D. Paulusma. Independent feedback vertex sets for graphs of bounded diameter. *Information Processing Letters*, 131:26–32, 2018.

[15] M. Bonamy, K. K. Dabrowski, C. Feghali, M. Johnson, and D. Paulusma. Independent feedback vertex set for P_5-free graphs. *Algorithmica*, 81:1342–1369, 2019.

[16] F. Bonomo, M. Chudnovsky, P. Maceli, O. Schaudt, M. Stein, and M. Zhong. Three-coloring and list three-coloring of graphs without induced paths on seven vertices. *Combinatorica*, 38:779–801, 2018.

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

[18] A. Brandstädt, V. B. Le, and T. Szymczak. The complexity of some problems related to graph 3-colorability. *Discrete Applied Math.*, 89:59–73, dec 1998.

[19] C. Brause, P. Golovach, B. Martin, D. Paulusma, and S. Smith. Partitioning H-free graphs of bounded diameter. In *ISAAC 2021*, Leibniz International Proceedings in Informatics. Schloss Dagstuhl, 2021.

[20] C. Brause, P. A. Golovach, B. Martin, D. Paulusma, and S. Smith. Acyclic, star and injective colouring: bounding the diameter. In *WG 2021*, Lecture Notes in Computer Science. Springer, 2021.

[21] H. Broersma, P. A. Golovach, D. Paulusma, and J. Song. Updating the complexity status of coloring graphs without a fixed induced linear forest. *Theoretical Computer Science*, 414:9–19, 2012.

[22] T. Calamoneri. The $L(h, k)$-labelling problem: An updated survey and annotated bibliography. *Computer Journal*, 54:1344–1371, 2011.

[23] C. T. Cheng, E. McDermid, and I. Suzuki. Planarization and acyclic colorings of subcubic claw-free graphs. *Proceedings of WG 2011, LNCS*, 6986:107–118, 2011.

[24] M. Chudnovsky, S. Huang, S. Spirkl, and M. Zhong. List 3-coloring graphs with no induced $P_6 + rP_3$. *Algorithmica*, 83, 2021.

[25] M. Chudnovsky, S. Spirkl, and M. Zhong. Four-coloring P_6-free graphs. *Proc. SODA 2019*, 1:1239–1256, 2019.

[26] T. F. Coleman and J.-Y. Cai. The cyclic coloring problem and estimation of sparse Hessian matrices. *SIAM Journal on Algebraic Discrete Methods*, 7:221–235, 1986.

[27] T. F. Coleman and J. J. Moré. Estimation of sparse hessian matrices and graph coloring problems. *Mathematical Programming*, 28:243–270, 1984.

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

[29] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.

[30] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. Solving the 2-disjoint connected subgraphs problem faster than 2^n. *Algorithmica*, 70:195–207, 2014.

[31] R. M. Damerell. On Moore graphs. *Proceedings of the Cambridge Philosophical Society*, 74:227–236, 1973.

[32] Z. Dvořák, B. Mohar, and R. Šámal. Star chromatic index. *Journal of Graph Theory*, 72:313–326, 2013.

[33] K. Edwards. The complexity of colouring problems on dense graphs. *Theoretical Computer Science*, 43:337–343, 1986.

[34] T. Emden-Weinert, S. Hougardy, and B. Kreuter. Uniquely colourable graphs and the hardness of colouring graphs of large girth. *Combinatorics, Probability and Computing*, 7:375–386, 1998.

[35] P. Erdős. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

[36] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5:691–703, 1976.

[37] S. Foldes and P. L. Hammer. Split graphs having dilworth number two. *Canadian Journal of Mathematics*, 29:666–672, 1977.

[38] E. Galby, P. T. Lima, D. Paulusma, and B. Ries. Classifying k-edge colouring for H-free graphs. *Information Processing Letters*, 146:39–43, 2019.

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

[40] M. R. Garey, D. S. Johnson, and R. E. Tarjan. The planar hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5:704–714, 1976.

[41] P. A. Golovach, M. Johnson, D. Paulusma, and J. Song. A survey on the computational complexity of colouring graphs with forbidden subgraphs. *Journal of Graph Theory*, 84:331–363, 2017.

[42] P. A. Golovach, D. Paulusma, and J. Song. Coloring graphs without short cycles and long induced paths. *Discrete Applied Mathematics*, 167:107–120, 2014.

[43] C. Gray, F. Kammer, M. Löffler, and R. I. Silveira. Removing local extrema from imprecise terrains. *Computational Geometry*, 45:334–349, 2012.

[44] C. Groenland, K. Okrasa, P. Rzążewski, A. Scott, P. Seymour, and S. Spirkl. H-colouring P_t-free graphs in subexponential time. *Discrete Applied Mathematics*, 267:184–189, 2019.

[45] B. Grünbaum. Acyclic colorings of planar graphs. *Israel journal of mathematics*, 14:390–408, 1973.

[46] F. Gurski and E. Wanke. Vertex disjoint paths on clique-width bounded graphs. *Theoretical Computer Science*, 359:188–199, 08 2006.

[47] P. Heggernes, P. van 't Hof, E. J. van Leeuwen, and R. Saei. Finding disjoint paths in split graphs. *SOFSEM 2014: Theory and Practice of Computer Science*, 1:315–326, 2014.

[48] C. T. Hoàng, M. Kamiński, V. V. Lozin, J. Sawada, and X. Shu. Deciding k-colorability of P_5-free graphs in polynomial time. *Algorithmica*, 57:74–81, 2010.

[49] A. J. Hoffman and R. R. Singleton. On Moore graphs with diameter 2 and 3. *IBM Journal of Research and Development*, 5:497–504, 1960.

[50] I. Holyer. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10:718–720, 1981.

[51] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:225–231, 1973.

[52] S. Huang. Improved complexity results on k-coloring P_t-free graphs. *European Journal of Combinatorics*, 51:336–346, 2016.

[53] S. Huang, M. Johnson, and D. Paulusma. Narrowing the complexity gap for colouring (C_s, P_t)-free graphs. *Computer Journal*, 58:3074–3088, 2015.

[54] K. ichi Kawarabayashi, Y. Kobayashi, and B. Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 102:424–435, 2012.

[55] R. Janczewski, A. Kosowski, and M. Małafiejski. The complexity of the $L(p, q)$-labeling problem for bipartite planar graphs of small degree. *Discrete Mathematics*, 309:3270–3279, 05 2009.

[56] K. Jansen. Complexity results for the optimum cost chromatic partition problem. *PhD Thesis, Universität Trier, Mathematik/Infomatik, Forschungsbericht*, 1996.

[57] R. M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

[58] W. Kern, B. Martin, D. Paulusma, S. Smith, and E. J. van Leeuwen. Disjoint paths and connected subgraphs for H-free graphs. *Theoretical Computer Science*, 898:59–68, 2022.

[59] W. Kern and D. Paulusma. Contracting to a longest path in H-free graphs. In *31st International Symposium on Algorithms and Computation (ISAAC 2020)*, volume 181 of *Leibniz International Proceedings in Informatics*, pages 22:1–22:18. Schloss Dagstuhl, 2020.

[60] T. Klimošová, J. Malík, T. Masařík, J. Novotná, D. Paulusma, and V. Slívová. Colouring $(P_r + P_s)$-free graphs. *Proceedings of ISAAC 2018, LIPIcs*, 123:5:1–5:13, 2018.

[61] A. V. Kostochka. Upper bounds of chromatic functions of graphs. *PhD Thesis, University of Novosibirsk*, 1978.

[62] D. Král', J. Kratochvíl, Zs. Tuza, and G. J. Woeginger. Complexity of coloring graphs without forbidden induced subgraphs. *Proceedings of WG 2001, LNCS*, 2204:254–262, 2001.

[63] M. S. Krishnamoorthy. An NP-hard problem in bipartite graphs. *SIGACT News*, 7:26, Jan. 1975.

[64] H. Lei, Y. Shi, and Z.-X. Song. Star chromatic index of subcubic multigraphs. *Journal of Graph Theory*, 88:566–576, 2018.

[65] D. Leven and Z. Galil. NP completeness of finding the chromatic index of regular graphs. *Journal of Algorithms*, 4:35–44, 1983.

[66] C. Linhares-Sales, A. K. Maia, N. A. Martins, and R. M. Sampaio. Restricted coloring problems on graphs with few P_4's. *Annals of Operations Research*, 217:385–397, 2014.

[67] E. L. Lloyd and S. Ramanathan. On the complexity of distance-2 coloring. *Proceedings of ICCI 1992*, 1:71–74, 1992.

[68] L. Lovász. Coverings and coloring of hypergraphs. *Congress Numerantium*, VIII:3–12, 1973.

[69] V. V. Lozin and M. Kaminski. Coloring edges and vertices of graphs without short or long cycles. *Contributions to Discrete Mathematics*, 2, 2007.

[70] A. Lyons. Restricted coloring problems and forbidden induced subgraphs. *Preprint ANL/MCS-P1611-0409, Mathematics and Computer Science Division, Argonne National Laboratory*, 2009.

[71] A. Lyons. Acyclic and star colorings of cographs. *Discrete Applied Mathematics*, 159:1842–1850, 2011.

[72] M. Mahdian. On the computational complexity of strong edge coloring. *Discrete Applied Mathematics*, 118:239–248, 2002.

[73] B. Martin, D. Paulusma, and S. Smith. Colouring H-free graphs of bounded diameter. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).*, volume 138 of *Leibniz international proceedings in informatics*, pages 14:1–14:14. Schloss Dagstuhl Leibniz-Zentrum für Informatik, Dagstuhl, Germany, August 2019.

[74] B. Martin, D. Paulusma, and S. Smith. Colouring graphs of bounded diameter in the absence of small cycles. In *CIAC 2021*, Lecture Notes in Computer Science. Springer, 2021.

[75] B. Martin, D. Paulusma, and S. Smith. Hard problems that quickly become very easy. *Information Processing Letters*, 174:106213, 2022.

[76] D. Marx, B. O'Sullivan, and I. Razgon. Treewidth reduction for constrained separation and bipartization problems. In *STACS*, 2010.

[77] S. Mccormick. Optimal approximation of sparse hessians and its equivalence to a graph coloring problem. *Mathematical Programming*, 26:153–171, 1983.

[78] G. B. Mertzios and P. G. Spirakis. Algorithms and almost tight results for 3-colorability of small diameter graphs. *Algorithmica*, 74:385–414, 2016.

[79] D. Mondal, R. I. Nishat, M. S. Rahman, and S. Whitesides. Acyclic coloring with few division vertices. *Journal of Discrete Algorithms*, 23:42–53, 2013.

[80] S. Natarajan and A. P. Sprague. Disjoint paths in circular arc graphs. *Nordic Journal of Computing*, 3:256–270, sep 1996.

[81] P. Ochem. Graph colouring and combinatorics on words. *PhD Thesis, University of Bordeaux*, 2005.

[82] S. Olariu. Paw-free graphs. *Information Processing Letters*, 28:53–54, 1988.

[83] C. M. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[84] D. Paulusma and J. M. van Rooij. On partitioning a graph into two connected subgraphs. *Theoretical Computer Science*, 412:6761–6769, 2011.

[85] N. Robertson and P. Seymour. Graph minors .xiii. the disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63:65–110, 1995.

[86] W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences*, 4:177–192, 1970.

[87] T. J. Schaefer. The complexity of satisfiability problems. *STOC*, 1:216–226, 1978.

[88] A. Sen and M. L. Huson. A new model for scheduling packet radio networks. *Wireless Networks*, 3:71–82, 1997.

[89] M. Shalu and C. Antony. Complexity of restricted variant of star colouring. In *CALDAM 2020*, pages 3–14, 01 2020.

[90] Y. Shiloach. A polynomial solution to the undirected two paths problem. *J. ACM*, 27:445–456, jul 1980.

[91] R. R. Singleton. There is no irregular Moore graph. *Amererican Mathematical Monthly*, 75:42–43, 1968.

[92] J. A. Telle and Y. Villanger. Connecting terminals and 2-disjoint connected subgraphs. In A. Brandstädt, K. Jansen, and R. Reischuk, editors, *Graph-Theoretic Concepts in Computer Science*, pages 418–428, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[93] E. J. Van Leeuwen. Optimizarion and approximation on systems of geometric objects. *PhD Thesis, Universiteit van Amsterdam*, 2009.

[94] P. van 't Hof, D. Paulusma, and G. J. Woeginger. Partitioning graphs into connected parts. *Theoretical Computer Science*, 410:4834–4843, 2009.

[95] R. J. Wilson. *Four Colors Suffice: How the Map Problem Was Solved - Revised Color Edition.* Princeton University Press, 2021.

[96] X. Zhou, Y. Kanari, and T. Nishizeki. Generalized vertex-coloring of partial k-trees. *IEICE Transactions on Fundamentals of Electronics, Communication and Computer Sciences*, E83-A:671–678, 2000.