Structural Solutions to Maximum Independent Set and Related Problems

by

Konrad Kazimierz Dabrowski

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

DIMAP & Mathematics Institute

October 2012

THE UNIVERSITY OF
WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

First and foremost, I would like to thank my supervisor Vadim V. Lozin for his extraordinary patience and for all the help and advice he has given me over the past three years. I would also like to thank him for his guidance and expertise in the research on which we collaborated. I would also like to extend these thanks to my coauthors.

I would like to thank the staff and students at The Centre for Discrete Mathematics and its Applications (DIMAP) and the Mathematics Institute at the University of Warwick. I would particularly like to thank the members of Warwick Folk Society for their humour (see also [McNoleg, 1996; Upper, 1974]) and the many happy hours I spent with them in-between doing research.

I would like to thank the members of the UK Mathematics Trust, who helped inspire me to study mathematics and the staff and students at the University of Cambridge for sustaining my further interest in the subject. Among these people, I would particularly like to thank Professor Imre Leader.

I would like to thank my parents for their support and love. My thanks also go to Claire for her endless patience, love and support.

This research was supported by the Mathematics Institute and DIMAP at the University of Warwick. DIMAP is partially funded by EPSRC Science and Innovation Award EP/D063191/1 and my research was also funded as part of this grant.

Declarations

All the work in this thesis is joint work with my thesis supervisor Vadim Lozin. In addition to this

Chapter 3 is based on [Dabrowski et al., 2011, 2012a], which was joint work with Vadim Lozin, Haiko Müller and Dieter Rautenbach.

Chapter 4 is based on [Dabrowski et al., 2013], which was joint work with Vadim Lozin, Marc Demange.

Chapter 5 is based on [Dabrowski et al., 2012b], which was joint work with Vadim Lozin and Juraj Stacho.

Section 5.6 is joint work with Vadim Lozin. The result in this section appeared in [Dabrowski et al., 2012c], which was joint work with Vadim Lozin and Victor Zamaraev.

Chapter 7 is based on [Dabrowski et al., 2010, 2012d], which was joint work with Vadim Lozin, Rajiv Raman and Bernard Ries.

The work in this thesis has not been submitted for a degree at another university.

Abstract

In this thesis, we study some fundamental problems in algorithmic graph theory. Most natural problems in this area are hard from a computational point of view. However, many applications demand that we do solve such problems, even if they are intractable. There are a number of methods in which we can try to do this:

- We may use an approximation algorithm if we do not necessarily require the best possible solution to a problem
- Heuristics can be applied and work well enough to be useful for many applications
- We can construct randomised algorithms for which the probability of failure is very small
- We may parameterize the problem in some way which limits its complexity

In other cases, we may also have some information about the structure of the instances of the problem we are trying to solve. If we are lucky, we may find that we can exploit this extra structure to find efficient ways to solve our problem. The question which arises is "How far must we restrict the structure of our graph to be able to solve our problem efficiently?"

In this thesis we study a number of problems, such as Maximum Independent Set, Maximum Induced Matching, Stable-Π, Efficient Edge Domination, Vertex Colouring and Dynamic Edge-Choosability. We try to solve problems on various hereditary classes of graphs and analyse the complexity of the resulting problem, both from a classical and parameterized point of view.

Chapter 1

Introduction

1.1 Introduction

Graphs are a very useful model for many real-world structures. Graph Theory has applications to all sorts of scientific disciplines, including everything from the structure of molecules in chemistry and physics, to analysing social networks in sociology.

With the development of the digital computer over the past 70 years, the field of computer science has blossomed and grown exponentially. Graphs are ubiquitous in the field of computer science and provide a natural framework with which to represent various concepts such as the organisation of data or communication over networks.

Hand in hand with the rise of computer science has been the proliferation of algorithmic graph theory, i.e. the study of how to solve various problems on graphs. Unfortunately, most natural problems in this field cannot be solved in an efficient manner. However, we still need to be able to solve these problems. Many approaches have been developed over the years to help us do this. For example:

- We may use an approximation algorithm if we do not necessarily require the best possible solution to a problem
- Heuristics can be applied and work well enough to be useful for many applications
- We can construct randomised algorithms for which the probability of failure is very small
- We may parameterize the problem in some way which limits its complexity

In other cases, we may have some extra information about the structure of the instances of the problem which we will have to solve. If we are lucky, we may find that we can exploit this extra structure to find efficient ways to solve our problem. The question which arises is "How far must we restrict the structure of our graph to be able to solve our problem efficiently?"

In this thesis we explore the above question and try to give partial answers to it for various problems and various definitions of what exactly it means to *efficiently* solve a problem. There are many problems in algorithmic graph theory that are very complicated and very specific to their applications. In this thesis, we mostly focus on problems that are in some sense "basic" in that they occur naturally in many different applications and as a result have been widely studied. Examples of such problems include MAXIMUM INDEPENDENT SET and VERTEX COLOURING.

1.2 Basic Definitions and Notation

A graph G consists of a *vertex set* $V(G)$ and an *edge set* $E(G)$. An edge in a graph consists of an unordered pair $\{x, y\}$ of distinct vertices of the graph and we will usually denote this as xy. Unless stated otherwise, n denotes the number of vertices in G and m denotes the number of edges. We say that two vertices x, y are adjacent if $xy \in E(G)$ and nonadjacent otherwise. All graphs in this thesis are finite, undirected, without loops or multiple edges. The *neighbourhood* $N(v)$ of a vertex v is the set of all vertices adjacent to v. The *degree* $d(v)$ of a vertex v is the size of its neighbourhood. If X is a set of vertices, we define $N_X(v) = N(v) \cap X$ to be the *neighbourhood of* v *in* X i.e. the set of vertices in X which are adjacent to v. Similarly, if U, X is a set of vertices, $N(U) = \cup_{v \in U} N(v)$ denotes the *neighbourhood of* U and $N_X(U) = \cup_{v \in U} N_X(v)$ denotes the *neighbourhood of* U *in* X. Note that if two distinct vertices have the same neighbourhood, they must be nonadjacent. The *closed neighbourhood* of a vertex x in a graph G is $N_G[x] = N_{V(G)}(x) \cup \{x\}$.

Two edges are *incident* if they share an end-vertex. They are *linked* if they are either incident or are both incident to a common third edge (see Fig. 1.1). A graph is *d-regular* if every vertex in the graph is of degree d. It is *regular* if it is d-regular for some d.

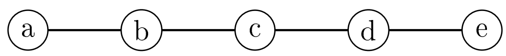


Figure 1.1: The edge ab is incident with bc, but not with cd or de. It is linked with bc and cd, but not with de.

Given two graphs G and H, it is customary to say that G is a *subgraph* of H if

$V(G) \subseteq V(H)$ and $E(G) \subseteq E(H)$. We say that G is an *induced subgraph* of H if G is a subgraph of H and $E(G) = E(H) \cap (V(G) \times V(G))$. If U is a set of vertices, $G[U]$ denotes the *subgraph of G induced by U*, i.e. the graph with vertex set U and edge set $E(G) \cap (U \times U)$.

We say that a graph is *empty* or *edgeless* if it contains no edges. A set X of vertices in a graph G is *independent* if $G[X]$ is edgeless. Such a set is sometimes referred to as a *stable* set. A set X is a *clique* if every vertex is X is adjacent to every other vertex in X. A graph is *bipartite* if its vertex set can be partitioned into two sets, both of which are independent. A graph is a *split graph* if its vertex set can be partitioned into an independent set and a clique. The *complement* $\overline{G}$ of a graph G is the graph with the same vertex set as G, but where an edge is present in $\overline{G}$ if and only if it is not present in G. If G is a bipartite graph with vertex partition $X \cup Y$, the *bipartite complement* of G is the graph with the same vertex partition and with edge set $(X \times Y) \setminus E(G)$.

For disjoint sets $A, B \subseteq V(G)$, we say that A is *complete* to B if every vertex in A is adjacent to every vertex in B, and that A is *anticomplete* to B if every vertex in A is non-adjacent to every vertex in B.

A *module* in a graph G is a set M of vertices of G, such that every vertex in $V(G) \setminus M$ is either adjacent to all vertices in M or none of them. A module is *trivial* if it contains either all vertices of G, exactly one vertex of G or if it is empty, otherwise it is *non-trivial*. A graph is said to be *prime* if it has no non-trivial modules. A module M is *maximal* if there is no module M' such that $M \subsetneq M' \neq V(G)$. Note that if two modules are disjoint, they must either be complete or anticomplete to each-other.

As usual, K_n, C_n and P_n denote the *complete graph*, the *chordless cycle* and the *chordless path* on n vertices, respectively. $K_{n,m}$ is the *complete bipartite graph*, also known as a bi-clique, with parts of size n and m. $K_n - e$ denotes the graph obtained from the graph K_n by deleting a single edge. If X is a set of vertices in a graph G, $G - X$ denotes the graph $G[V(G) \setminus X]$.

For non-negative integers i, j, k, the graph $S_{i,j,k}$ denotes the tree formed by taking 3 paths of length i, j, k respectively and identifying the vertices at one end of each of the paths. In other words, if i, j, k are positive integers, $S_{i,j,k}$ denotes a tree with exactly three leaves, which are at distance i, j and k from the only vertex of degree 3 (see also Figure 6.1). In particular, $S_{1,1,1} = K_{1,3}$ is known as the *claw*, and $S_{1,2,2}$ is sometimes denoted by E, since this graph can be drawn as the capital letter E (see Figure 1.2). H denotes the graph that can be drawn as the capital letter H (see Figure 1.2). The graph obtained from a $K_{1,4}$ by subdividing exactly one edge exactly once is called a *cross*. Given two graphs G and G', we write $G + G'$ to denote the disjoint union of G and G'.

In particular, mG is the disjoint union of m copies of G.

(a) H (b) E

Figure 1.2: The graphs H and E

The *distance* $d(x, y)$ between two vertices x, y is the minimum length of a path between them (or infinity if no such path exists). A graph is *connected* if every pair of vertices in the graph is connected by a path. For a graph G, the *line graph* of G, denoted $L(G)$, is the graph with vertex set $E(G)$, where two vertices are adjacent in $L(G)$ if and only if the respective edges of G are incident (i.e. share an end-vertex). The *square* G^2 of a graph G is the graph formed by connecting (with an edge) all pairs of vertices at distance at most 2 in the original graph.

The *independence number* $\alpha(G)$ of a graph G is the size of a largest independent set in G, while the *clique number* $\omega(G)$ is the size of a largest clique. A *vertex cover* is a set of vertices containing at least one end-vertex of every edge in the graph. A *matching* in a graph is a set of edges, no two of which are incident. An *induced matching* is a matching such that no two vertices belonging to a different edge of the matching are adjacent. Equivalently, an induced matching is a 1-regular induced subgraph of a graph. The minimum size of a vertex cover in a graph G is denoted $\nu(G)$. The maximum size of a matching or induced matching in a graph G are denoted $\mu(G)$ and $i\mu(G)$, respectively. A set of vertices D *dominates* a graph if every vertex not in D is adjacent to at least one vertex of D.

We use $R(r, s)$ to denote the Ramsey number, i.e. the minimum number n such that every graph with at least n vertices contains either an independent set of size r or a clique of size s. For a real number x, $\lceil x \rceil$ denotes the smallest integer $\geq x$.

The *clique-width* of a graph G is the minimum number of labels needed to construct G using the following four operations:

(i) Creating a new vertex v with label i (denoted by $i(v)$).

(ii) Taking the disjoint union of two labelled graphs G and H (denoted by $G \oplus H$).

(iii) Joining each vertex with label i to each vertex with label j ($i \neq j$, denoted by $\eta_{i,j}$).

(iv) Renaming label i to j (denoted by $\rho_{i\to j}$).

Every graph can be defined by an algebraic expression using these four operations. For instance, an induced path on five consecutive vertices a, b, c, d, e has clique-width equal to 3 and it can be defined as follows:

$$\eta_{3,2}(3(e) \oplus \rho_{3\to 2}(\rho_{2\to 1}(\eta_{3,2}(3(d) \oplus \rho_{3\to 2}(\rho_{2\to 1}(\eta_{3,2}(3(c) \oplus \eta_{2,1}(2(b) \oplus 1(a)))))))))$$

1.3 Hereditary Classes of Graphs

A *graph property* or *class of graphs* is a set of graphs which is closed under isomorphism. A class of graphs is *hereditary* if if is closed under taking induced subgraphs.

If $M = \{G_1, G_2, \ldots\}$ is a (not necessarily finite) set of graphs, we say that a graph G is *M-free* or *$(G_1, G_2, \ldots)$-free* if no graph in M is an induced subgraph of G. If M contains only a single graph H, for the sake of clarity, we sometimes omit the brackets and say that G is *H-free*. The set of all M-free graphs is denoted $Free(M)$. The set M is called the set of *forbidden induced subgraphs* for the class of graphs $Free(M)$. Clearly, any class of the form $Free(M)$ is hereditary. Conversely, if we have a hereditary class $\mathcal{C}$ of graphs and let M be the set of graphs not in C, it is easy to see that $\mathcal{C} = Free(M)$.

A graph G is a *minimal* forbidden induced subgraph for a hereditary class X if and only if $G \notin X$ but every proper induced subgraph of G belongs to X (or alternatively, the deletion of any vertex from G results in a graph that belongs to X). Let $MFIS(X)$ denote the set of all minimal forbidden induced subgraphs for a hereditary class X.

Theorem 1. *For any hereditary class X, we have $X = Free(MFIS(X))$. Moreover, $MFIS(X)$ is the unique minimal set with this property.*

Proof. First, suppose that $G \in X$. Then by definition all induced subgraphs of G belong to X and hence no graph from $MFIS(X)$ is an induced subgraph of G, since none of them belongs to X. As a result, $G \in Free(MFIS(X))$, so $X \subseteq Free(MFIS(X))$.

Suppose now that $G \in Free(MFIS(X))$, and suppose, for contradiction, that $G \notin X$. Let H be a minimal induced subgraph of G which is not in X, (it may happen that $H = G$). But then $H \in MFIS(X)$ contradicting the fact that $G \in Free(MFIS(X))$. This contradiction shows that $G \in X$ and hence proves that $Free(MFIS(X)) \subseteq X$.

To prove the uniqueness of the set $MFIS(X)$, we will show that for any set N such that $X = Free(N)$ we have $MFIS(X) \subseteq N$. Assume this is not true and let H be a graph in $MFIS(X) \setminus N$. By the minimality of the graph H, any proper induced subgraph of H is in X, and hence is in $Free(N)$. Together with the fact that H does not belong to N, we conclude that $H \in Free(N)$. Therefore $H \in Free(MFIS(X))$. However, this contradicts the fact that $H \in MFIS(X)$, completing the proof. $\square$

When specifying the forbidden induced subgraph characterisation of a hereditary class of graphs, we therefore normally only list the minimal ones. By similar arguments, we can also use the forbidden induced subgraph characterisation to test if one hereditary class contains another:

Theorem 2. *Let $X = Free(M)$ and $Y = Free(N)$ be two hereditary classes. Then X is a subclass of Y if and only if for every $H \in N$, there exists some $G \in M$ such that G is an induced subgraph of H.*

A vast multitude of hereditary graph classes has been studied in the literature (see e.g. [Brandstädt et al., 1999]), both with finite and infinite minimal forbidden induced subgraph characterisations. For example:

- $Free(C_3, C_4, C_5, \ldots)$ is the class of forests
- $Free(C_3, C_5, C_7, \ldots)$ is the class of bipartite graphs
- $Free(C_4, C_5, 2K_2)$ is the class of split graphs
- $Free(P_4)$ is the class of co-graphs

1.4 Introduction to Computational complexity

An algorithm runs in *polynomial time* if the number of elementary operations that the algorithm carries out is bounded by a polynomial in the size of the input instance. The class of problems which can be solved in polynomial time is usually denoted P.

Intuitively, we can solve a problem quickly if it can be solved by a polynomial-time algorithm and we cannot solve the problem efficiently if there is no polynomial-time algorithm. This does not always transfer over to real life applications. Indeed there are polynomial-time algorithms that would take hundreds of years to run even on small problem instances and, conversely, there are problems solved every day for which it is believed that no polynomial-time algorithm exists. However, saying that polynomial-time algorithms are efficient and other algorithms are not is a useful rule of thumb.

A *decision problem* is a yes-or-no question in some formal system. A decision problem is said to be in the class NP if it has the property that when the answer is "yes", a *proof* of this can be given and this proof can be verified in polynomial time. An example of such a problem is "Given an input graph G and an input integer k, does the graph G contain an independent set of size k?" Indeed, this problem lies in NP since an independent set of size k in G would constitute a proof.

To compare the hardness of solving problems, we introduce the idea of a *reduction* from one problem to another. If P and Q are decision problems, a polynomial-time algorithm A is a *polynomial-time reduction from P to Q* if it takes an instance x of problem P as input and outputs an instance y of problem Q with the property that $P(x) =$ "yes" if and only if $Q(y) =$ "yes." A problem is said to be *NP-complete* if every problem in NP has a polynomial-time reduction to this problem.

There is a large class of problems which have been shown to be NP-complete. Indeed, many standard problems in algorithmic graph theory fall into this category. It is widely assumed that $P \neq NP$, i.e. that NP-complete problems cannot be solved in polynomial time. (The NP-complete problems are "polynomially equivalent" in the sense that if any of them can be solved in polynomial time then they all can.) It should be noted that if $P \neq NP$, then there are also infinitely many intermediate levels of computational complexity in between them.

One way of dealing with NP-complete problems comes from the notion of parameterized complexity. We introduce a parameter k and hope that this parameter will somehow absorb all the "non-polynomial" behaviour in the problem. More formally, we say that an instance of a parameterized problem is a pair (G, k), where G is an input for the problem and k is a parameter assigning a natural number to each input. A parameterized problem is *fixed-parameter tractable (fpt)* if it can be solved in $f(k)n^{O(1)}$ time, where n is the size of the input G and $f(k)$ is a computable function depending only on the value of the parameter k. We say that such an algorithm runs in *fpt-time*. We usually think of the parameter as being "small" and fixed, while n tends to infinity.

As for classical complexity, we have a parameterized notion of a reduction. If P and Q are parameterized decision problems, an algorithm A is a *fixed-parameter reduction* from P to Q if it takes an instance (x, k) of problem P as input and outputs an instance (y, k') of problem Q with the properties that $P(x, k) =$ "yes" if and only if $P(y, k') =$ "yes", where $k' \leq g(k)$ for some function g, whose value depends only on the parameter k.

If an NP-complete problem is fixed-parameter tractable for the parameter k, intuitively, we see that a *reason* the problem is NP-complete is because this parameter can be large. Again, among parameterized problems, there are classes which are conjectured not to be fixed-parameter tractable. In fact, there is a whole hierarchy of such classes, known as the W-hierarchy.

To help define these classes, we introduce the WEIGHTED WEFT-t DEPTH-h CIRCUIT-SAT problem. This takes as input a boolean circuit C with a mixture of fan-in at most 2 and unbounded fan-in gates. The number of unbounded fan-in gates along any path from an input to the output is at most t and the total depth (both fan-in at most 2 and unbounded fan-in) is at most h. The problem asks whether C has a satisfying assignment (one where the output is "True") in which exactly k of the inputs are set to "True." For $t \geq 1$, we define $W[t]$ to be the class of parameterized problems that are fixed-parameter reducible to WEIGHTED WEFT-t DEPTH-h CIRCUIT-SAT for some fixed h (depending only on the problem). $W[0]$ is defined to be the class of problems solvable in fpt-time. Again, a problem is $W[t]$-complete if every problem in $W[t]$ is fixed-parameter reducible to this problem.

An example of a $W[1]$-complete problem is "does the graph G contain an independent set of size k?" An example of a $W[2]$-complete problem is "does the graph G contain an dominating set of size k?" Most natural parameterized problems seem to belong to either $W[0], W[1]$ or $W[2]$. For more information on the W-hierarchy and parameterized complexity in general, we refer the reader to [Downey and Fellows, 1999; Flum and Grohe, 2006].

One technique for producing fpt algorithms is the use of *kernelization.* A kernelization is an algorithm that takes an instance (G, k) of a problem and transforms it, in polynomial time, to an instance (G', k') such that both k' and the size of G' are bounded by a function of k. The output instance (G', k') is known as the *kernel.* In fact, a problem is fixed-parameter tractable if and only if it has a kernelization.

A maximum matching in a graph G is equivalent to a maximum independent set in $L(G)$. However, while a maximum matching can be found in polynomial time [Edmonds, 1965], finding a maximum independent set of maximum size is NP-hard.

Typically, in the parameterized complexity setting, we parameterize the MAXIMUM INDEPENDENT SET problem by the solution size, and the problem is known to be W[1]-hard under this parameterization (see e.g. [Downey and Fellows, 1999; Flum and Grohe, 2006]). However, we show that if the problem is parameterized by the size of a maximum matching in the input graph, the problem becomes fixed-parameter tractable.

Theorem 3. *The* MAXIMUM INDEPENDENT SET *and* MINIMUM VERTEX COVER *problem can be solved for graphs with n vertices and a maximum matching of size μ in time $f(\mu)p(n)$, where $f(\mu)$ is a function of μ independent of n and $p(n)$ is a polynomial in n*

independent of μ.

Proof. Finding a maximum independent set in a graph is equivalent to finding a minimum vertex cover, since X is a maximum independent set if and only if $V(G) \setminus X$ is a minimum vertex cover. Since the MINIMUM VERTEX COVER problem is fixed-parameter tractable, it can be solved for a graph G with n vertices and a minimum vertex cover of size ν in time $f(\nu)p(n)$, where $f(\nu)$ is a function independent of n and $p(n)$ is a polynomial independent of ν. Since $\nu \leq 2\mu$ [Lovász and Plummer, 1986], we conclude that one can solve both the MINIMUM VERTEX COVER and the MAXIMUM INDEPENDENT SET problems in time bounded by $f(2\mu)p(n)$. □

This result demonstrates that the choice of parameter is very important. If we change the parameter, we can get a problem with completely different complexity characteristics.

1.5 Outline of Thesis

Part I

In Part I, we study the MAXIMUM INDEPENDENT SET Problem. This is the problem of trying to find an independent set in a graph of maximum size.

In Chapter 2 we study augmenting graphs. We prove a Ramsey-type result on classes of augmenting graphs. We then study a set of subclasses of P_5-free graphs and show that augmenting graphs can be used to solve MAXIMUM INDEPENDENT SET in polynomial time in these classes.

In Chapter 3 we study the weighted version of the problem from the point of view of parameterized complexity. We exhibit a number of classes in which the problem is fixed-parameter tractable.

Chapter 4 deals with the MAXIMUM INDUCED MATCHING problem. This is equivalent to the MAXIMUM INDEPENDENT SET problem in $L(G)^2$, the square of the line graph of our input graph. We exhibit a number of classes where the problem is fixed-parameter tractable and some where the problem is hard from the point of view of approximation algorithms. We also exhibit a simple solution in the class of hypercubes.

Part II

In Part II, we deal with graph partition problems.

Chapter 5 considers the Stable-Π problem. In this problem, rather than finding an independent set of maximum size, we try to find an independent set such that the remainder of the graph (not in the independent set) obeys certain properties. More precisely, for a class of graphs Π, the Stable-Π problem asks whether we can partition the vertices of a graph into an independent set and a set which induces a graph in the class Π. We show that for many hereditary classes, the problem can be solved in polynomial time, as long as the class is small enough. We also demonstrate some other classes where the problem is hard. Finally, we exhibit a new class of graphs, which is large in a certain technical sense.

Chapter 6 deals with the Efficient Edge Domination problem. This is the particular case of the Stable-Π problem where Π is the class of 1-regular graphs. This problem is known to be NP-complete. We show that the problem is fixed-parameter tractable with respect to two natural parameterizations. We then classify the (classical) complexity of the problem in the class of F-free graphs for every graph F on at most 6 vertices.

Part III

In Part III, we consider colouring problems.

Chapter 7 considers the Vertex Colouring problem in various subclasses of triangle-free graphs. Vertex Colouring is the problem of partitioning the vertices of a graph into the minimum possible number of independent sets. While the decision version of the problem is NP-complete on K_3-free graphs, we find a number of subclasses where the problem can be solved in polynomial time. In particular, we completely classify the complexity of Vertex Colouring in (K_3, F)-free graphs for any graph F on at most 6 vertices.

Chapter 8 considers a colouring problem which was introduced recently. We show that, unusually for algorithmic graph problems, the Dynamic Edge-Choosability problem can be completely solved in polynomial time for general graphs using a simple algorithm.

Part I

Maximum Independent Sets

An independent set in a graph is a set of vertices, no two of which are adjacent. There are a number of problems associated with this notion. The most important of these is Maximum Independent Set. It finds applications across various fields, such as information theory and computer vision.

In the *decision* version of this problem, we are given a graph G and an integer k and have to determine whether or not the graph G has an independent set of size k. There is also an *optimisation* version, in which we are asked to find an independent set of maximum size. The maximum possible size of an independent set in a graph G is its *independence number* $\alpha(G)$. One more version of the problem asks us to determine the value of $\alpha(G)$. We shall use Maximum Independent Set to refer to the optimisation version of the problem.

From a computational point of view Maximum Independent Set is a hard problem, i.e. it is NP-hard. Moreover, it remains NP-hard under substantial restrictions, for instance, for triangle-free graphs [Murphy, 1992] and for planar cubic graphs [Alimonti and Kann, 1997]. The problem is also hard from a parameterized point of view. More precisely it is $W[1]$-hard when parameterized by the solution size (see e.g. [Downey and Fellows, 1999; Flum and Grohe, 2006]).

There are several main approaches to cope with intractability of computationally hard problems:

1. Polynomial-time algorithms that solve the problem exactly for graphs in special classes
2. Fixed-parameter tractable algorithms that solve the problem exactly for graphs in special classes
3. Polynomial-time algorithms that provide approximate solutions

The third approach to the Maximum Independent Set problem (approximate solutions) is not of much help, because a maximum independent set in a graph is hard to approximate. Indeed, for any $\epsilon > 0$, non-exact polynomial-time algorithms cannot approximate the size of a maximum independent set within a factor of $n^{1-\epsilon}$ [Håstad, 1999]. In this part of the thesis, we focus on the first two approaches, i.e. polynomial-time and fixed-parameter tractable algorithms for graphs in special classes.

The problem has been shown to be solvable in polynomial time in many particular classes of graphs, such as perfect graphs [Grötschel et al., 1993], claw-free graphs [Minty, 1980], and graphs of bounded clique-width [Courcelle et al., 2000].

The solution for claw-free graphs extends the celebrated matching algorithm due to Edmonds [1965] and exploits the idea of augmenting chains due to Berge [1957]. This idea was later developed into a general approach to the MAXIMUM INDEPENDENT SET problem, known as the augmenting graph technique, and was applied to obtain polynomial-time solutions in many restricted graph classes.

In Chapter 2, we first contribute to the theory of augmenting graphs by proving a Ramsey-type result and then apply the technique to solve the MAXIMUM INDEPENDENT SET problem in a particular family of subclasses of P_5-free graphs. Our interest in P_5-free graphs is motivated by the fact that the complexity status of the MAXIMUM INDEPENDENT SET problem in the class of P_5-free graphs is unknown and P_5 is the unique smallest forbidden graph for which this question is open. On the other hand, it is known that the problem can be solved in the class of P_5-free graphs [Randerath and Schiermeyer, 2010] in subexponential time.

In Chapter 3, we study parameterized algorithms for the MAXIMUM INDEPENDENT SET problem in particular graph classes. There is very little existing literature on this topic and we contribute several new results in this direction.

In addition to wide applicability of the MAXIMUM INDEPENDENT SET problem, the importance of this problem is also caused by the fact that it is related to many other problems in algorithmic graph theory. For example, if S is an independent set in G then S is a clique in $\overline{G}$ and $V(G) \setminus S$ is a vertex cover in G. Thus MAXIMUM INDEPENDENT SET in a graph G is equivalent to MAXIMUM CLIQUE in $\overline{G}$ and MINIMUM VERTEX COVER in G.

Two other problems closely related to MAXIMUM INDEPENDENT SET are MAXIMUM MATCHING and MAXIMUM INDUCED MATCHING. Solving these problems for a graph G is equivalent to solving MAXIMUM INDEPENDENT SET in the line graph $L(G)$ and its square $L(G)^2$, respectively. However, while MAXIMUM MATCHING can be solved in linear time [Edmonds, 1965], MAXIMUM INDUCED MATCHING is NP-hard, even for bipartite graphs of maximum degree 3 [Lozin, 2002b]. We discuss the MAXIMUM INDUCED MATCHING problem in more detail in Chapter 4.

Chapter 2

Augmenting Graphs

2.1 Introduction

We say that a bipartite graph G is a triple (B, W, E), where $B \cup W$ is the partition of G into independent sets and $E \subseteq B \times W$ is the set of edges in G.

Let G be a graph containing an independent set S and let $S' = V(G) \setminus S$. We say that the vertices in S are *black* and the vertices in S' are *white*. Suppose $B \subseteq S$ and $W \subseteq S'$. Note that B is an independent set. If W is an independent set, $|W| > |B|$ and $N_S(W) \subseteq B$, we say that the bipartite graph $H = G[W \cup B]$ is *augmenting* (for the set S in the graph G). The *increment* of an augmenting graph H is $\Delta(H) = |W| - |B|$. An augmenting graph is *minimal* if it does not contain a smaller augmenting graph of the same increment. For an independent set S, a *maximum* augmenting graph H is one that maximises $\Delta(H)$.

Note that if T is a larger independent set than S, then setting $W = T \setminus S$ and $B = S \setminus T$ will cause $G[W \cup B]$ to be an augmenting graph for S. And if $H = G[W \cup B]$ is an augmenting graph for an independent set S in G, then $T = (S \cup W) \setminus B$ is a larger independent set than S. In this case we say that T is obtained from S by applying a *H-augmentation*. Thus we have the following theorem:

Theorem 4 (Augmenting Graph Theorem). *An independent set S in a graph G is maximum if and only if there are no augmenting graphs for S.*

This theorem suggests the following general approach to find a maximum independent set in a graph G: begin with any independent set S in G and as long as S admits an augmenting graph H, apply H-augmentation to S. Clearly the problem of finding augmenting graphs is generally NP-hard, as the maximum independent set problem is

NP-hard. However, this approach has proven to be a useful tool to develop approximate solutions to the problem, to compute bounds on the independence number, and to solve the problem in polynomial time for graphs in special classes (see [Hertz and Lozin, 2005] for a survey of such results). For a polynomial-time solution, one has to

(a) find a complete list of augmenting graphs in the class under consideration,

(b) develop polynomial-time algorithms for detecting augmenting graphs in the class, if any are present.

Obviously, if the list of augmenting graphs is finite, then they must be bounded in size. If this bound is known, then all augmenting graphs can be detected in polynomial time. Therefore, only *infinite* families of augmenting graphs are of interest. In Section 2.2, we show that, with the restriction to hereditary classes, there are exactly three minimal infinite families of connected augmenting graphs. If we consider a hereditary class of graphs where the set of possible augmenting graphs contains none of these infinite families, then the list of connected augmenting graphs will be bounded and we will be able to solve the MAXIMUM INDEPENDENT SET problem in polynomial time.

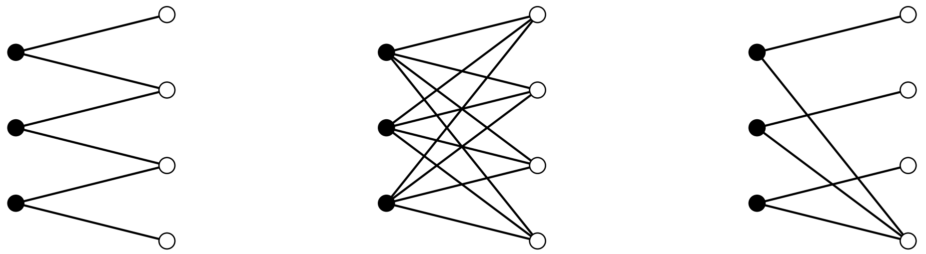
In Section 2.3, we study augmenting graphs in the class of P_5-free graphs. As we mentioned earlier, the complexity status of the MAXIMUM INDEPENDENT SET problem in the class of P_5-free graphs is unknown (although the problem can be solved in this class in subexponential time [Randerath and Schiermeyer, 2010]) and P_5 is the unique smallest forbidden induced subgraph for which this question is open.

Polynomial-time algorithms have been constructed for various subclasses of P_5-free graphs and for many of them, the problem was solved by means of augmenting graphs (see e.g. [Boliac and Lozin, 2003; Gerber et al., 2003; Lozin and Mosca, 2009]). In Section 2.3, we first prove some general results about P_5-free augmenting graphs and then apply the technique to solve the problem in the class of $(P_5, K_{3,z} - e)$ free graphs (Sections 2.3.1 and 2.3.2). Our solution generalises the results for $(P_5, K_{3,3} - e)$-free graphs [Lozin and Mosca, 2009], $(P_5, K_{2,z})$-free graphs [Gerber and Lozin, 2003] and for $(P_5, K_{2,z} - e)$-free graphs [Boliac and Lozin, 2003].

2.2 A Ramsey-type Result for Augmenting Graphs

According to Ramsey's theorem, there are precisely two minimal infinite hereditary classes of graphs, the class of complete graphs and the class of empty (edgeless) graphs. In this section, we prove a similar result for augmenting graphs and show that there are precisely three minimal infinite hereditary classes of connected augmenting graphs. These classes are as follows (see also Fig. 2.1).

1. Chordless paths of even length $\{P_{2k+1} : k \in \mathbb{N}\}$
2. Complete bipartite graphs $\{K_{k,k+1} : k \in \mathbb{N}\}$
3. Simple augmenting trees A_k, i.e. graphs formed from a star $(K_{1,k})$ by subdividing each edge exactly once



(a) Path (b) Complete Bipartite (c) Simple Augmenting Tree

Figure 2.1: The three special families of augmenting graphs.

We will show that in any hereditary class where the set of possible augmenting graphs does not fully contain any of these three families, the size of minimal connected augmenting graphs of increment 1 in the class will be bounded, in which case the Maximum Independent Set problem can be solved in polynomial time.

Augmenting paths, also known as augmenting chains, were first introduced in [Berge, 1957]. In the class of *claw*-free graphs, all connected augmenting graphs are paths. It is easy to show that in the class of co-graphs (P_4-free graphs), all connected augmenting graphs must be complete bipartite. Simple augmenting trees were introduced in [Mosca, 1999] to solve the Maximum Independent Set problem in (P_6, C_4)-free graphs.

We denote an induced matching with p edges by M_p. Also, we let $Rb(s,t)$ be the *non-symmetric bipartite Ramsey number*. That is, we define $Rb(s,t)$ to be the minimum number such that if G is a bipartite graph with at least $Rb(s,t)$ vertices in each part then either G contains $K_{s,s}$ as an induced subgraph or the bipartite complement of G contains $K_{t,t}$ as an induced subgraph. We start with a useful Lemma.

Lemma 5. *For any natural numbers t and p, there is a number $N(t,p)$ such that every bipartite graph with a matching of size at least $N(t,p)$ contains either a bi-clique $K_{t,t}$ or an induced matching M_p.*

Proof. For $p = 1$ and arbitrary t, we can define $N(t,p) = 1$. Now, for each fixed t, we prove the lemma by induction on p. Without loss of generality, we prove it only for values of the form $p = 2^s$ (since if the graph contains an induced matching of size r, it contains an induced matching of size $r-1$). Suppose we have shown the lemma for $p = 2^s$ for some $s \geq 0$. Let us now show that it is sufficient to set $N(t, 2p) = Rb(t, Rb(t, N(t,p)))$.

Consider a graph G with a matching of size at least $Rb(t, Rb(t, N(t,p)))$. Without loss of generality, we may assume that G contains no vertices outside of this matching. We also assume that G does not contain an induced $K_{t,t}$, since otherwise we are done. Then G must contain the bipartite complement of a $K_{Rb(t,N(t,p)),Rb(t,N(t,p))}$ with vertex classes, say, A and B. Now let C and D consist of the vertices matched to vertices in A and B respectively in the original matching in G.

Note that A, B, C, D are pairwise disjoint. $G[A \cup C]$ and $G[B \cup D]$ now each contain a matching of size $Rb(t, N(t,p))$. There are no edges between A and B. However there may exist edges between C and D. By our assumption, $G[C \cup D]$ is $K_{t,t}$-free, therefore it must contain the bipartite complement of $K_{N(t,p),N(t,p)}$, with vertex sets $C' \subset C$, $D' \subset D$. Let $A' \subset A$ and $B' \subset B$ be the set of vertices matched to C' and D' respectively in the original matching in G. Now there are no edges in $G[A' \cup B']$ and none in $G[C' \cup D']$, but $G[A' \cup C']$ and $G[B' \cup D']$ both contain a matching of size $N(t,p)$. Since G is $K_{t,t}$-free, by the induction hypothesis, we conclude that they both contain an induced M_p. Putting these together we find that G contains an induced M_{2p}. $\square$

Theorem 6. *Let $\mathcal{C}$ be a class of bipartite graphs closed under isomorphism and under taking induced subgraphs (i.e. closed under vertex-deletion). Let $\mathcal{C}^*$ be the class of connected minimal augmenting graphs of increment 1 in $\mathcal{C}$, i.e. those with $|W| = |B| + 1$. If $\mathcal{C}^*$ is infinite, then it contains one of the following classes (see also Figure 2.1):*

1. *Chordless paths of even length $\{P_{2k+1} : k \in \mathbb{N}\}$*
2. *Complete bipartite graphs $\{K_{k,k+1} : k \in \mathbb{N}\}$*
3. *Simple augmenting trees A_k, i.e. graphs formed from a star ($K_{1,k}$) by subdividing each edge exactly once*

Proof. Suppose the theorem is false, i.e. there is a class $\mathcal{C}$ of bipartite graphs such that $\mathcal{C}^*$ is infinite, but there is a t such that $\mathcal{C}^*$ does not contain any P_t, $K_{t,t+1}$ or A_t. The graphs in $\mathcal{C}^*$ are connected, but are P_t-free, so there must be graphs in $\mathcal{C}^*$ with vertices of degree at least $N(t,t) + 2$.

Consider a graph $G = (B, W, E)$ in $\mathcal{C}^*$. For any proper subset $W' \subsetneq W$, we must have $|N_B(W')| \geq |W'|$, since otherwise $(N_B(W'), W', E \cap (N_B(W') \times W'))$ would be a smaller augmenting graph, contradicting the minimality of G. By Hall's Marriage Theorem, there must be a matching M from B to W (one vertex of W will not be matched to any vertex of B since $|W| = |B| + 1$).

Now let $G = (B, W, E)$ be any graph in $\mathcal{C}^*$ containing a vertex x of degree at least $N(t,t)+2$. Let X be the set of vertices in the neighbourhood of x which form part of the matching M, but are not matched with x. X must contain at least $N(t,t)$ vertices. Let Y be the set of vertices which M matches to the vertices of X. Then $G[X \cup Y]$ contains a matching of size $N(t,t)$, but is $K_{t,t}$-free. This means that it must contain an induced matching on t edges. Let Z be the set of vertices that occur in this induced matching. Then $G[Z \cup \{x\}]$ forms an A_t, so $A_t \in \mathcal{C}$ and therefore $A_t \in \mathcal{C}^*$. This contradiction completes the proof. □

Clearly, when using the augmenting graph technique for finding maximum independent sets, we need only consider minimal, connected augmenting graphs of increment 1. If there is some t such that our graph class is $(K_{t,t}, P_t, A_t)$-free then there are at most finitely many such augmenting graphs (up to isomorphism), which leads to the following result:

Corollary 7. *For positive integers i, j, k, the* Maximum Independent Set *problem can be solved in polynomial time in the class of $(P_i, K_{j,j+1}, A_k)$-free graphs.*

It should be noted that the proofs in this section do yield an upper bound on the size of any minimal, connected augmenting graphs of increment 1 in the class of $(P_i, K_{j,j+1}, A_k)$-free graphs. However, this result is only of theoretical interest, because even for small i, j, k, the resulting bounds are much to large to be of use for practical algorithms.

In the remainder of this chapter, we demonstrate a family of subclasses of P_5-free graphs where the Maximum Independent Set problem can be solved in polynomial time using augmenting graphs. Clearly, P_5-free graphs are A_t-free for $t \geq 2$ and P_t-free for $t \geq 5$, but they are not $K_{t,t+1}$-free for any t. Thus, we need to deal with the fact that there may be infinitely many possible connected minimal augmenting graphs of increment 1.

2.3 Augmenting graphs in P_5-free graphs

We say that a bipartite graph H is *chain bipartite* if, for any two vertices x and y in the same part of H, either $N(x) \subseteq N(y)$ or $N(y) \subseteq N(x)$. Clearly, any chain bipartite graph must be P_5-free. It is easy to prove (see [Gerber et al., 2003]) that every connected P_5-free bipartite graph must be a chain bipartite graph. Thus, we get the following conclusion:

Lemma 8. *A connected augmenting graph is P_5-free if and only if it is a chain bipartite graph.*

We can describe such graphs with the following notation: For positive integers $d_1, \ldots, d_k$, with $d_1 \geq d_2 \geq \ldots \geq d_k$, let $B_k(d_1, \ldots, d_k)$ be the bipartite graph with parts $B = \{b_1, \ldots, b_k\}$ and $W = \{w_1, \ldots, w_{d_1}\}$ such that w_i is adjacent to b_j if and only if $i \leq d_j$. Note that $B_k(d_1, \ldots, d_k)$ is a chain bipartite graph and that every chain bipartite graph can be uniquely (up to isomorphism) described in this way. With this notation, we can rewrite Lemma 8 as follows:

Lemma 9. *A connected augmenting graph on at least 2 vertices is P_5-free if and only if it is isomorphic to a graph of the form $B_k(d_1, \ldots, d_k)$ with $k < d_1 \geq d_2 \geq \ldots \geq d_k > 0$.*

The following two lemmas provide more useful information:

Lemma 10. *Suppose $H = G[W \cup B]$ is a minimal connected augmenting graph for a maximal (with respect to set inclusion) independent set S. Then each vertex of B has at least 2 neighbours in W.*

Proof. Since H is connected, each vertex of B must have at least one neighbour in W. Suppose B contains a vertex x which has exactly one neighbour y in W. Then $H - \{x, y\}$ is also an augmenting graph for S and has the same increment as H, contradicting the minimality of H. $\square$

Lemma 11. *Suppose $B_k(d_1, \ldots, d_k) = G[W \cup B]$ is an augmenting graph for a maximal (with respect to set inclusion) independent set S in G. If G does not contain any augmenting $K_{1,2}$ then $k > 1$ and $d_2 \geq d_1 - 1$.*

Proof. We know that $k > 1$ since otherwise $G[b_1, w_1, w_2]$ would be an augmenting $K_{1,2}$. Similarly, if $d_2 < d_1 - 1$ then $G[b_1, w_{d_1}, w_{d_1-1}]$ would be an augmenting $K_{1,2}$. $\square$

2.3.1 A Class of Augmenting Graphs

Fix $z \geq 4$. Let $K_{3,z} - e$ be the graph obtained from the graph $K_{3,z}$ by deleting an edge. This is the same graph as that described by $B_3(z, z, z-1)$ and $B_z(3, 3, \ldots, 3, 2)$ (with $(z-1)$ 3's).

Lemma 12. *Let $z \geq 4$ and let G be a $(P_5, K_{3,z}-e)$-free graph, containing a maximal (with respect to set inclusion) independent set S. Suppose $H = B_k(d_1, \ldots, d_k) = G[W \cup B]$ is a connected minimal augmenting graph for S. Then one of the following must hold:*

1. *G contains an augmenting graph for S on at most $4z+1$ vertices,*
2. *$H = B_k(\ell, \ell, \ldots, \ell)$ for some k, ℓ (i.e. a $K_{k,\ell}$)*
3. *$H = B_k(\ell+1, \ell, \ell, \ldots, \ell)$ for some k, ℓ (i.e. a $K_{k,\ell}$ with a pendant white vertex).*

Proof. First, we may assume that G does not contain an augmenting graph for S on at most $4z+1$ vertices, in which case there is no augmenting $K_{1,2}$. It also means that we must have $d_1 > 2z, k \geq 2z$. By Lemma 11, $d_2 \geq d_1 - 1 \geq 2z$ and $d_k > 0$.

Suppose that $d_z \leq z-1$ then $G[b_1, \ldots, b_z, w_z, \ldots, w_{2z}]$ would be an augmenting graph on $2z+1$ vertices. Therefore $d_z \geq z$. If $d_{z+i} < d_z$ for some $i > 0$ then $G[b_1, b_2, \ldots, b_{z-1}, b_{z+i}, w_1, w_2, w_{d_z}]$ would be a $K_{3,z} - e$ (since $z \geq 4$), so we must have $d_z = d_{z+1} = \cdots = d_k$. This means that $d_i \geq z$ for $i \in \{1, \ldots, k\}$. Now suppose, for contradiction, that $d_i < d_{i-1}$ for some $i \geq 3$. Then $G[b_1, b_2, b_i, w_1, \ldots, w_{z-1}, w_{d_{i-1}}]$ would be a $K_{3,z} - e$. This shows that $d_2 = d_3 = \cdots = d_z$. Combined with the fact that $d_1 \geq d_2 \geq d_1 - 1$, this completes the proof of the lemma. □

2.3.2 An Augmenting Graph Algorithm

We now proceed in a similar way to the case of $(P_5, K_{3,3} - e)$-free graphs in [Lozin and Mosca, 2009]. (That proof assumes that all augmenting graphs on 7 vertices have been applied. For our proof, we increase this to $4z+1$ vertices and then use similar arguments.)

Let G be a $(P_5, K_{3,z} - e)$-free graph containing an independent set S. Without loss of generality, we will assume that G contains no augmenting graphs for S with at most $4z+1$ vertices, since we can apply all such augmentations in polynomial time. We will now construct an augmenting graph of maximum possible increment. Applying this augmenting graph to the set S will yield an independent set of maximum size in G.

If two white vertices x and y have the same set of black neighbours, we say they are *similar*. We start by partitioning the white vertices into similarity classes. We say that a white vertex x is *light* if it has exactly one black neighbour. Otherwise, we say that x is *heavy*. Each class of light vertices must be a clique (since there is no augmenting $K_{1,2}$). We may assume that every heavy vertex must have at least $2z+1$ black neighbours. Indeed, if a heavy vertex x has less than $2z+1$ black neighbours, then by Lemma 12, either there is an augmenting graph on at most $4z+1$ vertices or x is not in any minimal augmenting graph. Since there are no augmenting graphs with at most $4z+1$ vertices, if a heavy vertex has less than z black neighbours, we can safely delete it from G.

If G has no light similarity classes, then all augmenting graphs in G must be P_4-free, in which case we proceed as in [Boliac and Lozin, 2003], which finds a P_4-free augmenting graph of maximum increment in any P_5-free graph (not necessarily $(K_{3,z}-e)$-free). From now on, we can therefore assume that G contains light similarity classes.

Let C be a heavy similarity class. We say that a light vertex x is *C-attached* if $N_S(x) \subseteq N_S(C)$. For each C-attached vertex x, let $C(x)$ denote the subset of vertices of C non-adjacent to x. We partition $C(x)$ into co-components, i.e. sets of vertices which form components in the complement of $G[C(x)]$. We call each such co-component a *node class of C associated with x*. If no light vertices are attached to C, then the node classes of C are its co-components. Note that, by definition, any two node classes are disjoint if they are associated with the same light vertex. We now show that any two node classes are disjoint, regardless of which (if any) light vertices they are associated with.

Lemma 13. *[Lozin and Mosca, 2009] Let C be a heavy similarity class. If C^1 and C^2 are two distinct node classes of C, then they are disjoint.*

Proof. Suppose, for contradiction, that C^1 and C^2 have non-empty intersection $C^{12} := C^1 \cap C^2$, which contains a vertex u. Since $C^1 \neq C^2$, without loss of generality, we may assume $C^{11} := C^1 \setminus C^2$ is non-empty and contains a vertex v. If C has no attached light vertices then the lemma follows immediately from the definition of node class. In the same way, we find that C^1 and C^2 cannot be associated with the same light vertex. Let C^1 be associated with x and C^2 be associated with y.

We may assume that u and v are nonadjacent, since they both belong to the node class C^1. Indeed, if every vertex in C^{11} is adjacent to every vertex in C^{12}, then C^1 can be partitioned into co-components, contradicting the definition of node class.

This means that y must be adjacent to v. If not, then $v \in C(y)$, while $v \notin C^2 \subseteq C(y)$. The only way this could happen would be if v were adjacent to every vertex in C^2, which cannot be the case, since v is not adjacent to u. Thus y must indeed be adjacent to v.

If x is adjacent to y, then letting z be a black neighbour of u and v, but not of

x and y would mean that $G[x, y, v, z, u]$ would form a P_5. Thus x cannot be adjacent to y. Since each light class is a clique, x and y must have different black neighbours, say a and b, respectively. But then $G[x, a, u, b, y]$ would be a P_5, which is a contradiction. $\square$

Let C^0 denote the subset of vertices in C which are adjacent to *every* C-attached vertex. To make the terminology consistent, we shall say that C^0 is also a node class. To distinguish C^0 from the normal node classes of C, we shall call it the *specific* node class of C. With this new node class, Lemma 13 tells us that the node classes form a partition of C.

Suppose that S admits an augmenting graph. Let H be an augmenting graph of maximum increment. Without loss of generality, we may assume that every connected component of H forms a minimal augmenting graph, i.e. by Lemma 12, it is either a $K_{k,\ell}$ for some k, ℓ or it is a $K_{k,\ell}$ with a pendant white vertex. It is easy to see that if the pendant white vertex is present, it must be a vertex from a light class and all the other white vertices must be from heavy classes. Clearly, all the heavy vertices from one component of H must belong to the same heavy class. In fact, a stronger statement holds:

Lemma 14. *If S is not a maximum independent set, then S admits a maximum augmenting graph H such that for any component of H, the set of heavy vertices in this component belong to the same node class.*

Proof. Consider a component H' of a maximum augmenting graph H for S. Let W' be the set of heavy vertices in H' and let C be the heavy similarity class containing the vertices of W'.

First, we consider the case where the class C^0 contains at least 2 vertices of W', say x and y. Assume that $W' \nsubseteq C^0$ and let z' be a vertex in a non-specific node class C^j. Let a be a C-attached vertex such that C^j is associated with a (i.e. $z' \in C^j \subseteq C(a)$). Let $b_1, \ldots, b_{z-1}$ be black vertices in the neighbourhood of C, non-adjacent to a. But now $G[a, b_1, \ldots, b_{z-1}, x, y, z']$ is isomorphic to $K_{3,z} - e$, which is a contradiction. Thus if C^0 contains at least 2 vertices of W' then $W' \subseteq C^0$.

From now on, we may assume that the specific class C^0 contains at most one vertex of W'. Since H' contains more than $4z + 1$ vertices, we know that W' must contain at least $2z + 1$ vertices. We consider two cases:

Case 1: Three vertices x, y, z' of W' belong to at least two different non-specific node classes, one of which we will denote by C^i. Without loss of generality, let $x \in W' \cap C^i, y \notin W' \cap C^i, z' \notin W' \cap C^i$. Let a be a C-attached light vertex such that C^i is associated with a. Then x must be non-adjacent to a. Conversely, since y is not adjacent to x and $y \notin C^i \subseteq C(a)$, we must have $y \notin C(a)$, in which case y must be adjacent to a. Similarly, z' must be adjacent to a. Let $b_1, \ldots, b_{z-1}$ be $z-1$ black vertices adjacent to y and z', but not a. Then $G[a, b_1, \ldots, b_{z-1}, x, y, z']$ is a $K_{3,z} - e$, which is a contradiction.

Case 2: All vertices of B (with at most one exception, which belongs to C^0) are in the same non-specific node class, say C^i. Let a be the C-attached light vertex such that C^i is associated with a. If $B \cap C^0 = \emptyset$ then we are done. If C^0 contains a vertex of B, say x, then a cannot be in H since x and a are adjacent. But now a cannot have a neighbour in H outside H', otherwise a P_5 would arise (let $a_1 \in V(H) \setminus V(H')$ be a neighbour of a, $a_2 \in N_S(a)$, $a_3 \in B \setminus \{x\}$, $a_4 \in N_S(C^i) \setminus N_S(a)$ then $G[a_1, a, a_2, a_3, a_4]$ would be a P_5). But now we can replace x by a in H, which produces a new augmenting graph of the same increment where the vertices of our modified component now satisfy the lemma. □

By using the above lemma and the fact that S does not admit an augmenting graph on less than $4z + 1$ vertices, without loss of generality we may assume that:

- (*) Every heavy node class contains an independent set on at least $2z$ vertices.

Indeed, if some node classes have less than z vertices, then there is a maximum augmenting graph which does not contain any of these vertices. We may thus safely delete any vertices in such node classes. We now prove the following lemma:

Lemma 15. *Assuming (*), let C^i and C^j be two node classes which are not similar. If $N_S(C^i) \cap N_S(C^j) = \emptyset$ then no vertex in C^i is adjacent to a vertex in C^j.*

Proof. First suppose that every vertex in C^i is adjacent to every vertex in C^j. Then any $z-1$ non-adjacent vertices in C^i, two nonadjacent vertices in C^j, any single vertex in $N_S(C^j)$ and any single vertex in $N_S(C^i)$ form a $K_{3,z} - e$, which is a contradiction.

Now suppose that $x \in C^i$ has both a non-neighbour $y \in C^j$ and a neighbour $w \in C^j$. We may assume that w and y are nonadjacent. (If they were adjacent, then they would be adjacent in $\overline{G}$, so by definition of C^j they must be connected by a path in $\overline{G[C^j]}$. But now we can replace w and y by vertices on this path with the required property.) However, for $a \in N_S(C^i)$ and $b \in N_S(C^j)$, we find that $G[a, x, w, b, y]$ is a P_5. This contradiction completes the proof. □

We now associate an *auxiliary* graph Γ with G and S. The vertices of Γ are the node classes C^i. Two vertices C^i and C^j in Γ are adjacent if and only if $N_S(C^i) \cap$

$N_S(C^j) \neq \emptyset$. By Lemma 15, C^i and C^j are nonadjacent if and only if both $N_S(C^i) \cap N_S(C^j) = \emptyset$ and no vertex in C^i is adjacent to any vertex in C^j.

We put an integer *weight* $w(C^j)$ on each vertex C^j of Γ as follows: if C^j is a specific node class then $w(C^j) = \alpha(G[C^j]) - |N_S(C^j)|$ and if C^j is a non-specific class, then $w(C^j) = \alpha(G[C^j]) + 1 - |N_S(C^j)|$, where the value of $\alpha(G[C^j])$ will be calculated recursively (see later).

Let $Q = \{v_1, \ldots, v_p\}$ be an independent set in Γ. With each vertex v, we associate an independent set I_v of maximum cardinality in the node class represented by v. Let $H(v)$ be the bipartite graph whose black vertices $H^B(v) := N_S(I_v)$ and whose white vertices $H^W(v)$ are defined as follows. If v represents a specific node class C^i then $H^W(v) := I_v$, i.e. $H(v)$ is a complete bipartite graph. If v represents a non-specific node class C^i then find any light vertex a such that C^i is associated with a, and define $H^W(v) := I_v \cup \{a\}$, i.e. $H(v)$ is a complete bipartite $K_{k,\ell}$ with a pendant white vertex a.

By definition of Γ, the sets $H^B(v_1), \ldots, H^B(v_p)$ are pairwise disjoint. Using Lemma 15 and the fact that G is P_5-free, we find that $\cup_{i=1}^{p} H^W(v_i)$ is an independent set. Let H_Q denote the union of the graphs $H(v_i)$. This is a bipartite graph, whose increment coincides with the weight of Q. If the weight of Q is positive, then H_Q is an augmenting graph for S. Furthermore, if Q is an independent set of maximum total weight, then H_Q is a maximum augmenting graph. We can thus use the following recursive procedure to solve the maximum independent set problem in a $(P_5, K_{3,z} - e)$-free graph.

Algorithm	MIS(G)
Input:	A $(P_5, K_{3,z} - e)$-free graph G
Output:	An independent set S of maximum size in G

1. Find an arbitrary maximal independent set S in G.

2. Keep applying H-augmentations for augmenting graphs H on at most $4z+1$ vertices as long as such graphs are present.

3. Partition the vertices in $V(G) \setminus S$ into similarity classes. Delete any heavy vertices with less than z black neighbours. Partition the vertices of each heavy class into node classes.

4. In each node class C^i find a maximum independent set $S(C^i)$ =MIS($G[C^i]$). Delete any node classes where $|\text{MIS}(G[C^i])| < z$.

5. Construct the auxiliary graph Γ and find an independent set Q of maximum weight in it.

6. If the weight of Q is positive, construct the augmenting graph H_Q and enlarge S by applying a H_Q-augmentation.

7. Return S and STOP.

The recursion in Step 4 above applies to disjoint subgraphs of G, so to prove that the algorithm runs in polynomial time, it is sufficient to prove that all of the other steps run in polynomial time. This is easy to see for all steps apart from Step 5. To show that Step 5 can be done in polynomial time, we use the following observation:

Lemma 16. *The graph Γ is (P_4, C_4)-free.*

Proof. Suppose, for contradiction, that Γ contains a P_4 or C_4 on vertices C^1, C^2, C^3, C^4, with edges C^1C^2, C^2C^3, C^3C^4 and non-edges C^1C^3, C^2C^4 (C^1C^4 may or may not be an edge). Note that if $N_S(C^i) \cap N_S(C^j) \neq \emptyset$ then either $N_S(C^i) \subseteq N_S(C^j)$ or $N_S(C^j) \subseteq N_S(C^i)$. Indeed, if $a_1 \in N_S(C^i) \setminus N_S(C^j), a_2 \in C_i, a_3 \in N_S(C^i) \cap N_S(C^j), a_4 \in C^j, a_5 \in N_S(C^j) \setminus N_S(C^i)$, then $G[a_1, a_2, a_3, a_4, a_5]$ would be a P_5, contradicting the fact that G is P_5-free. Without loss of generality, we may assume that $N_S(C^2) \subseteq N_S(C^3)$. But C^1 and C^3 are not adjacent, so $N_S(C^1) \cap N_S(C^3) = \emptyset$. Thus $N_S(C^1) \cap N_S(C^2) = \emptyset$, which contradicts the fact that C^1C^2 is an edge in Γ. This completes the proof. □

Graphs which are (P_4, C_4)-free graphs are also known as trivially perfect [Golumbic, 1978] or quasi-threshold [Jing-Ho et al., 1996] graphs and have been much studied in the literature. Finding an independent set of maximum weight can be solved in linear time in the class of P_4-free graphs using their “co-tree” structure [Corneil et al., 1981]. (This is a simplified version of modular decomposition. Modular decomposition is discussed in more detail in Chapter 3). Summarising all of the above, we conclude the following theorem:

Theorem 17. *The maximum independent set problem is solvable in polynomial time in the class of $(P_5, K_{3,z} - e)$-free graphs.*

2.4 Conclusion

In this chapter we studied the use of augmenting graphs to find maximum independent sets. We proved a Ramsey-type result on classes of augmenting graphs. We also showed that augmenting graphs can be used to solve Maximum Independent Set in polynomial time in the class of $(P_5, K_{3,z} - e)$-free graphs. The complexity of Maximum Independent Set in P_5-free graphs remains a very challenging open question.

Chapter 3

Parameterized Algorithms for Finding Independent Sets

3.1 Introduction

One approach to deal with NP-hard problems is based on the notion of fixed-parameter tractability (fpt), which is a relaxation of classical polynomial-time solvability. A parameterized problem is said to be *fixed-parameter tractable* if it can be solved in time $f(k)p(n)$ on instances of input size n, where $f(k)$ is a computable function depending only on the value of the parameter k and $p(n)$ is a polynomial independent of k. Unfortunately, if k is the independence number, the Maximum Independent Set problem remains hard even under this relaxation. More formally, it is W[1]-hard [Downey and Fellows, 1999]. However, for graphs in some restricted families the problem becomes fixed-parameter tractable. In particular, this is true for graphs without large cliques, which follows from a simple Ramsey argument (see e.g. [Raman and Saurabh, 2006]). This argument alone implies fixed-parameter tractability of the problem for graphs of bounded degree, of bounded degeneracy, of bounded chromatic number, in all proper minor-closed graph classes (which includes, in particular, classes of graphs excluding single-crossing graphs as minors [Demaine et al., 2005]) and all proper classes closed under taking subgraphs (not necessarily induced). Beyond this argument, very little is known about the parameterized complexity of the problem in restricted graph families. Other classes where the problem is known to be fixed-parameter tractable are the complements of t-multiple-interval graphs [Fellows et al., 2009] and segment intersection graphs with a bounded number of directions [Kára and Kratochvíl, 2006].

We develop fpt-algorithms that solve the Maximum Independent Set problem in several new classes of graphs, generalising some of the previously known results. In fact, our results apply to a natural generalisation of the problem for weighted graphs. We say that a graph G is a weighted graph if each vertex of G is assigned a real number ≥ 1, the weight of the vertex. The Maximum Weighted Independent Set problem is that of finding an independent set of maximum weight in a weighted graph, where the weight of a set of vertices is the sum of the weights of its elements. This maximum weight is denoted $\alpha_w(G)$. We study the following parameterization of the Maximum Weighted Independent Set problem:

Weighted Independent Set

Instance:	A weighted graph G with weight function $w : V(G) \to \mathbb{R}_{\geq 1}$ and a positive real number W.
Parameter:	W.
Problem:	Decide whether G has an independent set of weight at least W and find such a set if it exists. If no such set exists, find an independent set of weight $\alpha_w(G)$ instead.

3.2 $(K_r - e)$-free graphs

As we noted above, a simple Ramsey argument implies the fixed-parameter tractability of Maximum Independent Set in K_r-free graphs. We first extend this result to the weighted case.

Theorem 18. *For $r \in \mathbb{N}$, the* Weighted Independent Set *problem is fixed-parameter tractable in the class of K_r-free graphs.*

Proof: Let (G, W) be an instance of the Weighted Independent Set problem, with G being a K_r-free graph on n vertices. Since the weight of each vertex is ≥ 1, the weight of every independent set is at least its size. Therefore, if G has at least $R(\lceil W \rceil, r)$ vertices, then it necessarily has an independent set of size (and therefore of weight) at least W. If the number of vertices of G is at least $R(\lceil W \rceil, r)$, we can delete any $n - R(\lceil W \rceil, r)$ vertices from G, since the remaining graph still necessarily has an independent set of weight at least W. Now the number of vertices of G is at most $R(\lceil W \rceil, r)$, so the problem can be solved in time independent of n. This implies the fixed-parameter tractability of Weighted Independent Set for K_r-free graphs. □

Since K_{r-1} is an induced subgraph of $K_r - e$, our next result generalises Theorem 18.

Theorem 19. *For $r \in \mathbb{N}$, $r \geq 2$, the* Weighted Independent Set *problem is fixed-parameter tractable in the class of $(K_r - e)$-free graphs.*

Proof: Let (G, W) be an instance of the WEIGHTED INDEPENDENT SET problem, with G being a $(K_r - e)$-free graph on n vertices. Let I be an independent set of G such that I is maximal with respect to set-inclusion and there are no two non-adjacent vertices u and v in $V(G) \setminus I$ for which $(N_G(u) \cup N_G(v))$ contains exactly one vertex of I, (i.e. I admits no *augmenting* K_1 or *augmenting* $K_{1,2}$). Clearly, if one of these two conditions fails, one can immediately construct a larger independent set. This implies that a set with these properties can be found in time polynomial in n. Since the vertices of the graph have weights ≥ 1, if we find an independent set of size $\geq W$, then returning this set correctly solves WEIGHTED INDEPENDENT SET. Hence we suppose $|I| < W$. (If this happens, the procedure actually solves the MAXIMUM WEIGHTED INDEPENDENT SET problem.)

We partition the vertices in $V(G) \setminus I$ into *classes* according to their neighbourhood in I, i.e. two vertices of $V(G) \setminus I$ belong to the same class if and only if they have the same neighbours in I. A class is *light* if its elements have exactly one neighbour in I and *heavy* otherwise.

By the choice of I, each light class is a clique and hence any independent set in G contains at most one vertex from each of the $|I|$ light classes. Furthermore, no vertex u from a light class has $r - 2$ neighbours in another light class, since otherwise a $K_r - e$ arises using u, some $r - 2$ neighbours of u in another light class, and their unique neighbour in I.

Since G is $(K_r - e)$-free, every heavy class C induces a K_{r-2}-free graph, since otherwise a clique K of order $r - 2$ in C together with two neighbours in I of the vertices in K would form a $K_r - e$. Hence, if some heavy class contains at least $R(\lceil W \rceil, r - 2)$ vertices, we can find an independent set of size at least W as explained in the proof of Theorem 18. Therefore, we suppose that each heavy class contains less than $R(\lceil W \rceil, r-2)$ vertices, which implies that the union H of I and all the heavy classes contains at most $(W - 1) + 2^W R(\lceil W \rceil, r - 2)$ vertices, which is bounded in terms of W and r.

We can now proceed as follows:

Step 1: Generate all independent sets contained in H. Clearly, the number of such sets and the time needed to generate all of them is bounded in terms of W and r. For each independent set I_H found in this step, execute Step 2.

Step 2: Let L denote the set of vertices u in light classes such that u has no neighbour in I_H. Let L_1 denote the set of vertices in L that belong to light classes C with $|C \cap L| < r\lceil W \rceil$. Furthermore, let $L_2 \subseteq L$ contain the $r\lceil W \rceil$ vertices of largest weight (breaking ties arbitrarily) in $C \cap L$ for each light class C with $|C \cap L| \geq r\lceil W \rceil$.

Note that $L_1 \cup L_2$ contains at most $r\lceil W \rceil^2$ vertices, which is bounded in terms of W and r. Therefore, we can determine an independent set $I_L \subseteq L_1 \cup L_2$ such that $I_H \cup I_L$ is of largest possible weight, in time bounded in terms of W and r.

Let J be an independent set of G with $J \cap H = I_H$ such that J has maximum possible weight and, subject to this condition, J has largest possible intersection with I_L. Let $J_L = J \setminus H$. Since $J \cap H = I_H$ and J is independent, we have $J_L = J \cap L$. We claim that $J_L = I_L$. For contradiction, we assume that $J_L \neq I_L$. In this case, the choice of I_L and J implies that J_L must contain a vertex $x \in L \setminus (L_1 \cup L_2)$. Note that x necessarily belongs to a light class C with $|C \cap L| \geq r\lceil W \rceil$. Since there are less than W vertices in $J_L \setminus \{x\}$ and every vertex in a light class has less than $r-2$ neighbours in C, the set $C \cap L_2$ contains a vertex x' that is not adjacent to any vertex in $J_L \setminus \{x\}$. By the choice of L_2, the weight of x' is at least the weight of x. Therefore, the set $(J \setminus \{x\}) \cup \{x'\}$ is independent, has at least the weight of J and a larger intersection with I_L than J, which contradicts the choice of J. This proves $J_L = I_L$, which means that the set $I_H \cup I_L$ found in the second step is an independent set of maximum weight intersecting H in I_H. Since we execute the second step for all possible choices of I_H, returning a set of the form $I_H \cup I_L$ that is of largest possible weight correctly solves WEIGHTED INDEPENDENT SET. Clearly, the running time of the sketched procedure is $f_r(W)p(n)$ where, for fixed r, $f_r(W)$ is a computable function depending on W and $p(n)$ is a polynomial independent of W. $\square$

Note that the polynomial $p(n)$ above is independent of r as well as independent of W, so the problem is fixed-parameter tractable even if parameterized by both W and r.

3.3 Splittable graphs

In this section, we consider graphs that allow a certain type of decomposition; either of its vertex set or of its edge set.

Definition 20. *For $r \in \mathbb{N}$ and a graph G, a partition $V(G) = X \cup Y$ of the vertex set of G is an* r-split partition of G *if $\omega(G[X]) < r$ and $\alpha(G[Y]) < r$. If a graph G has an r-split partition, then G is an* r-split graph.

The notion of r-split graphs generalises K_r-free graphs and many other important hereditary classes. To see the importance of this notion, observe that for every hereditary class X (see e.g. [Balogh et al., 2000]), there is a natural number k (called the *index* for the class) such that the number X_n of n-vertex labelled graphs (also known as the *speed* of X) satisfies $\lim_{n\to\infty} \frac{\log_2 X_n}{\binom{n}{2}} = 1 - \frac{1}{k(X)}$, Furthermore, if $\mathcal{E}^{i,j}$ denotes the class of graphs whose vertices can be partitioned into at most i independent sets and j cliques, then the index $k(X)$ of a class X is the maximum k such that X contains a class $\mathcal{E}^{i,j}$ with $i + j = k$. In other words, the classes $\mathcal{E}^{i,j}$ with $i + j = k$ are the only minimal classes of index k. Therefore, any class X of index > 1 can be approximated by a minimal class $\mathcal{E}^{i,j}$ of the same index, in the sense that $\lim_{n\to\infty} \frac{\log_2 X_n}{\mathcal{E}_n^{i,j}} = 1$. Clearly, $\mathcal{E}^{i,j}$ is a subclass of $\max\{i + 1, j + 1\}$-split graphs.

Note that the class of split graphs (i.e. graphs partitionable into an independent set and a clique) is exactly the class $\mathcal{E}^{1,1}$ and that the graphs in this class are precisely the 2-split graphs. Among various nice properties, split graphs admit polynomial-time recognition. In the next lemma we show that this property extends to r-split graphs for all values of r.

Lemma 21. *For every $r \in \mathbb{N}$, the class of r-split graphs can be recognised in polynomial time, and a certifying r-split partition of the vertex set can be constructed within this time (where r is a constant which is not part of the input).*

Proof: Let $G = (V, E)$ be a graph and Y an arbitrary subset of its vertices with $\alpha(G[Y]) < r$. It is not difficult to see that in polynomial time one can check if G contains a set Y' such that

(1) $|Y \setminus Y'| < R(r, r)$, $\alpha(G[Y']) < r$ and $|Y'| = |Y| + 1$.

As long as G admits such a set Y', replace Y with Y', i.e. set $Y := Y'$. If no such set can be found, then check if G contains a set Y' such that

(2) $|Y \setminus Y'| < R(r, r)$, $|Y' \setminus Y| < R(r, r)$, $\alpha(G[Y']) < r$ and $\omega(G[V \setminus Y']) < r$.

If the answer is affirmative, then obviously G is an r-split graph and $Y' \cup (V \setminus Y')$ is a respective partition. Otherwise, G is not an r-split graph. To see this, suppose for contradiction that G admits an r-split partition $V = X_0 \cup Y_0$ with $\omega(G[X_0]) < r$ and $\alpha(G[Y_0]) < r$. By the choice of Y, the graph $G[Y \setminus Y_0]$ is $\overline{K}_r$-free. Also, since $Y \setminus Y_0$ is a subset of X_0, the graph $G[Y \setminus Y_0]$ is K_r-free. Therefore $|Y \setminus Y_0| < R(r, r)$. If additionally $|Y_0 \setminus Y| < R(r, r)$, then $Y' = Y_0$ satisfies (2), contradicting our assumption. If $|Y_0 \setminus Y| \geq R(r, r)$, then $|Y_0| > |Y|$ in which case a subset $Y' \subset Y_0$ satisfying (1) can be found. A contradiction in both cases proves correctness of the procedure. The polynomiality follows from the fact that r and $R(r, r)$ are constants independent of the number of vertices in G. □

Now we proceed to algorithms that solve the Weighted Independent Set problem for r-split graphs. For $r = 2$ the problem is known to be solvable in polynomial time, since it is a subclass of perfect graphs. However, for large values of r the problem is NP-hard. In the next theorem we show that the problem is fixed-parameter tractable in the class of r-split graphs for any value of r. Since K_r-free graphs are r-split graphs, our result generalises Theorem 18.

Theorem 22. *For $r \in \mathbb{N}$, the* Weighted Independent Set *problem is fixed-parameter tractable in the class of r-split graphs.*

Proof: Let (G, W) be an instance of the Weighted Independent Set problem with G an r-split graph. First, we apply Lemma 21 in order to find a partition $V(G) = X \cup Y$ such that $G[X]$ is K_r-free and $G[Y]$ is $\overline{K}_r$-free. This takes polynomial time. Since $G[Y]$ is $\overline{K}_r$-free, the graph $G[Y]$ has only polynomially many independent sets. For each such set I_Y of weight $w(I_Y)$, we solve the Weighted Independent Set problem for the instance $(G[X \setminus N_G(I_Y)], W - w(I_Y))$ using Theorem 18, which yields a set $I_X(I_Y)$. Returning an independent set of the form $I_Y \cup I_X(I_Y)$ of maximum weight correctly solves Weighted Independent Set. $\square$

The notion of r-split graphs admits a further generalisation as follows:

Definition 23. *Let $r \in \mathbb{N}$ and $\mathcal{G}$ be a hereditary class of graphs. A partition $E(G) = E_0 \cup E_1$ of the edge set of G is an $(r, \mathcal{G})$*-split *if $G_0 = (V, E_0)$ is rK_2-free and $G_1 = (V, E_1)$ belongs to $\mathcal{G}$. If a graph G has an $(r, \mathcal{G})$-split partition, then G is an $(r, \mathcal{G})$*-split graph.

It is not difficult to see that any r-split graph is $(r, Free(K_r))$-split, where $Free(K_r)$ stands for the class of K_r-free graphs. Indeed, let $G = (V, E)$ be an r-split graph with an r-split partition $V = X \cup Y$ where $\omega(G[X]) < r$ and $\alpha(G[Y]) < r$, and let $E_0 \cup E_1$ be a partition of E with $E_1 = E(G[X])$ and $E_0 = E \setminus E_1$. Then obviously $G_1 = (V, E_1)$ is K_r-free. To see that $G_0 = (V, E_0)$ is rK_2-free, observe that in this graph the set X is independent and hence every edge contains at least one of its endpoints in the set Y, which means that if G_0 would contain an induced rK_2, then Y would contain an independent set of size r, which is impossible.

As we saw earlier, for any natural r, the class of r-split graphs enjoys the nice property that graphs in this class can be recognised in polynomial time, which in turn implies fixed-parameter tractability of the Weighted Independent Set problem in this class. This is obviously not true for general $(r, \mathcal{G})$-split graphs. However, as we show below, if $\mathcal{G}$ is a class such that the problem is fixed-parameter tractable in it and an

$(r, \mathcal{G})$-split partition can be found in polynomial time for any $(r, \mathcal{G})$-split graph, then the problem is also fixed-parameter tractable in the class of $(r, \mathcal{G})$-split graphs.

Theorem 24. *Let $r \in \mathbb{N}$ and $\mathcal{G}$ be a hereditary class of graphs. If*

- *the* WEIGHTED INDEPENDENT SET *problem is fixed-parameter tractable in $\mathcal{G}$, and*
- *an $(r, \mathcal{G})$-split partition can be found in polynomial time for any $(r, \mathcal{G})$-split graph,*

then the WEIGHTED INDEPENDENT SET *problem is fixed-parameter tractable in the class of $(r, \mathcal{G})$-split graphs.*

Proof: Given an instance (G, W) of WEIGHTED INDEPENDENT SET with G being an $(r, \mathcal{G})$-split graph, we first apply the polynomial time algorithm to find an $(r, \mathcal{G})$-split partition $E(G) = E_0 \cup E_1$ of the edge set of G such that $G_0 = (V, E_0)$ is rK_2-free and $G_1 = (V, E_1)$ belongs to $\mathcal{G}$. Note that the rK_2-free graph G_0 only has a polynomial number of maximal independent sets [Balas and Yu, 1989], which can all be generated in polynomial time [Tsukiyama et al., 1977], and that a set of vertices is independent in G if and only if it is independent in G_1 and a subset of some maximal independent set of G_0. Therefore, solving the WEIGHTED INDEPENDENT SET problem in $G_1[I_0]$ for each of the polynomially many maximal independent sets I_0 of G_0 and returning an independent set of maximum weight obtained in this way, correctly solves WEIGHTED INDEPENDENT SET on the instance (G, W). Since WEIGHTED INDEPENDENT SET is fixed-parameter tractable in $\mathcal{G}$, the desired result follows. $\square$

3.4 Beyond triangle-free graphs

In the search of further results, in this section we study extensions of triangle-free graphs, which is the simplest nontrivial class of graphs of bounded clique number. We start by analysing H-free graphs, where H is a one-vertex extension of a triangle.

Theorem 25. *For each one-vertex extension H of a triangle, the* WEIGHTED INDEPENDENT SET *problem is fixed parameter tractable in the class of H-free graphs.*

Proof: It is not difficult to see that (up to isomorphism) there are four one-vertex extensions of a triangle: K_4, $K_4 - e$, $K_3 + e$ and $K_3 \cup K_1$, where $K_3 + e$ stands for a triangle plus a pendant edge (also known as a *paw*, see also Figure 6.4) and $K_3 \cup K_1$ denotes the union of a triangle and an isolated vertex.

The fixed-parameter tractability of the problem in the classes of K_4-free graphs and $(K_4 - e)$-free graphs follows from Theorems 18 and 19, respectively.

The structure of (K_3+e)-free graphs has been characterised in [Olariu, 1988] as follows: A connected (K_3+e)-free graph is either triangle-free or a complete multipartite graph (i.e. the complement of the disjoint union of cliques). Together with the trivial observation that the WEIGHTED INDEPENDENT SET problem can be reduced to connected graphs, this proves the theorem for (K_3+e)-free graphs.

Finally, to derive the same conclusion for $(K_3 \cup K_1)$-free graphs, we invoke the obvious fact that a graph G is $(K_3 \cup K_1)$-free if and only if $G - N_G[u]$ is K_3-free for every vertex $u \in V(G)$. Together with the trivial identity

$$\alpha_w(G) = \max_{u \in V(G)} \{\omega(u) + \alpha_w(G - N_G[u])\},$$

the fixed-parameter tractability of the problem in the class of $(K_3 \cup K_1)$-free graphs follows from Theorem 18. □

To further extend one of the classes covered by Theorem 25, we employ the notion of modular decomposition. The idea of modular decomposition was first introduced in the 1960s by Gallai [Gallai, 1967], and also appeared in the literature under various other names such as *prime tree decomposition* [Ehrenfeucht and Rozenberg, 1990], *X-join decomposition* [Habib and Maurer, 1979], or *substitution decomposition* [Möhring, 1985], and this technique has previously been used to construct fpt-algorithms (see e.g. [Protti et al., 2009]). To describe this idea, let us fix some terminology.

Given a graph $G = (V, E)$, a subset of vertices $U \subseteq V$ and a vertex $x \in V$ outside U, we say that x *distinguishes* U if x has both a neighbour and a non-neighbour in U. A subset $U \subseteq V$ is called a *module* of G if no vertex in $V \setminus U$ distinguishes U. A module U is *nontrivial* if $1 < |U| < |V|$, otherwise it is *trivial*. A graph is called *prime* if it has only trivial modules. A module U is *maximal* if $|U| < |V|$ and there is no module U' such that $U \subsetneq U' \neq V$. Maximal modules have the following useful property:

Lemma 26. *[Gallai, 1967] If G is a connected graph whose complement is also connected, the maximal modules of G are pairwise disjoint i.e. they form a partition of the vertex set of G.*

Moreover, from the definition of maximal module, it follows that if U and W are distinct maximal modules, then there are either no edges between them or every vertex in U is adjacent to every vertex in W. Using these properties of maximal modules, we can find a maximum weight independent set in G by

(1) reducing the problem to smaller instances if G or its complement are disconnected,

(2) recursively solving the problem in the subgraphs of G induced by maximal modules,

(3) contracting each maximal module M to a single vertex and assigning to it the weight $\alpha_w(G[M])$, obtaining in this way a new graph G^0,

(4) solving the problem for the graph G^0.

The graph G^0 constructed in step 3 of the outlined procedure is prime. So, the procedure reduces the Maximum Weighted Independent Set problem for any hereditary class to prime graphs in the class. This reduction can be implemented in polynomial time (see e.g. [McConnell and Spinrad, 1999]). Let us show that this is also an fpt-reduction, i.e. it preserves fixed-parameter tractability.

Theorem 27. *Let $\mathcal{X}$ be a hereditary class of graphs and let $\mathcal{X}_0$ denote the class of prime graphs in $\mathcal{X}$. If the* Weighted Independent Set *problem is fixed-parameter tractable in $\mathcal{X}_0$, then it is fixed-parameter tractable in $\mathcal{X}$.*

Proof: Let (G, W) be an instance of the Weighted Independent Set problem with $G \in \mathcal{X}$. Recall that the modular decomposition tree T of G can be determined in linear time [McConnell and Spinrad, 1999; Tedder et al., 2008] and that the set of leaves of T equals the vertex set V of G. To each node v of T we associate the subgraph G_v of G induced by the leaves of the subtree of T rooted at v. Processing the vertices of T in an order of non-increasing height, we will find for each node v of T an independent set I_v of G_v such that the weight $w(I_v)$ of I_v is at least $\min\{W, \alpha_w(G_v)\}$. If the weight of I_v is at least W, we stop the procedure and output I_v. Otherwise, we assign the independent set I_v of weight $\alpha_w(G_v)$ to the node v. The procedure starts by assigning the independent set $I_v = \{v\}$ to each leaf v of T. Now let v be an inner node of T.

If G_v is disconnected, then the children $v_1, v_2, \ldots, v_l$ of v correspond to the connected components of G_v. In this case, we let $I_v = I_{v_1} \cup I_{v_2} \cup \ldots \cup I_{v_l}$.

If the complement of G_v is disconnected, then the children $v_1, v_2, \ldots, v_l$ of v correspond to the connected components of the complement of G_v. In this case we let $I_v = I_{v_i}$, where $w(I_{v_i}) = \max\{w(I_{v_1}), w(I_{v_2}), \ldots, w(I_{v_l})\}$.

Finally, if both G_v and its complement are connected, then the children $v_1, \ldots, v_l$ of v correspond to the subgraphs of G_v induced by the maximal modules $U_1, U_2, \ldots, U_l$ of G_v, which partition the vertex set of G_v. Let the graph G_v^0 arise from G_v by contracting each maximal module U_i of G_v into a single vertex denoted i to which we assign the weight $w(i) = w(I_{v_i})$. Since G_v^0 belongs to $\mathcal{X}_0$, there is an algorithm $\mathcal{A}$ that solves Weighted Independent Set on the instance (G_v^0, W) in time $f(W)l^c \le f(W)n^c$, where c is a constant. If I is the output of $\mathcal{A}$, then let $I_v = \bigcup_{i \in I} I_{v_i}$. It is not difficult to see that the set assigned to the root of T correctly solves Weighted Independent Set on the instance (G, W). Since T has $O(n)$ vertices, the overall time complexity is at most $f(W)n^{c+1}$. $\square$

Theorem 27 reduces the Weighted Independent Set problem from general graphs to prime graphs. The corresponding result for the non-parameterized problem is well-known.

Now we apply Theorem 27 in order to develop an fpt-algorithm for the Weighted Independent Set problem in the class of $\{house, bull\}$-free graphs. The graphs *house* and *bull* are shown in Fig. 3.1. Observe that both these graphs contain K_3+e. Therefore, the class of $\{house, bull\}$-free graphs extends the class of (K_3+e)-free graphs for which an fpt solution was shown in Theorem 25.

(a) house (b) bull

Figure 3.1: The house and the bull graphs

Theorem 28. *The* Weighted Independent Set *problem is fixed-parameter tractable in the class of* $\{house, bull\}$*-free graphs.*

Proof: To prove the theorem, we use the following characterisation of $\{house, bull\}$-free graphs proposed in [Olariu, 1991]: Every prime $\{house, bull\}$-free graph is either triangle-free or the complement of a bipartite chain graph. (A bipartite graph is a bipartite chain graph if the vertices in both parts of the bipartition are linearly ordered by inclusion of neighbourhoods.) Obviously, for the complements of bipartite graphs, the Maximum Weighted Independent Set problem can be solved in polynomial time, since the size of any independent set in such a graph is at most 2. Also, by Theorem 18, the Weighted Independent Set problem is fixed-parameter tractable in the class of triangle-free graphs. Therefore, by Theorem 27, it is fixed-parameter tractable in the class of $\{house, bull\}$-free graphs. $\square$

3.4.1 A Simpler Algorithm

Modular decomposition is a very powerful technique that is often very useful. A modular decomposition for a graph can be found in linear time [McConnell and Spinrad, 1999], however the algorithm for doing this is very complex. If we do not use such complicated methods, it is also possible to prove a slightly weaker result in the unweighted case using a very simple algorithm. In particular, we show that the MAXIMUM INDEPENDENT SET problem is fixed-parameter tractable in the class of to $(C_4, bull)$-free graphs, when parameterized by the solution size. Note that this is a subclass of $(house, bull)$-free graphs. The solution is based on the following technical lemma.

Lemma 29. *In any triangle in a $(C_4, bull)$-free graph G, there are two distinct vertices u, v such that $N_G[u] \subseteq N_G[v]$.*

Proof. Let xyz be a triangle in a $(C_4, bull)$-free graph G. For contradiction, we assume that for every two distinct vertices $u, v \in V(C)$, the set $N_G[u] \setminus N_G[v]$ is not empty.

First, we suppose that there is some $x' \in N_G[x] \setminus (N_G[y] \cup N_G[z])$, i.e. $V(C) \cup \{x'\}$ induces a paw. Let $y' \in N_G[y] \setminus N_G[z]$ and $z' \in N_G[z] \setminus N_G[y]$. Since neither $G[x, y, z, x', y']$ nor $G[x, y, z, x', z']$ can be a bull and neither $G[x, y, y', x']$ nor $G[x, z, z', x']$ can be a C_4, xy' and xz' are edges of G. Since $G[y, z, z', y']$ is not a C_4, the vertices y' and z' are not adjacent. Let $x'' \in N_G[y] \setminus N_G[x]$. Since $G[x, y, x'', x']$ is not a C_4, the vertices x' and x'' are not adjacent. Since $G[x, y, z, x', x'']$ is not a bull, $x''z$ is an edge of G. Since $G[x'', y, z, y', z']$ is not a bull, we may assume, by symmetry, that $x''y'$ is an edge of G. Now $G[x, z, x'', y']$ is a C_4, which is a contradiction.

Hence, we may assume that G contains no induced paw (a graph obtained from a bull by deleting a vertex of degree one). This implies the existence of vertices $x' \in (N_G[y] \cap N_G[z]) \setminus N_G[x]$, $y' \in (N_G[x] \cap N_G[z]) \setminus N_G[y]$, and $z' \in (N_G[x] \cap N_G[y]) \setminus N_G[z]$. If $\{x', y', z'\}$ is independent, then $G[x', z, y, y', z']$ is a bull. Hence, we may assume, by symmetry, that $x'y'$ is an edge of G. Now $G[x, y, x', y']$ is a C_4 which is a contradiction. This completes the proof. $\square$

Theorem 30. *The* MAXIMUM INDEPENDENT SET *problem parameterized by the solution size is fixed-parameter tractable in the class of $(C_4, bull)$-free graphs.*

Proof. If a graph G contains two vertices u, v such that $N_G[u] \subseteq N_G[v]$, then deletion of the vertex v does not change the independence number of G, since any independent set S containing v contains neither u nor any neighbour of u. Therefore, v can be replaced in S by u. This argument and Lemma 29 imply that if a $(C_4, bull)$-free graph G contains a triangle, then one of the vertices of this triangle can be deleted without changing the independence number of G. In other words, in at most n^3 steps the problem can be reduced from G to a triangle-free induced subgraph of G with the same independence number. Since for the triangle-free graphs the problem is fixed-parameter tractable, the result follows. □

3.5 Conclusion

In this chapter, we obtained new results on the parameterized complexity of the WEIGHTED INDEPENDENT SET problem in hereditary classes of graphs. These new results, together with some previously known results, allow us to conclude, in particular, that the problem is fixed-parameter tractable in all hereditary classes defined by a single forbidden induced subgraph G with at most 4 vertices, except for $G = C_4$. Finding the parameterized complexity of the problem in the class of C_4-free graphs is a challenging open problem. In addition to the techniques studied in this chapter, some other approaches may be useful for finding an answer to the above question, such as graph transformations [Lozin, 2011], separating cliques [Brandstädt and Hoàng, 2007], and split decomposition [Rao, 2008].

There has recently been a lot of research on kernel sizes for fpt problems. The kernel sizes given by the algorithms in this chapter are quite large. Finding lower bounds for the kernel size is an interesting direction for future research.

Chapter 4

The Maximum Induced Matching Problem

4.1 Introduction

A *matching* in a graph is a subset of edges no two of which share of vertex. A matching is *induced* if no two vertices belonging to different edges of the matching are adjacent. In other words, an induced matching in a graph G is formed by the edges of a 1-regular induced subgraph of G. Induced matchings have also appeared under the name "strong matchings" [Golumbic and Laskar, 1993]. Faudree et al. [1989] were the first to study induced matchings in the context of bipartite graphs.

The Maximum Induced Matching problem is that of finding an induced matching of maximum cardinality in a graph. We use $i\mu(G)$ to denote the maximum size of an induced matching in G. Another way of describing it is as the Maximum Independent Set problem in $L(G)^2$, the square of the line graph of G.

Due to various applications (see e.g. [Golumbic and Lewenstein, 2000]), the Maximum Induced Matching problem has received much attention in recent years. It was originally introduced in [Stockmeyer and Vazirani, 1982], where it was called the "risk-free marriage problem".

From a computational point of view, finding an induced matching of maximum cardinality in a graph is an intractable problem in many respects. First of all, this problem is NP-hard, which was proved independently in [Cameron, 1989] and [Stockmeyer and Vazirani, 1982]. Moreover, it remains NP-hard under substantial restrictions, for instance: for bipartite graphs of vertex degree at most 3 [Lozin, 2002b; Rusu, 2008], line graphs [Kobler and Rotics, 2003], planar graphs of vertex degree at most 4 [Ko and Shepherd, 2003] and even for cubic planar graphs [Duckworth et al., 2005]. On the other hand, polynomial-time algorithms for this problem have been developed for weakly chordal graphs [Cameron et al., 2003], AT-free graphs [Chang, 2003], circular arc graphs [Golumbic and Laskar, 1993], graphs of bounded clique-width [Kobler and Rotics, 2003] and some other classes of graphs [Brandstädt and Hoàng, 2008; Brandstädt et al., 2007; Cameron, 2004; Golumbic and Lewenstein, 2000; Lozin, 2002b].

The problem also remains intractable in terms of finding approximation algorithms. In particular, in [Duckworth et al., 2005] the problem was shown to be APX-complete in cubic graphs and in bipartite graphs where the minimum degree is $2s$ and the maximum degree is $3s$, for any positive integer s. In [Orlovich et al., 2008] it was shown that the problem is not approximable within a factor of $n^{1/2-\varepsilon}$ for any $\varepsilon > 0$. Some other inapproximability results can be found in [Chlebík and Chlebíková, 2008] and in [Duckworth et al., 2005], where some explicit lower bounds for the approximation ratio are given. Conversely, a subclass of bipartite graphs has been found in which the problem admits a polynomial-time approximation scheme [Duckworth et al., 2003]. For all notions related to approximation theory not defined in this thesis, the reader is referred to [Ausiello et al., 1999].

The problem was also shown to be intractable from a parameterized point of view. More precisely, it is W[1]-hard in general [Moser and Thilikos, 2009] and even when restricted to bipartite graphs [Moser and Sikdar, 2009]. In Section 4.2 we reveal a number of graph classes, including subclasses of bipartite graphs, where the problem admits fixed-parameter tractable algorithms.

Much attention has been given to the problem in regular graphs (see e.g. [Assiyatun, 2005; Chlebík and Chlebíková, 2008; Duckworth et al., 2002, 2005; Gotthilf and Lewenstein, 2006]). We contribute to this topic in two different ways. In Section 4.3 we show that the problem is APX-complete in k-regular bipartite graphs, for any $k \geq 3$, which was previously unknown, despite the fact that finding a maximum induced matching was known to be APX-hard both for regular graphs and for bipartite graphs. In contrast to this negative result we show that the problem admits a simple solution for hypercubes (a proper subclass of regular bipartite graphs).

4.2 Parameterized complexity of the problem

Recall that in parameterized complexity theory, an instance of a graph problem is a pair (G, k), where G is a graph and k is a parameter assigning a natural number to each graph. A parameterized problem is *fixed-parameter tractable* if it can be solved in

$f(k)n^{O(1)}$ time, where n is the number of vertices in G and $f(k)$ is a computable function depending only on the value of the parameter k. An fpt algorithm is one that solves the parameterized problem in $f(k)n^{O(1)}$ time.

We study the following parameterization of the Maximum Induced Matching problem:

k-Induced Matching

Instance:	A graph G and a positive integer k.
Parameter:	k.
Problem:	Decide whether G has an induced matching of size at least k and find such a matching if it exists. If no such matching exists, find an induced matching of size $i\mu(G)$ instead.

The parameterized complexity of the k-Induced Matching problem was studied in [Moser and Thilikos, 2009] and [Moser and Sikdar, 2009]. In particular, in [Moser and Thilikos, 2009] it was shown that the problem is W[1]-hard and hence unlikely to be fixed-parameter tractable when parameterized by the solution size. In [Moser and Sikdar, 2009], this result was strengthened by showing that the problem is also W[1]-hard when restricted to bipartite graphs. On the other hand, in [Moser and Sikdar, 2009] the problem was shown to be fixed-parameter tractable in several classes such as planar graphs (see also [Kang et al., 2010] for a better fpt algorithm for planar graphs with maximum degree 3), graphs of bounded degree and the class of (C_3, C_4, C_5)-free graphs. Observe that the latter class includes, in particular, all C_4-free bipartite graphs, where the problem is known to be NP-hard [Lozin, 2002b]. We generalise the fixed-parameter tractability of the problem in the class of C_4-free bipartite graphs in two different ways. First, in Section 4.2.1 we prove fixed-parameter tractability of the problem in the class of $(K_s, K_{t,t})$-free graphs for arbitrary values of s and t, which also generalises the results for (C_3, C_4, C_5)-free graphs, planar graphs and graphs of bounded degree. Second, in Section 4.2.2 we present an fpt algorithm for so-called A-free bipartite graphs.

Now, let us note that in any graph, if two vertices x and y have the same neighbourhood, at most one of them can be the endpoint of an edge in any induced matching. Consequently, for computing a maximum induced matching we can first look for every pair (x, y) of vertices with the same neighbourhood and arbitrarily delete one of them. This can be done in polynomial time.

Remark 1. *If two vertices in a graph have the same neighbourhood, we can arbitrarily delete one of them and the size of the maximum induced matching will be unchanged.*

Modules in bipartite graphs have the following property (recall that modules were defined in Chapter 1).

Remark 2. *In a connected bipartite graph G, every non-trivial module must be an independent set.*

Indeed, if a non-trivial module X of vertices in G contains two vertices y, y' that are adjacent, then they must be in different parts of the bipartition of G. Since X is a non-trivial module and the graph G is connected, there must be a vertex z outside of X with a neighbour $x \in X$. This vertex z can be connected to at most one of y and y', since the graph is bipartite, contradicting the claim that X is a module.

From this remark, it follows that in any non-trivial module X of a connected bipartite graph G, every vertex of X must have the same neighbourhood. This means that if G is a bipartite graph and no two vertices of G have the same neighbourhood, then every component of G is prime.

Since the problem can be solved independently on each component of a graph, we can therefore draw the following conclusion:

Remark 3. *The k-*INDUCED MATCHING *problem is fixed-parameter tractable in a hereditary class of bipartite graphs $\mathcal{C}$ if and only if it is fixed-parameter tractable in the class of prime graphs in $\mathcal{C}$.*

4.2.1 An fpt algorithm for $(K_s, K_{t,t})$-free graphs

We denote an induced matching with p edges by M_p. Also, we let $R(s,t)$ be the non-symmetric Ramsey number. That is, we define $R(s,t)$ to be the minimum number such that if G is a graph on at least $R(s,t)$ vertices, then either G contains K_s as an induced subgraph or the complement of G contains K_t as an induced subgraph (i.e. G contains an independent set of size t). The number $N(t,p)$ is defined as in Lemma 5 i.e. it is a number such that every bipartite graph with a matching of size at least $N(t,p)$ contains either a bi-clique $K_{t,t}$ or an induced matching M_p (without loss of generality, we may assume $N(t,p)$ is the minimal number with this property).

We start by generalising Lemma 5 to non-bipartite classes of graphs.

Lemma 31. *For any natural numbers s, t and p, there is a number $N'(s,t,p)$ such that every graph with a matching of size at least $N'(s,t,p)$ contains either a clique K_s, an induced bi-clique $K_{t,t}$ or an induced matching M_p.*

Proof. We will show that setting $N'(s,t,p) = R(s, R(s, N(t,p)))$ is sufficient. Indeed, suppose G is a $(K_s, K_{t,t})$-free graph with a matching of size $R(s, R(s, N(t,p)))$. G is

K_s-free, so it must contain an independent set A of size $R(s, N(t,p))$. Let B be the set of vertices matched to A. Since $G[B]$ is K_s-free, it must contain an independent set B' of size $N(t,p)$. Let A' be the set of vertices matched to B'. Now $G[A' \cup B']$ is a bipartite graph with a matching of size $N(t,p)$. By Lemma 5, $G[A' \cup B']$ contains an induced matching M_p. $\square$

Theorem 32. *For each fixed s and t, the k-*Induced Matching *problem is fixed-parameter tractable in the class of $(K_s, K_{t,t})$-free graphs.*

Proof. Fix s and t and let G be a $(K_s, K_{t,t})$-free graph with n vertices. We will show that the problem of determining whether G has an induced matching of size k can be solved in time $f(k)p(n)$, where $f(k)$ is a function of k only and $p(n)$ is a polynomial in n independent of k.

Let M be a maximal (with respect to set inclusion) matching in G. Clearly, such a matching can be found in polynomial time. If M is of size at least $N'(s,t,k)$, then by Lemma 31, G has an induced matching of size k. To find such a matching, we can restrict ourselves to $N'(s,t,k)$ edges of M. This reduces the problem to a subgraph G induced by $2N'(s,t,k)$ vertices, at which point the problem can be solved in $O(N'(s,t,k)^{2k}k^2)$ time, i.e. independent of n.

If M contains less than $N'(s,t,k)$ edges we proceed as follows. Let V_M be the set of vertices which are endpoints of edges in M. If $xy \in E(G)$, then either $x \in V_M$ or $y \in V_M$ (otherwise M would not be maximal). By Remark 1, we may assume that every vertex of G has a different neighbourhood.

So we are now reduced to a graph in which the neighbourhood of every vertex $v \in V \backslash V_M$ is contained in V_M and no two vertices have the same neighbourhood. Thus the graph contains at most $2^{2N'(s,t,k)} + 2N'(s,t,k)$ vertices and we can solve the k-Induced Matching problem in $O((2^{2N'(s,t,k)} + 2N'(s,t,k))^{2k}k^2)$ time, which is independent of n.

Summarising, we conclude that the problem is fixed-parameter tractable in the class of $(K_s, K_{t,t})$-free graphs. $\square$

This proof can also be generalised to the weighted version of the problem, where the $w(xy)$ weight of any edge xy is at least 1:

k-WEIGHTED INDUCED MATCHING

Instance:	A graph G, a weight function $w : E(G) \to \mathbb{R}$ and a positive integer k.
Parameter:	k.
Problem:	Decide whether G has an induced matching of total weight at least k and find such a matching if it exists. If no such matching exists, find an induced matching of maximum weight instead.

Theorem 33. *For each fixed s and t, the k-WEIGHTED INDUCED MATCHING problem is fixed-parameter tractable in the class of $(K_s, K_{t,t})$-free graphs.*

Proof. Note that since we insist that every edge has weight at least 1, the total weight of any induced matching must be greater than or equal to its size. The proof follows similarly, except that we cannot immediately assume that the input graph G is prime. If G is not prime, say x and y have the same neighbourhood in G. In this case we delete both x and y and replace them with a new vertex z with the same neighbourhood as x and y. For vertices a in the neighbourhood of z we define the weight function $w(za) = \max\{w(xa), w(ya)\}$. We keep doing this until the graph is prime. If the algorithm selects an induced matching containing the edge za for some a in the resulting prime graph, we simply take that to mean that the algorithm selects the edge (xa or ya) in the original graph with the corresponding weight. $\square$

Corollary 34. *For each fixed t, the k-WEIGHTED INDUCED MATCHING (and therefore also the k-INDUCED MATCHING) problem is fixed-parameter tractable in the class of $(K_{t,t})$-free bipartite graphs.*

4.2.2 An fpt algorithm for A-free bipartite graphs

In this section, we develop an fpt algorithm for the k-INDUCED MATCHING problem in the class of A-free bipartite graphs, where A is the graph represented in Figure 4.1. Since A contains a C_4, the class of A-free bipartite graphs extends C_4-free bipartite graphs.

Clearly, the problem can be reduced to connected graphs. More importantly, by Remark 3, the problem can be reduced to bipartite graphs which are prime.

Let G be a connected prime A-free bipartite graph. If G contains no C_4, we apply the fpt algorithm from the previous section (since $C_4 = K_{2,2}$). If G contains a C_4, we apply the following structural characterisation of G.

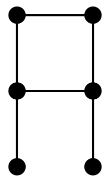


Figure 4.1: The graph A

Lemma 35. *Let $G = (U, V, E)$ be a connected prime A-free bipartite graph, containing a C_4. Then the vertices of G can be partitioned into four subsets U_0, U_1, V_0, V_1 in such way that $U_0 \cup V_0$ and $U_1 \cup V_1$ induce complete bipartite graphs, while $U_0 \cup V_1$ and $U_1 \cup V_0$ induce P_6-free bipartite graphs.*

Proof. Consider a connected prime A-free bipartite graph $G = (U, V, E)$ containing a C_4 and let $H = G[U_0 \cup V_0]$ be a maximal (with respect to inclusion) complete bipartite subgraph containing this C_4. Also, for $i \geq 1$, let U_i and V_i be the set of vertices in U and V at distance i from $U_0 \cup V_0$. Then $U_1 \cup V_1$ induces a complete bipartite graph. Indeed, assume for contradiction that a vertex $a \in U_1$ is not adjacent to a vertex $x \in V_1$. By definition, a must have a neighbour $b \in V_0$ and a non-neighbour $c \in V_0$ (since otherwise H would not be a maximal complete bipartite subgraph containing the initial C_4). Similarly, x must have a neighbour $y \in U_0$ and a non-neighbour $z \in U_0$. But then a, b, c, x, y, z induce an A.

Notice that each of U_0 and V_0 contains at least 2 vertices, which together with the primality of G implies that U_1 is not empty and V_1 is not empty, since otherwise any two vertices of U_0 or V_0 would have the same neighbourhood. As a result, we can conclude that for all $i > 1$ the sets U_i and V_i are empty. Indeed, assume for contradiction that U_2 contains a vertex a. Then by definition it must have a neighbour $x \in V_1$, while x must have a neighbour c in U_0. Since U_1 is not empty, we may consider an arbitrary vertex $b \in U_1$, an arbitrary neighbour $y \in V_0$ of b and an arbitrary non-neighbour (which exists due to maximality of H) $z \in V_0$ of b. But then a, b, c, x, y, z induce an A. This contradiction shows that U_2 is empty, and by symmetry we conclude that V_2 is empty.

Assume now that $G[U_1 \cup V_0]$ contains an induced $P_6 = (x_1, x_2, x_3, x_4, x_5, x_6)$ with $x_1, x_3, x_5 \in U_1$ and $x_2, x_4, x_6 \in V_0$, and let a be an arbitrary vertex in U_0. Then $a, x_1, x_2, x_3, x_4, x_6$ induce an A in G. This contradiction proves that $G[U_1 \cup V_0]$ is P_6-free, and by symmetry we conclude that $G[U_0 \cup V_1]$ is P_6-free. $\square$

Corollary 36. *The clique-width of A-free connected prime bipartite graphs containing a C_4 is bounded by a constant.*

Proof. It is known (see e.g. [Fouquet et al., 1999]) that the clique-width of bipartite graphs in a given hereditary class is bounded if and only if it is bounded for graphs in this class which are connected and whose bipartite complement is also connected.

Let $\mathcal{C}$ denote the class of connected prime A-free bipartite graphs containing a C_4. Let H be an induced subgraph of a graph in $\mathcal{C}$. By Lemma 35, H either has disconnected bipartite complement or it is a P_6-free bipartite graph. Indeed, suppose H is an induced subgraph of some graph $G \in \mathcal{C}$ and let U_0, U_1, V_0, V_1 be defined for the graph G as in Lemma 35. If both $V(H) \cap (U_0 \cup V_1)$ and $V(H) \cap (U_1 \cup V_0)$ are non-empty then these two sets of vertices must be in different components of the bipartite complement of H. Otherwise, one of these sets must be empty, in which case the graph H is P_6-free. However, P_6-free bipartite graphs are known to have bounded clique-width (see e.g. [Giakoumakis and Vanherpe, 2003]), which completes the proof. □

For graphs of bounded clique-width, the Maximum Induced Matching problem is known to be solvable in polynomial-time [Kobler and Rotics, 2003]. This fact together with Corollary 36 and an fpt algorithm for the k-Induced Matching problem in the class of C_4-free graphs, leads to the following conclusion.

Theorem 37. *The k-Induced Matching problem is fixed-parameter tractable in the class of A-free bipartite graphs.*

4.3 Regular bipartite graphs

As we mentioned in the introduction, much attention has been given to the problem in regular graphs. In particular, [Duckworth et al., 2002] and [Assiyatun, 2005] study the problem in random regular graphs, while [Chlebík and Chlebíková, 2008; Duckworth et al., 2005; Gotthilf and Lewenstein, 2006] study approximability of the problem in regular graphs. This interest in regular graphs is partly due to the fact that the problem remains NP-hard in this class. It is also known that the problem is NP-hard for bipartite graphs of bounded degree. However, the complexity of the problem in regular bipartite graphs was unknown. We answer this question negatively by showing that the problem is APX-complete in k-regular bipartite graphs for any $k \geq 3$, implying that there is a constant $c > 1$ such that it is NP-hard to approximate the problem to within a factor of c. An interesting subclass of regular bipartite graphs is the class of hypercubes. We show that in this case the problem admits a simple solution.

4.3.1 APX-completeness

In this section we show that the Maximum Induced Matching problem is APX-complete in the class of r-regular bipartite graphs for any $r \geq 3$. The proof follows from an approximation preserving reduction. We use the L-reduction as defined in [Papadimitriou and Yannakakis, 1991]. Let P be a maximisation problem. For every instance x of P, and every solution y of x, let $c_P(x, y)$ be the *cost* of the solution y. Let $opt_P(x)$ be the cost of an optimal solution. If $c \geq 1$ is a constant and there is a polynomial time algorithm that computes a solution $y(x)$, such that $\forall x, c_P(x, y(x)) \geq \frac{1}{c} opt_P(x)$, then the algorithm is said to approximate P to within a *ratio* of c. If this holds for a constant $c > 1$, then P is said to be constant-factor approximable and it belongs to the class APX. If, for any positive ε, P has a polynomial time algorithm which approximates P to within a ratio of $\leq 1 + \varepsilon$, then we say that P has a polynomial-time approximation scheme (PTAS).

Definition 38. *Let P and Q be two maximisation problems. An* L-reduction *from P to Q is a four-tuple $(t_1, t_2, \alpha, \beta)$, where t_1 and t_2 are polynomial time computable functions and α and β are positive constants with the following properties:*

(a) t_1 maps instances of P to instances of Q and for every instance x of P, $opt_Q(t_1(x)) \leq \alpha opt_P(x)$.

(b) For every instance x of P, t_2 maps pairs $(t_1(x), y')$ (where y' is a solution of $t_1(x)$) to a solution y of x so that $|opt_P(x) - c_P(x, t_2(t_1(x), y'))| \leq \beta |opt_Q(t_1(x)) - c_Q(t_1(x), y')|$.

As shown in [Papadimitriou and Yannakakis, 1991], if P and Q are maximisation problems and there is an L-reduction from P to Q then if Q has a PTAS, P must also have a PTAS. Conversely, the definition of APX-hardness implies that if P is APX-complete, then Q is APX-hard. If furthermore Q is in APX, then it is APX-complete.

For any finite set D of positive integers, we say a graph G is a *D-graph* if D is the set of vertex degrees in G. For example a $\{k\}$-graph is a non-empty k-regular graph.

Theorem 39. *Let D be a finite set of positive integers such that $\max_{d \in D} d \geq 3$, then* Maximum Induced Matching *is APX-complete in the class of bipartite D-graphs. In particular, it is APX-complete in the class of k-regular bipartite graphs for any $k \geq 3$.*

Proof. The Maximum Induced Matching problem is known to be approximable to within a constant factor in k-regular graphs [Zito, 1999]; so it remains to show it is APX-hard.

For any fixed $k \geq 3$, we define the gadget $H_k = (V_k, E_k)$ (see Figure 4.2) as follows:

The set of vertices is defined by $V_k = L_1 \cup L_2 \cup L_3 \cup L_4 \cup L_5 \cup L_6$ with

$$\begin{array}{lll} L_1 = \{1_1, \ldots, 1_k\}, & L_2 = \{2_1, \ldots, 2_k\}, & L_3 = \{3_1, \ldots, 3_{k(k-1)}\}, \\ L_4 = \{4_1, \ldots, 4_{k(k-1)}\}, & L_5 = \{5_1, \ldots, 5_{(k-1)^2}\}, & L_6 = \{6_1, \ldots, 6_{(k-1)(k-2)}\}. \end{array}$$

For $i = 1, \ldots, k-1$, we denote

$$\begin{array}{ll} S_3^i = \{3_{(i-1)k+1}, \ldots, 3_{ik}\}, & S_4^i = \{4_{(i-1)k+1}, \ldots, 4_{ik}\}, \\ S_5^i = \{5_{(i-1)(k-1)+1}, \ldots, 5_{i(k-1)}\}, & S_6^i = \{6_{(i-1)(k-2)+1}, \ldots, 6_{i(k-2)}\}. \end{array}$$

So $|S_3^i| = |S_4^i| = k$, $|S_5^i| = k-1$ and $|S_6^i| = k-2$.

The set of edges E_k is defined as follows:

(1) $L_1 \cup L_2$ induces a matching of size k: $(1_i, 2_i) \in E_k, i = 1, \ldots, k$.

(2) $(2_i, 3_{(i-1)(k-1)+j}) \in E_k, i = 1, \ldots, k, j = 1, \ldots, k-1$.

(3) $L_3 \cup L_4$ induces a matching: $(3_i, 4_i) \in E_k, i = 1, \ldots, k(k-1)$.

(4) For every $i = 1, \ldots, k-1$, S_4^i and S_5^i induce a $K_{k,k-1}$.

(5) For every $i = 1, \ldots, k-1$, S_3^i and S_6^i induce a $K_{k,k-2}$.

Note that every vertex of L_1 is of degree 1 in H_k while the other vertices are of degree k. Note also that $\forall i \in \{1, \ldots, k-1\}, N(S_3^i) \cap L_2 = \{2_i, 2_{i+1}\}$.

For any graph $G = (V, E)$ and any set of k vertices $S = \{v_1, \ldots, v_k\} \subset V$, we define the graph $G \cup_S H_k$ obtained by adding an H_k to G and identifying L_1 and S. More formally its set of vertices is $V \cup L_2 \cup L_3 \cup L_4 \cup L_5 \cup L_6$ and $(G \cup_S H_k)[V] = G$ and $(G \cup_S H_k)[S \cup L_2 \cup L_3 \cup L_4 \cup L_5 \cup L_6] = H_k$. For any two graphs $G = (V, E), G' = (V', E')$ we denote $G \cup G' = (V \cup V', E \cup E')$.

Lemma 40. *For any $k \geq 3$, $\{(3_i, 4_i), i = 1, \ldots, k(k-1)\}$ is a maximum induced matching of H_k.*

Proof. Note first that, since vertices 1_i, $i = 1, \ldots, k$ are of degree 1 in H_k, for any induced matching M of H_k containing an edge $(2_i, 3_{(i-1)(k-1)+j})$, with $i \in \{1, \ldots, k\}$ and $j \in \{1, \ldots, k-1\}$, $M \setminus \{(2_i, 3_{(i-1)(k-1)+j})\} \cup \{(1_i, 2_i)\}$ is also an induced matching. Consequently, without loss of generality we can restrict ourselves to the case where M does not contain any edge $(u, v), u \in L_2, v \in L_3$. For every $i = 1, \ldots, k-1$, we let $M_i = M \cap [\{(1_i, 2_i), (1_{i+1}, 2_{i+1})\} \cup \{(u, v), u \in S_3^i, v \in S_6^i\} \cup \{(u, v), u \in S_4^i, v \in S_5^i\}]$. Note that $|M_i| \leq 3$. Since edges $(u, v), u \in S_3^i, v \in S_4^i$ constitute an induced matching and

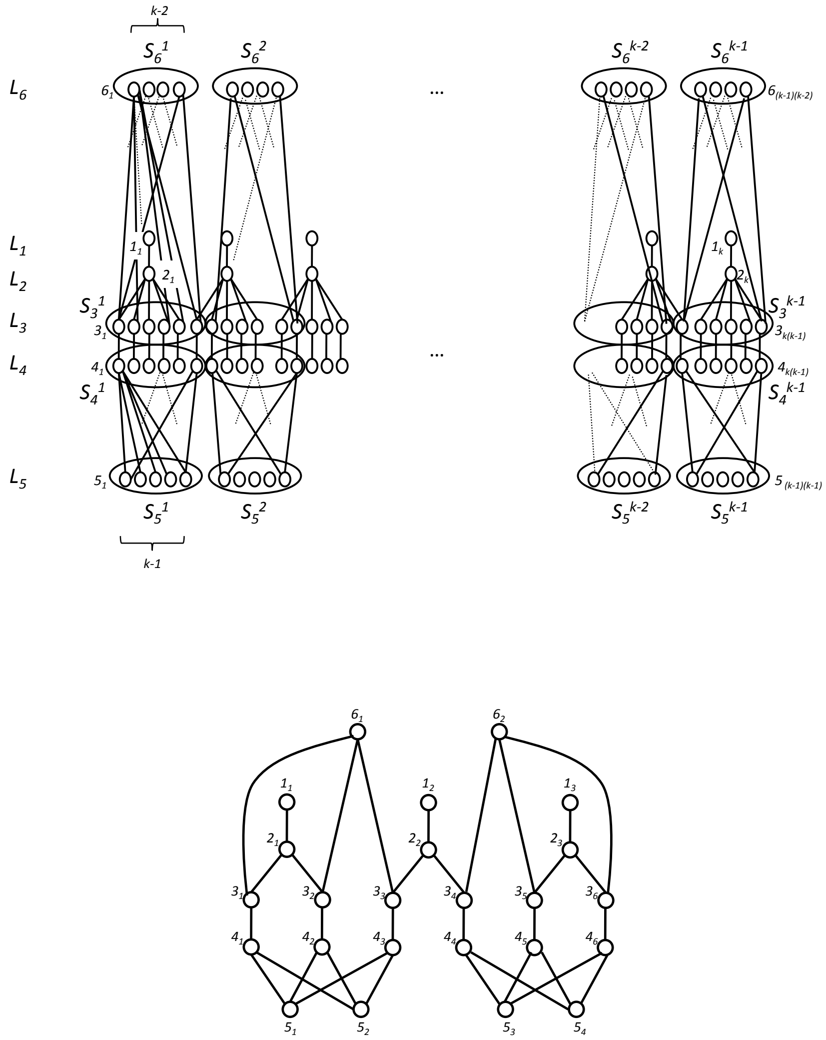


Figure 4.2: Gadget H_k and example for $k = 3$.

are only linked, in M, to edges belonging to M_i, $M' = M \setminus M_i \cup \{(u,v), u \in S_3^i, v \in S_4^i\}$ is also an induced matching. Moreover, if M contains an edge $(u,v), u \in S_3^i, v \in S_4^i$, then it does not contain any edge $(u',v'), u' \in S_3^i, v' \in S_6^i$ or any edge $(u'',v''), u'' \in S_4^i, v'' \in S_5^i$. So $|M'| \geq |M|$ and consequently, there is a maximum induced matching of H_k containing edges $(u,v), u \in L_3, v \in L_4$. Since this matching is maximal, Lemma 40 follows. □

A direct consequence of Lemma 40 is that there is a maximum induced matching in $G \cup_S H_k$ containing $\{(3_i, 4_i), i = 1, \ldots, k(k-1)\}$ and consequently

$$i\mu(G \cup_S H_k) = i\mu(G) + k(k-1) \tag{4.1}$$

Moreover, if G is bipartite and S is monochromatic for a fixed 2-colouring of G, then $G \cup_S H_k$ is also bipartite.

We can now describe the reduction. Let $G = (V, E)$ be any bipartite D-graph. Let us first note that for any positive integer d, $G \cup K_{d,d}$ is a bipartite $(D \cup \{d\})$-graph and

$$i\mu(G \cup K_{d,d}) = i\mu(G) + 1 \tag{4.2}$$

On the other hand, for any $d \in D$, let $u_d^1, \ldots, u_d^p$, $p \geq 1$ denote the vertices of degree d. Let $k \geq 3$. We consider k copies of G denoted by $G_1, \ldots, G_k$ and for any vertex $v \in V(G)$ we let $S(v)$ denote the set of copies of v in $G_1, \ldots, G_k$ (so $|S(v)| = k$). We then define:

$$T_d^k(G) = (G_1 \cup \ldots \cup G_k) \cup_{S(u_d^1)} H_k \ldots \cup_{S(u_d^p)} H_k$$

Using relation (4.1) we immediately obtain:

$$i\mu(T_d^k(G)) = ki\mu(G) + pk(k-1) \tag{4.3}$$

It is also straightforward to verify that, if $k \in D \cup \{d+1\}, d \neq k$, then $T_d^k(G)$ is a bipartite $(D \setminus \{d\} \cup \{d+1\})$-graph. Since $G \cup K_{d,d}$ and $T_d^k(G)$ can be performed in polynomial time, relations (4.2) and (4.3) imply that the related reduction preserves polynomial approximation schema.

The first of these is an L-reduction with $\alpha = 2, \beta = 1$, where t_1 maps G to $G \cup K_{d,d}$ and t_2 maps $(G \cup K_{d,d}, M)$ to (G, M'), where M' is the set of edges in M that do not occur in the $K_{d,d}$. Since G is a D-graph, we know that $i\mu(G) \geq 1$, so the first inequality in the definition of L-reduction holds. The second inequality follows from the fact that at most one edge in the $K_{d,d}$ can occur in M.

The second reduction is also an L-reduction with $\alpha = k(1+2(2\Delta(\Delta-1)+1)(k-1))$ and $\beta = 1/k$, where $\Delta = \max(D)$. We define to t_1 to map G to $T_d^k(G)$ and define t_2 to map $(T_d^k(G), M)$ to (G, M'), where M' is the set of edges in M that belong to one of the copies of G in $T_d^k(G)$, where this copy of G is chosen so that the size of M' is maximised.

Indeed, observe that since the degree in G is bounded above by Δ, any edge in G is linked to at most $2\Delta(\Delta-1)$ edges. Since D is made up of positive integers, the minimum degree in G is at least 1, so there are at least $\frac{n}{2}$ edges in G. Thus $i\mu(G) \geq \frac{n}{2(2\Delta(\Delta-1)+1)}$. Note that $p \leq n$ for this transformation. This means that

$$\begin{aligned} i\mu(T_d^k(G)) &= ki\mu(G) + pk(k-1) \\ &\leq ki\mu(G) + nk(k-1) \\ &\leq k(1 + 2(2\Delta(\Delta-1)+1)(k-1))i\mu(G) \end{aligned}$$

as required by the first inequality in the definition of L-reduction. For the second inequality, consider $(G, M') = t_2(T_d^k(G), M)$. Let M'' consist of the edges in $T_d^k(G)$ of the form $(3_i, 4_i)$ along those edges in in every copy of G in $T_d^k(G)$ that correspond to the edges in M'. Then $|M''| = k|M| + pk(k-1)$. Using Lemma 40 as before, we conclude that $|M| \leq |M''|$, so

$$\begin{aligned} i\mu(G) - |M'| &= \frac{1}{k}(ki\mu(G) + pk(k-1) - |M''| \\ &\leq \frac{1}{k}(i\mu(T_d^k(G)) - |M|) \end{aligned}$$

as required.

Consequently if the Maximum Induced Matching problem is APX-complete in bipartite D-graphs, then for any positive integer d it is also APX-complete in bipartite $(D \cup \{d\})$-graphs and, using the transformation T_d^{d+1} for any $d \in D, d \geq 2$, it is also APX-complete for bipartite $(D \setminus \{d\} \cup \{d+1\})$-graphs.

The problem is shown to be APX-complete for bipartite $\{2,3\}$-graphs [Duckworth et al., 2005]. (More precisely, for any $\varepsilon > 0$, the problem of approximating Maximum Induced Matching within a factor of $\frac{9570}{9569} - \varepsilon$ is NP-hard for graphs in this class.) Then, using the above remarks successively for $d = 2, 3, \ldots$ we deduce that it is APX-complete in bipartite $\{3\}$-, $\{4\}$-, $\ldots$, $\{k\}$-graphs for any $k \geq 3$ and consequently, that it is APX-complete in bipartite D-graphs for any finite D with at least one element $k \geq 3$. $\square$

4.3.2 Hypercubes

The n-dimensional hypercube Q_n is the graph with vertex set $\{0,1\}^n$, where two vertices are adjacent if and only if they differ in precisely 1 coordinate. Thus, the number of vertices in Q_n is 2^n and every vertex has degree n, i.e. Q_n is a regular graph. It is not difficult to see that Q_n is also a bipartite graph, since vertices of the same parity are necessarily non-adjacent. Hypercubes enjoy many more nice graph-theoretic properties (see e.g. [Harary et al., 1988]). Nonetheless, there are algorithmic problems on hypercubes for which no efficient algorithms are known. This is the case, for instance, for the crossing number and the size of a smallest maximal matching for which only bounds are available (see e.g. [Forcade, 1973; Kainen, 1972]) and no efficient algorithms to compute the respective numbers are known. However, this is not the case for the size of a maximum induced matching. Below we present a simple formula for this number. The proof is constructive and exhibits an easy way of finding an induced matching of this size.

Theorem 41. *For $n \geq 2$, a maximum induced matching in the hypercube Q_n has 2^{n-2} edges.*

Proof. Consider the set of vertices $M = \{(x_1, \ldots, x_n),\, x_2 \equiv x_3 + \ldots + x_n \bmod(2)\}$. M contains 2^{n-1} points. Note that $x, y \in M$ are neighbours in Q_n if and only if they differ in the x_1 coordinate, but do not differ in any of the other coordinates. Therefore $Q_n[M]$ is an induced matching in Q_n which contains 2^{n-2} edges.

Clearly this is optimal if $n = 2$. For $n > 2$, fix $x_1, \ldots, x_{n-2} \in \{0,1\}$ and consider the points

$$(x_1, \ldots, x_{n-2}, 0, 0), (x_1, \ldots, x_{n-2}, 0, 1), (x_1, \ldots, x_{n-2}, 1, 0), (x_1, \ldots, x_{n-2}, 1, 1).$$

At most 2 of these points can be endpoints of edges in any induced matching in Q_n. Therefore in any induced matching in Q_n, at most half of the vertices of Q_n can be endpoints of edges in the matching. So the number of edges in such a matching is at most 2^{n-2}. □

4.4 Conclusion

In Theorem 39, we showed that the Maximum Induced Matching problem is APX-complete for k-regular bipartite graphs. A few results with a constant multiplicative error have been established for regular graphs [Duckworth et al., 2005; Gotthilf and

Lewenstein, 2006; Zito, 1999], but no such results exist specifically for the bipartite case. An interesting direction for future research is to investigate the approximation of the Maximum Induced Matching problem in regular bipartite graphs and in particular to see whether approximation results in regular graphs can be widely improved in the bipartite case.

In [Duckworth et al., 2003], the problem was found to have a polynomial-time approximation scheme in the class of bipartite graphs with the property that all the vertices in one part of the partition have degree 2 and all the vertices in the other part have degree 3. Since the problem is APX-complete in the class of bipartite {2,3}-graphs, interesting approximation results may emerge from the study of bipartite instances where all vertices in one part have degree 2 and all vertices in the other part have some other degree d.

Since the problem, is fixed-parameter tractable in the class of $(K_s, K_{t,t})$-free graphs a natural question that arises is whether one can find a small problem kernel in these classes (see [Downey and Fellows, 1999] for the technical definition of kernel). Algorithms to obtain such kernels have been found in some subclasses of $(K_s, K_{t,t})$-free graphs (see e.g. [Moser and Sikdar, 2009]).

Part II

Graph Partitions

Chapter 5

Stable-Π Partitions

5.1 Introduction

Let Π be a graph property (or graph class), i.e. a set of graphs closed under isomorphism. A property Π is *hereditary* if it is closed under taking induced subgraphs, and it is *additive* if it is closed under taking disjoint unions of graphs.

For a property Π, the STABLE-Π problem asks, given a graph G, to determine whether G has an independent set S such that $G - S \in \Pi$. The family of STABLE-Π problems has been extensively studied in the literature (see e.g. [Brandstädt et al., 1998; Cai and Corneil, 1996; Demange et al., 2005; Ekim and Gimbel, 2009; Garey et al., 1976; Hell et al., 2004; Hoàng and Le, 2000; Huang and Chu, 2007; Stacho, 2008]) and includes many important representatives such as VERTEX 3-COLOURABILITY, in which case Π is the set of all bipartite graphs, and EFFICIENT EDGE DOMINATION (also known as DOMINATING INDUCED MATCHING), in which case Π is the set of 1-regular graphs. Both of these examples represent algorithmically hard, i.e. NP-complete, problems. The STABLE-Π problem is also NP-complete for various other properties Π such as forests or trivially perfect graphs [Brandstädt et al., 1998]. More generally, the problem remains NP-complete for any additive hereditary property Π other than the set of edgeless graphs [Kratochvil and Schiermeyer, 1997].

On the other hand, for some properties Π, the STABLE-Π problem can be solved in polynomial time. This is the case, for instance, if Π is the class of co-bipartite graphs [Brandstädt et al., 1998] or the class of complete bipartite graphs [Brandstädt et al., 2005]. The case of co-bipartite graphs was generalised independently in [Alekseev et al., 2004] and [Feder et al., 2003] to arbitrary hereditary properties Π which are of bounded independence number and which can be recognised in polynomial time. The case where

Π is the class of complete bipartite graphs has also received a wide generalisation. To describe this generalisation, let us observe that the class of complete bipartite graphs is quite small. In the terminology of [Balogh et al., 2000], it is *subfactorial*, i.e. for any constant $c > 0$, Π has less than n^{cn} labelled graphs on n vertices, if n is sufficiently large. Subfactorial graph properties have a simple structural characterisation (see Theorem 42). This was used in [Lozin, 2005] to prove that the STABLE-Π problem is polynomial-time solvable for any subfactorial hereditary property Π of bipartite graphs.

In the present chapter, we further generalise this result to arbitrary subfactorial hereditary properties Π (not necessarily of bipartite graphs). Then we switch to hereditary properties with the *factorial* speed of growth, i.e. those containing at least $n^{c_1 n}$ and at most $n^{c_2 n}$ labelled graphs on n vertices for some constants $c_1, c_2 > 0$, when n is sufficiently large. The family of factorial graph properties is much wider and contains many classes of theoretical or practical importance. For instance the classes of threshold graphs, line graphs, permutation graphs, interval graphs are factorial and all classes of graphs of bounded vertex degree, of bounded clique-width and all proper minor closed graph classes have at most factorial speed of growth.

The family of factorial hereditary classes is very rich and varied, but there are only a few such classes for which the complexity of the STABLE-Π problem is known. It is therefore natural to focus on the simplest classes in this family, namely those that are *minimal* (when ordered by set inclusion). There are exactly nine such classes [Alekseev, 1997; Balogh et al., 2000]. Three of them are subclasses of bipartite graphs:

$\mathcal{M}_1$ **bipartite matching graphs:** *graphs partitionable into two independent sets, where the edges between them form a matching (equivalently, graphs of maximum degree one)*

$\mathcal{M}_2$ **bipartite almost complete graphs:** *graphs partitionable into two independent sets such that each vertex has at most one non-neighbour in the opposite part*

$\mathcal{M}_3$ **chain graphs:** *bipartite $2K_2$-free graphs*

Three other minimal factorial classes are subclasses of co-bipartite graphs: these are precisely the classes of complements of graphs in $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$, which we denote by $\overline{\mathcal{M}_1}$, $\overline{\mathcal{M}_2}$, and $\overline{\mathcal{M}_3}$, respectively. The remaining three minimal factorial classes are subclasses of split graphs. They are also closely related to $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$ and can be obtained from graphs in these classes by converting one of the independent sets in the bipartition into a clique. We denote these classes as follows:

$\mathcal{M}_4$ **split matching graphs:** *graphs partitionable into a clique and an independent set, where the edges between them form a matching*

$\overline{\mathcal{M}_4}$ **complements of split matching graphs:** *graphs partitionable into a clique and an independent set so that each vertex has at most one non-neighbour in the opposite part*

$\mathcal{M}_5$ **threshold graphs:** *split P_4-free graphs*

It is known that Stable-$\mathcal{M}_1$ is an NP-complete problem [Mahadev and Peled, 1995], while Stable-$\mathcal{M}_5$ is solvable in polynomial time [Brandstädt et al., 1998]. For the remaining seven minimal factorial classes, the complexity of the problem was unknown and we study it in the present chapter.

The borderline between factorial and subfactorial properties was also studied in [Yannakakis, 1981] for the following problem associated with a hereditary class Π of bipartite graphs: given a bipartite graph G, find the largest induced subgraph of G that belongs to Π. Yannakakis [1981] showed that this problem is solvable in polynomial time if Π is a subfactorial hereditary class, and is NP-hard otherwise (except for the case when Π coincides with the class of all bipartite graphs, in which case the problem is trivial). Inspired by this result, Lozin conjectured [Lozin, 2005] that the Stable-Π problem is NP-complete for all hereditary factorial classes of bipartite graphs, including the three minimal hereditary factorial classes. Contrary to this conjecture, we show that Stable-Π is solvable in polynomial time for nearly all minimal hereditary factorial classes Π (not necessarily bipartite).

Let us emphasise that these nine minimal classes of graphs are hereditary and most of the instances of the Stable-Π problem that have been studied in the literature deal with hereditary properties Π. On the other hand, some important examples of the problem appear in the context of non-hereditary properties Π. We already mentioned Efficient Edge Domination, which is equivalent to Stable-Π when Π is the set of 1-regular graphs. We denote the class of 1-regular graphs by $\mathcal{M}_1^S$. Observe that this set is a restriction of the class $\mathcal{M}_1$. More precisely, $\mathcal{M}_1$ is the hereditary closure of the set of 1-regular graphs (i.e. it is the set containing all 1-regular graphs and all their induced subgraphs). In the same spirit, we define $\mathcal{M}_2^S$ to be the class of graphs partitionable into two independent sets such that each vertex has exactly one non-neighbour in the opposite part and define $\mathcal{M}_4^S$ to be the class of graphs partitionable into a clique and an independent set such that every vertex in one part has exactly one neighbour in the opposite part. As before, we write $\overline{\mathcal{M}_1^S}$, $\overline{\mathcal{M}_2^S}$ and $\overline{\mathcal{M}_4^S}$ to denote the classes of graphs whose complements are in $\mathcal{M}_1^S$, $\mathcal{M}_2^S$ and $\mathcal{M}_4^S$, respectively.

We find that for some minimal factorial classes Π for which Stable-Π can be solved in polynomial time, the restriction to Π^S leads to an NP-complete problem. A summary of our results is given in Table 5.1.

Π	Stable-Π		Stable-Π^S	
$\mathcal{M}_1$	NP-C	[Mahadev and Peled, 1995]	NP-C	[Grinstead et al., 1993]
$\overline{\mathcal{M}_1}$	P	Thm 47	P	Thm 56
$\mathcal{M}_2$	P	Thm 55	NP-C	Thm 60
$\overline{\mathcal{M}_2}$	P	Thm 47	P	Thm 56
$\mathcal{M}_3$	open		n/a	
$\overline{\mathcal{M}_3}$	P	Thm 47	n/a	
$\mathcal{M}_4$	P	Thm 48	NP-C	Thm 58
$\overline{\mathcal{M}_4}$	P	Thm 49	NP-C	Thm 59
$\mathcal{M}_5 = \overline{\mathcal{M}_5}$	P	[Brandstädt et al., 1998]	n/a	

Table 5.1: Summary of complexity results for Stable-Π.

5.2 Preliminaries

A *graph property*, or *graph class* is any set Π of simple graphs closed under isomorphism. The *graph-complement* $\overline{\Pi}$ of a property Π is defined as $\overline{\Pi} = \{\overline{G} \mid G \in \Pi\}$. A graph property is *hereditary* if it is closed under vertex removal, or equivalently, under taking induced subgraphs. A hereditary graph property Π is *factorial* if there exist constants c_1, c_2, N such that $n^{c_1 n} \le |\Pi_n| \le n^{c_2 n}$ when $n > N$, where Π_n denotes the set of n-vertex labelled graphs in Π. A class is *subfactorial* if for every $c > 0$, $|\Pi_n| \le n^{cn}$ when n is sufficiently large.

The structure of subfactorial classes is rather simple and can be characterised as follows.

Theorem 42. [Alekseev, 1997; Balogh et al., 2000] *For every subfactorial hereditary class Π, there is a constant k (depending only on Π) such that for every graph $G \in \Pi$, there exists a partition of $V(G)$ into at most k subsets $V_1, \ldots, V_k$, where each subset V_i is either an independent set or a clique in G, and for any two distinct subsets V_i, V_j, there are either no edges or all possible edges between the vertices in V_i and the vertices in V_j.*

5.3 Subfactorial properties

Theorem 43. *For any subfactorial hereditary property* Π*, the* STABLE-Π *problem is solvable in polynomial time.*

Proof. The proof of this result is based on Theorem 42 and is similar to the proof in [Lozin, 2005] of the case where Π is a class of bipartite graphs. We therefore only sketch the proof.

Given a graph $G = (V, E)$, we want to determine if there is a partition $V = S \cup R$, such that S is an independent set and $G[R] \in \Pi$. Let k be the constant associated with the class Π. We call any partition of R satisfying Theorem 42 *canonical* and call the subsets in a canonical partition *bags*.

We start by picking a representative for each bag. There are $O(n^k)$ ways to do so. Once the set of representatives is fixed, which is our current set R, the adjacencies between the bags are defined by the adjacencies between their representatives. For each choice of at most k representatives, there are at most 2^k ways to choose the type for each bag (a clique or an independent set). Without loss of generality we may assume that for each vertex $v \in V - R$ there is at most one candidate bag for the inclusion of v, since otherwise any two "similar" bags can be replaced by a single bag containing both of them. If there is no candidate bag for v, we move it to S.

For the vertices v not in $R \cup S$ we proceed as follows: if v has a conflict in S (i.e. has a neighbour in S) we move it to the respective bag of R, and if v has a conflict in R (i.e. moving it to its candidate bag in R makes the partition of R non-canonical) we move it to S. If no vertex outside of $R \cup S$ has a conflict in S or R, then the rest of the task can be solved by a reduction to the 2SAT problem.

To this end, we associate with each vertex $v \notin R \cup S$ a Boolean variable x_v. For any two vertices $u, v \notin R \cup S$, we create a set of clauses in the following way. If u and v cannot both appear in R (because, for instance, they are adjacent, but their candidate bags are not) we create the clause $x_u \vee x_v$, and if they cannot both appear in S we create the clause $\overline{x}_u \vee \overline{x}_v$. It is easy to verify that the set of clauses created in this way is satisfiable if and only if there is proper partition of G in which every vertex v with $x_v = true$ is placed in S and the remaining vertices are placed in R. □

5.4 Minimal factorial properties

In this section, we discuss the complexity of STABLE-Π for minimal factorial hereditary classes Π. We investigate each case as set out in the introduction to this chapter.

The following cases have already been established in the literature.

Theorem 44. [Mahadev and Peled, 1995] *The* STABLE-$\mathcal{M}_1$ *problem is NP-complete.*

Theorem 45. [Brandstädt et al., 1998] *The* STABLE-$\mathcal{M}_5$ *problem is solvable in polynomial time.*

Further results in this section are based on the notion of Sparse-Dense partitions:

Theorem 46. (Sparse-Dense Theorem) [Alekseev et al., 2004; Feder et al., 2003] *For all positive integers k, l, there exists a polynomial time algorithm that, given a graph G, constructs all partitions of its vertex set into sets A, B such that $G[A]$ contains no independent set of size k and $G[B]$ contains no clique of size l.*

Namely, there are at most $n^{2R(k,l)-2}$ such partitions of an n-vertex graph G and all can be enumerated in time $O\big(n^{2R(k,l)+\max\{k,l\}}\big)$, where $R(k,l)$ denotes the Ramsey number of k and l.

Theorem 47. *The* STABLE-$\overline{\mathcal{M}_1}$, STABLE-$\overline{\mathcal{M}_2}$, *and* STABLE-$\overline{\mathcal{M}_3}$ *problems are solvable in polynomial time.*

Proof. Let $\Pi \in \{\overline{\mathcal{M}_1}, \overline{\mathcal{M}_2}, \overline{\mathcal{M}_3}\}$. All three problems ask to partition vertices of the input graph G into one independent set V_1, and a co-bipartite graph V_1' (consisting of two cliques V_2 and V_3). By Theorem 46, there are only polynomially many such partitions of $V(G)$ and all of them can be found in polynomial time. For each such partition, we test whether the co-bipartite subgraph of G induced V_1' is in Π. This yields a polynomial-time algorithm. □

Theorem 48. *The* STABLE-$\mathcal{M}_4$ *problem is solvable in polynomial time.*

Proof. We rephrase the problem as: given a graph G, decide whether the vertices of G can be partitioned into three sets V_1, V_2, V_3 such that V_3 is a clique, V_1 and V_2 are independent sets and every vertex in V_2 has at most one neighbour in V_3 and vice-versa.

Let G be the input graph. By Theorem 46, we can find, in polynomial time, the collection $\mathcal{P}$ of all partitions of the vertex set of G into a clique C and a set X such that $G[X]$ contains no clique of size three. Note that if G admits a STABLE-$\mathcal{M}_4$ partition V_1, V_2, V_3, then the partition $C = V_3$, $X = V_1 \cup V_2$ is a partition in $\mathcal{P}$. Thus to solve the problem, we try all partitions C, X in $\mathcal{P}$ by setting $V_3 = C$ and testing whether X can be split into V_1, V_2 so that V_1, V_2, V_3 is a STABLE-$\mathcal{M}_4$ partition of G.

Let C, X be a partition from $\mathcal{P}$. We construct an instance $\mathcal{I}$ of 2SAT:

(i) Create a variable x_v for every vertex $v \in X$,

(ii) for every edge $uv \in E(G[X])$, add the clauses $(x_u \vee x_v)$ and $(\overline{x_u} \vee \overline{x_v})$,

(iii) for every pair of vertices $u, v \in X$ with a common neighbour in C, add the clause $(\overline{x_u} \vee \overline{x_v})$,

(iv) for every vertex $v \in X$ such that v has at least two neighbours in C, add the clauses $(\overline{x_v} \vee \overline{a})$ and $(\overline{x_v} \vee a)$, where a is a new variable.

We claim that $\mathcal{I}$ has a satisfying assignment if and only if G admits a STABLE-$\mathcal{M}_4$ partition V_1, V_2, V_3 such that $V_3 = C$ and $V_1 \cup V_2 = X$.

Suppose that the instance $\mathcal{I}$ has a satisfying truth assignment φ. Namely, φ is a mapping from the variables of $\mathcal{I}$ to $\{true, false\}$ such that in every clause C_j, there is at least one literal that φ evaluates to $true$ (where the value $\varphi(\overline{z})$ is defined as the negation of $\varphi(z)$, for any variable z)

Define $V_1 = \{v \mid \varphi(x_v) = false\}$ and $V_2 = \{v \mid \varphi(x_v) = true\}$. We claim that V_1, V_2, V_3 is a STABLE-$\mathcal{M}_4$ partition of G. Indeed, by (ii), V_1 and V_2 are independent sets; by (iii), no two vertices in V_2 have a common neighbour in V_3; by (iv), every vertex from V_2 has at most one neighbour in V_3.

Conversely, let V_1, V_2, V_3 be a STABLE-$\mathcal{M}_4$ partition of G where $V_3 = C$. We define a truth assignment for $\mathcal{I}$ as follows. We set $\varphi(x_v) = false$ if $v \in V_1$ and $\varphi(x_v) = true$ if $v \in V_2$. For each of the new variables a defined in (iv) above, we set $\varphi(a) = true$. We claim that φ is a satisfying truth assignment for $\mathcal{I}$. Indeed, all clauses defined in (ii) are satisfied, since V_1 and V_2 are independent sets. Also, all clauses defined in (iii) are satisfied as every vertex in V_3 has at most one neighbour in V_2. Similarly, every vertex in V_2 has at most one neighbour in V_3 implying that all clauses in (iv) are satisfied. Thus $\mathcal{I}$ is satisfied by φ. That concludes the proof. □

A similar argument works for the complementary class and results in the following theorem.

Theorem 49. *The* STABLE-$\overline{\mathcal{M}_4}$ *problem is solvable in polynomial time.*

Proof. Similarly to the proof of Theorem 48, we can rephrase the problem as: given a graph G, decide whether the vertices of G can be partitioned into three sets V_1, V_2, V_3 such that V_3 is a clique, V_1 and V_2 are independent sets and every vertex in V_2 has at most one non-neighbour in V_3 and vice-versa.

Again, defining $\mathcal{P}$ as before, we solve the problem by trying all partitions C, X in $\mathcal{P}$. For each such partition we set $V_3 = C$ and test whether X can be split into V_1, V_2 so that V_1, V_2, V_3 is a STABLE-$\overline{\mathcal{M}_4}$ partition of G.

Let G'_C be the graph obtained from G by complementing (i.e. replacing edges by non-edges and vice versa) the edges between C and X. Now G has a STABLE-$\overline{\mathcal{M}_4}$ partition with $V_3 = C$ if and only if G'_C has a STABLE-$\mathcal{M}_4$ partition with V_3. Indeed, if $V_1 \cup V_2$ is a partition of X, then $G[C]$ is a clique and $G[V_1], G[V_2]$ are independent sets if and only if $G'_C[C]$ is a clique and $G'_C[V_1], G'_C[V_2]$ are independent sets. Further, each vertex in V_2 (resp. V_3) has at most one non-neighbour in V_3 (resp. V_2) in G if and only if it has at most one neighbour in V_3 (resp. V_2) in G'_C.

We now reduce the problem to an equivalent instance of 2SAT as in the proof of Theorem 48. This concludes the proof. □

We are left with the case of the STABLE-$\mathcal{M}_2$, which needs more work. We solve this in the following section.

5.4.1 The STABLE-$\mathcal{M}_2$ problem

In this section, we prove that the STABLE-$\mathcal{M}_2$ problem is solvable in polynomial time.

We cast the problem for the complement and solve (in polynomial time) a more general version with lists as follows.

An *instance* of the problem is a pair (G, ℓ) where G is a graph and $\ell : V(G) \to 2^{\{1,2,3\}}$. We say that $\ell(v)$ is the *list* belonging to the vertex v. For $S \subseteq \{1, 2, 3\}$, we let U^ℓ_S denote the set of vertices in G with $\ell(v) = S$.

Given an instance (G, ℓ), we seek to partition $V(G)$ into three cliques V_1, V_2, V_3 such that

- each vertex in V_2 has at most one neighbour in V_3,
- each vertex in V_3 has at most one neighbour in V_2, and
- for all $\alpha \in \{1, 2, 3\}$, each $v \in V_\alpha$ satisfies $\alpha \in \ell(v)$.

Input: Instance (G, ℓ) where G is a graph and $\ell(v) : V(G) \to 2^{\{1,2,3\}}$
Output: A reduced instance (G, ℓ)

for $\alpha \in \{1, 2, 3\}$ **do**
  if *for* $u \in U^{\ell}_{\{\alpha\}}$*, there exists* $v \in V(G) \setminus N(u)$ *with* $\alpha \in \ell(v)$ **then**
    remove α from $\ell(v)$ and **goto** 1
for $(\alpha, \beta) \in \Big\{(2, 3), (3, 2)\Big\}$ **do**
  if *for* $u \in U^{\ell}_{\{\alpha\}}$*, there exists* $v \in N(u) \cap U^{\ell}_{\{\beta\}}$ **then**
    for all $w \in N(u) \setminus \{v\}$ with $\beta \in \ell(w)$, **remove** β from $\ell(w)$
    for all $w \in N(v) \setminus \{u\}$ with $\alpha \in \ell(w)$, **remove** α from $\ell(w)$
    remove u, v from G and **goto** 1
  if *there exists* $v \in U^{\ell}_{\{1,\beta\}}$ *with* $|N(v) \cap U^{\ell}_{\{\alpha\}}| \geq 2$ **then**
    remove β from $\ell(v)$ and **goto** 1
  if *for* $u \in U^{\ell}_{\{\alpha\}}$*, there are* $v, w \in N(u) \cap U^{\ell}_{\{1,\beta\}}$ *where* $(N(v) \setminus N(w)) \cap U^{\ell}_{\{1,\alpha\}} \neq \emptyset$ **then**
    remove β from $\ell(v)$ and **goto** 1
  if *for* $u \in U^{\ell}_{\{\alpha\}}$*, there are* $v, w \in N(u) \cap U^{\ell}_{\{1,\beta\}}$ *and* $x \in U^{\ell}_{\{1,\alpha\}}$ *with* $v, w \notin N(x)$ **then**
    remove 1 from $\ell(x)$ and **goto** 1
  if *for* $u \in V(G)$ *with* $1 \in \ell(u)$*, the set* $U^{\ell}_{\{1,\alpha\}} \setminus N(u)$ *is not a clique* **then**
    remove 1 from $\ell(u)$ and **goto** 1
  if *for* $u \in V(G)$ *with* $\beta \in \ell(u)$*, the subgraph* $G\Big[N(u) \cap U^{\ell}_{\{1,\alpha\}}\Big]$ *contains an induced 4-cycle,* $2K_2$*, or* P_4 **then**
    remove β from $\ell(u)$ and **goto** 1
return (G, ℓ)

Algorithm 1: Reduction algorithm

If such a partition exists, we call it a *solution* for (G, ℓ). Note that if the list of some vertex is empty, then there is no solution for the problem instance. Thus for the rest of the proof, we assume that $U^{\ell}_{\emptyset} = \emptyset$.

To solve the problem, we consider several special cases and reduce the general case to these cases in polynomial time.

First, we consider the procedure depicted in Algorithm 1. We say that an instance (G, ℓ) is *reduced*, if it is the result of Algorithm 1.

We have the following claim:

Lemma 50. *Let (G, ℓ) be an instance and let (G', ℓ') be the result of applying Algorithm 1 to (G, ℓ). Then there exists a solution for (G, ℓ) if and only if there exists a solution for (G', ℓ').*

Proof. If $x \in U^{\ell}_{\{i\}}$ for some $i \in \{1, 2, 3\}$, then in any solution (V_1, V_2, V_3) of the instance, we must have $x \in V_i$.

Line 2: Let $\alpha \in \{1, 2, 3\}$. Since V_α must be a clique in any solution, if $u \in U^{\ell}_{\{\alpha\}}$ and u, v are not adjacent, then $v \notin V_\alpha$ for any solution for (G, ℓ).

Line 4: If u, v are adjacent for some $u \in U^{\ell}_{\{\alpha\}}$ and $v \in U^{\ell}_{\{\beta\}}$, then in any valid solution, these must be two matched vertices of V_2 and V_3. In this case v must be the unique neighbour of u in V_β and u must be the unique neighbour of v in V_α. We therefore remove either α or β from the list of each vertex in $N(u) \cup N(v) \setminus \{u, v\}$, as appropriate. We then remove u and v from G. The resulting instance has a solution if and only if the original one does.

Line 5: In any solution, if $v \in V_\beta$, then v can have at most one neighbour in V_α.

Line 6: Suppose $u \in U^{\ell}_{\{\alpha\}}$, such that $v, w \in N(u) \cap U^{\ell}_{\{1,\beta\}}$ and $z \in (N(v) \setminus N(w)) \cap U^{\ell}_{\{1,\alpha\}}$. If there were a solution in which $v \in V_\beta$ then since $u \in V_\alpha$ and every vertex in V_α can have at most one neighbour in V_β and vice versa, we must have $w, z \in V_1$. But this is impossible, since w, z are not adjacent. This contradiction implies that v cannot be in V_β.

Line 7: Suppose $u \in U^{\ell}_{\{\alpha\}}, x \in U^{\ell}_{\{1,\alpha\}}$ and $v, w \in (N(u) \setminus N(x)) \cap U^{\ell}_{\{1,\beta\}}$. Then in any solution we must have $u \in V_\alpha$. Since u can have at most one neighbour in V_β, at least one of v, w must be in V_1. But V_1 is a clique and v, w are nonadjacent to x. Thus $x \notin V_1$.

Line 8: Suppose $u \in V(G)$ with $1 \in \ell(u)$ and $v, w \in U^{\ell}_{\{1,\alpha\}} \setminus N(u)$ with v, w non-adjacent. Since for any solution, V_i must be a clique for $i \in \{1, 2, 3\}$, exactly one of v, w must be in V_1 and the other in V_α. But u is non-adjacent to both v and w, so $u \notin V_1$.

Line 9: Suppose $\beta \in \ell(u)$. In any solution, if $u \in V_\beta$ then $N(u) \cap V_1$ must be a clique and u can have at most one neighbour in V_α. The 4-cycle, $2K_2$ and P_4 are neither cliques, nor are they partitionable into a clique and a single vertex. Thus if any of these three graphs is an induced subgraph of $N(u) \cap U^{\ell}_{\{1,\alpha\}}$, then any solution must satisfy $u \notin V_\beta$. □

Note that Algorithm 1 has polynomial running time. This allows us to assume that the instance we consider is always reduced. (If not, we use Algorithm 1 to produce an equivalent reduced instance.)

Assuming this, we consider the some special cases of the problem, which we will later use as steps in finding a solution for the general problem.

Lemma 51. *If there exists a solution (V_1, V_2, V_3) for the reduced instance (G, ℓ), such that there is no edge between a vertex in V_2 and a vertex in V_3, it can be found in polynomial time.*

Proof. This amounts to finding a partition of $\overline{G}$ into an independent set and a complete bipartite graph, in a way that respects the lists of the vertices. This can been solved in polynomial time [Feder et al., 2003]. $\square$

Lemma 52. *If $U^\ell_{\{1,2,3\}} = U^\ell_{\{2,3\}} = \emptyset$, and $U^\ell_{\{1,2\}} = \emptyset$ or $U^\ell_{\{1,3\}} = \emptyset$, and the instance is reduced, the problem can be solved in polynomial time.*

Proof. We may assume by symmetry that $U^\ell_{\{1,3\}} = \emptyset$ and we reduce the problem to an instance of 2Sat constructed as follows.

- For each vertex $x \in U^\ell_{\{1,2\}}$, introduce a new variable v_x.
- For all $z \in U^\ell_{\{3\}}$ and all $x, y \in N(z) \cap U^\ell_{\{1,2\}}$, add clause $(\neg v_x \vee \neg v_y)$.
- For all $x, y \in U^\ell_{\{1,2\}}$ with $xy \notin E(G)$, add clauses $(v_x \vee v_y)$, $(\neg v_x \vee \neg v_y)$.

Since (G, ℓ) is a reduced instance, it has a solution if and only if the above instance of 2Sat is satisfiable. In particular, if φ is a satisfying assignment, the following sets (V_1, V_2, V_3) form a solution for (G, ℓ).

$$V_1 = U^\ell_{\{1\}} \cup \{x \mid \varphi(v_x) = false\} \qquad V_2 = U^\ell_{\{2\}} \cup \left(U^\ell_{\{1,2\}} \setminus V_1\right) \qquad V_3 = U^\ell_{\{3\}}$$ $\square$

Lemma 53. *If $U^\ell_{\{1,2,3\}} = U^\ell_{\{2,3\}} = \emptyset$ and $U^\ell_{\{1,2\}}$, $U^\ell_{\{1,3\}}$ are cliques of G, and the instance is reduced, the problem can be solved in polynomial time.*

Proof. We show that the following is a solution for (G, ℓ)

$$V_1 = U^\ell_{\{1\}} \cup U^\ell_{\{1,2\}} \cup \bigcup_{\substack{u \in U^\ell_{\{2\}} \\ |N(u) \cap U^\ell_{\{1,3\}}| \geq 2}} \left(N(u) \cap U^\ell_{\{1,3\}}\right)$$

$$V_2 = U^\ell_{\{2\}} \qquad V_3 = U^\ell_{\{3\}} \cup \left(U^\ell_{\{1,3\}} \setminus V_1\right)$$

Indeed, note that the instance (G, ℓ) is reduced. By Line 2 of Algorithm 1 and the fact that $U^\ell_{\{1,3\}}$ is a clique, we conclude that V_2 and V_3 must be cliques. By Line 4 of Algorithm 1 and the definition of V_1 and V_3, every vertex in V_2 has at most one neighbour in V_3. By Lines 4 and 5 of Algorithm 1, each vertex of V_3 has at most one neighbour in V_2. By Line 2 of Algorithm 1 and since $U^\ell_{\{1,2\}}, U^\ell_{\{1,3\}}$ are cliques, we need only verify that every vertex in $V_1 \cap U^\ell_{\{1,2\}}$ is adjacent to every vertex in $V_1 \cap U^\ell_{\{1,3\}}$. We therefore assume that these sets are not empty. Let $u \in U^{\ell}_{\{2\}}$ and $v, w \in N(u) \cap U^{\ell}_{\{1,3\}}$. By Line 7 of Algorithm 1, any vertex in $U^{\ell}_{\{1,2\}}$ must be adjacent to at least one of v or w. But by Line 6 of Algorithm 1, the vertices v, w have the same neighbourhood in $U^{\ell}_{\{1,2\}}$. Thus every vertex of $U^{\ell}_{\{1,2\}}$ must be adjacent to every vertex of $V_1 \cap U^{\ell}_{\{1,3\}}$. We therefore conclude that V_1 is indeed a clique. $\square$

We can now generalise Lemmas 52 and 53 as follows:

Lemma 54. *If $U^{\ell}_{\{1,2,3\}} = U^{\ell}_{\{2,3\}} = \emptyset$, and the problem instance is reduced, the problem can be solved in polynomial time.*

Proof. Assume that $U^{\ell}_{\{1,2,3\}} = U^{\ell}_{\{2,3\}} = \emptyset$, but Lemma 52 does not apply. Thus $U^{\ell}_{\{1,2\}} \neq \emptyset$ and $U^{\ell}_{\{1,3\}} \neq \emptyset$. We fix any $u \in U^{\ell}_{\{1,2\}}$. Then we either do nothing, or choose $w \in N(u) \cap U^{\ell}_{\{1,3\}}$ and set $\ell(w) = \{3\}$. After that, we remove 3 from $\ell(v)$ for each $v \in N(u)$ that belongs to a non-trivial (≥ 2 vertices) connected component of $\overline{G}\left[U^{\ell}_{\{1,3\}}\right]$ unless that component contains w (if w exists).

If after these modifications $U^{\ell}_{\{1,3\}}$ is still non-empty, we similarly fix $u' \in U^{\ell}_{\{1,3\}}$, do nothing or set $\ell(w') = \{2\}$ for some $w' \in N(u') \cap U^{\ell}_{\{1,2\}}$, and then remove 2 from $\ell(v)$ for each $v \in N(u') \cap U^{\ell}_{\{1,2\}}$ in a non-trivial component of $\overline{G}\left[U^{\ell}_{\{1,2\}}\right]$ unless that component contains w' (if w' exists).

We try all possible choices for w and w', creating $O(n^2)$ instances. It follows that the initial instance has a solution if and only if one of these $O(n^2)$ instances has.

Consider the $O(n^2)$ instances produced in this way from the initial instance (G, ℓ). First, we show that (G, ℓ) has a solution if and only if (at least) one of the $O(n^2)$ instances has a solution.

Clearly, if one of the $O(n^2)$ instances has a solution, then this is also a solution for (G, ℓ), since during the construction of the instances, we only remove elements from lists.

Conversely, suppose that there exists a solution V_1, V_2, V_3 for (G, ℓ). Let $H = G[U^{\ell}_{\{1,3\}}]$, i.e. H denotes the subgraph of G induced by $U^{\ell}_{\{1,3\}}$, and consider the vertex $u \in U^{\ell}_{\{1,2\}}$ that we fix.

Case(i): Suppose that $u \in V_1$. There are two possibilities to consider. First, suppose that there exists a neighbour of u that is in V_3 and also in some non-trivial connected component of $\overline{H}$. Consider the instance where we choose w to be this neighbour (We shall henceforth refer to it as the "modified" instance.) In this instance, we remove 3 from each neighbour of u in $V(H) = U^{\ell}_{\{1,3\}}$ that belongs to a non-trivial connected component of $\overline{H}$ unless that component contains w.

We claim that each such neighbour v belongs to V_1. Suppose otherwise. Then v belongs to V_3, since $\ell(v) = \{1, 3\}$. Recall that v is in a non-trivial connected component of $\overline{H}$. Thus it has a neighbour z in $\overline{H}$. We conclude that z is non-adjacent to v in H, and hence, in G. If z is also non-adjacent to u, then z can be neither in V_1 nor in V_3, as these are both cliques. But then V_1, V_2, V_3 cannot be a solution for (G, ℓ) as $\ell(z) = \{1, 3\}$. So, we conclude that z is adjacent to u.

Now, recall that w is also in a non-trivial connected component of $\overline{H}$. So, w has a neighbour x in this component, and we conclude that $xw \notin E(G)$. This implies $ux \in E(G)$ as otherwise V_1, V_2, V_3 is not a solution. But now x, z, w, v induce a 4-cycle in the neighbourhood of u, which is impossible by Line 9 of Algorithm 1. (For this, recall that (G, ℓ) is a reduced instance and that the connected component of $\overline{H}$ containing w and x is different from the one containing v and z.)

This proves that V_1, V_2, V_3 is also a solution to the modified instance. As this is one of the $O(n^2)$ instances, we are done.

So, we may assume that each neighbour of u in $V_3 \cap V(H)$ is itself a connected component (isolated vertex) of $\overline{H}$. In this case, we consider the instance where we do not choose w (referred to as the "modified" instance). In this instance, we remove 3 from each neighbour of u in $V(H)$ that belongs to a non-trivial connected component of $\overline{H}$. By our assumption, this does not modify the lists of the neighbours of u in $V_3 \cap V(H)$. Thus V_1, V_2, V_3 is a solution to the modified instance, and we are again done.

Case(ii): Suppose that $u \in V_2$. If u has a neighbour in $V_3 \cap V(H)$, consider the instance where w is chosen to be this neighbour (referred to as the "modified" instance). In this instance, we remove 3 from each neighbour of u in a non-trivial connected component of $\overline{H}$ unless that component contains w. Clearly, any such vertex v cannot belong to V_3, since then u has two neighbours in V_3, which is impossible. Thus V_1, V_2, V_3 is also a solution to the modified instance, and we are done.

Finally, suppose that u has no neighbour in $V_3 \cap V(H)$, and consider the instance where we do not choose w. Again, we remove 3 from every neighbour of u in a non-trivial component of $\overline{H}$, and conclude that V_1, V_2, V_3 is a solution to this modified instance, since we assume that $N(u) \cap V(H) \cap V_3 = \emptyset$. This completes all cases.

This proves that one of the choices for w must succeed if (G, ℓ) has a solution. By a symmetric argument, it follows that, for an appropriate choice of w, one of the choices for w' (if at all we consider w') must also succeed. This concludes the first argument.

For the second argument, consider one of the $O(n^2)$ instances (G^+, ℓ^+). We constructed this instance from the initial instance (G, ℓ), by fixing a vertex u and choosing

w (or not), and then fixing a vertex u' (if possible) and choosing w' (or not). We also reduced this instance using Algorithm 1.

We shall now prove that $U^{\ell+}_{\{1,2\}}$ and $U^{\ell+}_{\{1,3\}}$ are both cliques of G, i.e. that Lemma 53 can be applied. Suppose otherwise, and assume first that $U^{\ell+}_{\{1,3\}}$ contains non-adjacent vertices v, v'. As ℓ^+ is a reduction of ℓ, we conclude that v, v' are also vertices in $U^{\ell}_{\{1,3\}}$. Again, use H to denote the graph $G[U^{\ell}_{\{1,3\}}]$.

First, we observe that u is adjacent to at least one of v, v'. Indeed, if u is non-adjacent to both v and v', then 1 was removed from $\ell(u)$ in Line 8 of Algorithm 1 (recall that (G, ℓ) is a reduced instance). This is impossible as $\ell(u) = \{1, 2\}$. By symmetry, we shall assume that u is adjacent to v.

Now, if w was not chosen when constructing (G^+, ℓ^+), then 3 was removed from all neighbours of u in non-trivial connected components of $\overline{H}$. One of these components contains both v and v' as they are non-adjacent, and so 3 was removed from $\ell(v)$ when constructing ℓ^+ (recall that we assume that u is adjacent to v). However, this is impossible, since $\ell^+(v) = \{1, 3\}$. We similarly arrive at a contradiction when w is chosen, but it is not a vertex of the connected component of $\overline{H}$ containing v. So we conclude that w was chosen from the connected component of $\overline{H}$ containing v. But now, we have that either $v = w$, or, since (G^+, ℓ^+) is reduced, 1 or 3 was removed from $\ell(v)$ in Line 2 at some point when running Algorithm 1 to produce the instance (G^+, ℓ^+). This is, of course, impossible as $\ell(w) = \{3\}$ and $\ell^+(v) = \{1, 3\}$. This concludes the argument for $U^{\ell+}_{\{1,3\}}$.

The argument for $U^{\ell+}_{\{1,2\}}$ is similar, using u' and w'. Finally, note that if u' (and hence w') cannot be chosen because the first modification of lists removed all candidates, then the Lemma 52 can be applied. $\square$

We are ready to discuss the general case and prove the main theorem of this section.

Theorem 55. *The* STABLE-$\mathcal{M}_2$ *problem is solvable in polynomial time.*

Proof. First, we test whether or not (G, ℓ) we are in the situation of Lemma 51. If so, we find a solution for (G, ℓ) using [Feder et al., 2003]. If not, we conclude that if there is a solution (V_1, V_2, V_3) for (G, ℓ), then there must exist $u \in V_2$ and $v \in V_3$ with $uv \in E(G)$. We try all possible choices for such a pair u, v. This reduces the problem to solving $O(n^2)$ separate instances. For each such choice u, v, we set $\ell(u) = \{2\}$, $\ell(v) = \{3\}$, and run Algorithm 1. If the list of some vertex becomes empty, we reject this choice of u, v. Otherwise, we observe that the resulting reduced instance (G^+, ℓ^+) satisfies

$U^{\ell+}_{\{1,2,3\}} = U^{\ell+}_{\{2,3\}} = \emptyset$. So we can apply Lemma 54 and we try to find a solution for (G^+, ℓ^+), if it exists, as described above. This concludes the proof of Theorem 55. $\square$

5.5 Restricted Minimal Factorial Properties

First, we briefly examine the polynomial-time cases. Using essentially the same arguments as in the proof of Theorem 47, we obtain the following.

Theorem 56. *The* STABLE-$\overline{\mathcal{M}_1^S}$ *and* STABLE-$\overline{\mathcal{M}_2^S}$ *problems are solvable in polynomial time.*

All the remaining cases are hard. We discuss them in separate claims.

All the subsequent proofs will be essentially along the same lines and based on the following useful lemma.

Lemma 57. *Any instance of* ONE-IN-THREE-3SAT *can be transformed in polynomial time to an equivalent instance of* ONE-IN-THREE-3SAT *such that:*

(i) *There is no clause of the form $(X \vee X \vee Y)$ or $(X \vee \overline{X} \vee Y)$ where X and Y are (not necessarily distinct) literals.*

(ii) *If X appears in some clause, then $\overline{X}$ also appears in some clause.*

(iii) *Every literal appears at least twice in the instance.*

(iv) *There are at least 4 clauses and at least 4 variables in the instance.*

Proof. Apply the following steps in order. First, for each clause of the form $(X \vee X \vee Y)$, replace it by the clauses $(u \vee v \vee X)$, $(\overline{u} \vee \overline{v} \vee X)$, $(w \vee z \vee \overline{Y})$, $(\overline{w} \vee \overline{z} \vee \overline{Y})$, where u, v, w, z are new variables. Next, for each clause of the form $(X \vee \overline{X} \vee Y)$, replace it by the clauses $(u \vee v \vee \overline{Y})$, $(\overline{u} \vee \overline{v} \vee \overline{Y})$, where u, v are new variables. Then, for each literal X, add the clauses $(u \vee v \vee X)$, $(\overline{u} \vee w \vee \overline{X})$, $(\overline{v} \vee w \vee z)$, $(v \vee \overline{w} \vee z)$, $(v \vee w \vee \overline{z})$, where u, v, w, z are new variables. Note that since the original instance was non-empty, the new instance must now have at least 4 clauses and at least 4 variables. Finally, make a copy of each clause, i.e. make each clause appear twice in the instance.

It is easy to see that the instance produced in this way is equivalent to the original instance and satisfies all the conditions of the lemma. $\square$

Theorem 58. *The* STABLE-$\mathcal{M}_4^S$ *problem is NP-complete.*

Proof. We can rephrase the problem as follows: given a graph G, decide whether the vertices of G can be partitioned into 3 sets V_1, V_2, V_3 such that V_3 is a clique, V_1 and V_2 are independent sets and the edges between V_2 and V_3 form a perfect matching.

The proof proceeds by reduction from One-In-Three-3Sat. Consider an instance $\mathcal{I}$ of the problem, namely the instance consists of m clauses $C_1, \ldots, C_m$ containing variables $v_1, \ldots, v_n$. We may assume it satisfies the properties listed in Lemma 57. Let J_i denote the set of indices j such that v_i appears in C_j. Let $\overline{J_i}$ denote the indices j such that $\overline{v_i}$ appears in C_j.

For the instance $\mathcal{I}$, we construct the graph $G_{\mathcal{I}}$ as follows. First, we create a complete graph on vertices $y_1, \ldots, y_m$. Then for every occurrence of a variable v_i (resp. $\overline{v_i}$) in a clause C_j, we add a new vertex $x_{i,j}$ (resp. $\overline{x_{i,j}}$) and we add an edge between y_j and $x_{i,j}$ (resp. $\overline{x_{i,j}}$). Finally, we add an edge between $x_{i,j}$ and $\overline{x_{i,\ell}}$ for all $i \in \{1, \ldots, n\}$, all $j \in J_i$ and all $\ell \in \overline{J_i}$.

We prove that $G_{\mathcal{I}}$ admits a Stable-$\mathcal{M}_3^S$ partition if and only if $\mathcal{I}$ has a satisfying truth assignment (as an instance of One-In-Three-3Sat).

Suppose that the instance $\mathcal{I}$ has a satisfying truth assignment φ. In other words, φ is a mapping from $\{v_1, \ldots, v_n\}$ to $\{true, false\}$ such that for every clause C_j, φ evaluates exactly one of the literals in C_j to *true*, where $\varphi(\overline{v_i})$ is defined as the negation of $\varphi(v_i)$.

Let us define a partition of $V(G_{\mathcal{I}})$ as follows:

$$V_1 = \Big\{x_{i,j} \;\Big|\; j \in J_i \wedge \varphi(v_i) = false\Big\} \cup \Big\{\overline{x_{i,j}} \;\Big|\; j \in \overline{J_i} \wedge \varphi(v_i) = true\Big\},$$
$$V_2 = \Big\{x_{i,j} \;\Big|\; j \in J_i \wedge \varphi(v_i) = true\Big\} \cup \Big\{\overline{x_{i,j}} \;\Big|\; j \in \overline{J_i} \wedge \varphi(v_i) = false\Big\},$$
$$V_3 = \Big\{y_j \;\Big|\; j \in \{1, \ldots, m\}\Big\}.$$

It is not difficult to verify that V_1 and V_2 are independent sets of $G_{\mathcal{I}}$, that V_3 is a clique, and that the edges between V_2 and V_3 form a perfect matching. Indeed, each vertex y_j in V_3 is adjacent to a unique vertex $x_{i,j}$ or $\overline{x_{i,j}}$ in V_2, namely the one for which v_i, resp. $\overline{v_i}$ is the literal of C_j that φ evaluates to *true*. Thus $G_{\mathcal{I}}$ admits a Stable-$\mathcal{M}_3^S$ partition as required.

Conversely, suppose that $G_{\mathcal{I}}$ admits a Stable-$\mathcal{M}_3^S$ partition. In other words, there exists a partition of $V(G_{\mathcal{I}})$ into three sets V_1, V_2, V_3 such that V_1, V_2 are independent sets, V_3 is a clique, and the edges between V_2 and V_3 form a perfect matching.

First, we show that we must have $V_3 = \{y_j \mid j \in \{1, \ldots, m\}\}$. By Lemma 57, there are at least four y_j's. Thus, since V_1 and V_2 are independent sets, V_3 must contain at least two y_j's. This implies that V_3 contains no $x_{i,j}$ or $\overline{x_{i,j}}$, since each has at most one neighbour in $\{y_1, \ldots, y_m\}$ and V_3 is a clique. It also implies that if $y_j \in V_2$ for some j, then y_j has at least 2 neighbours in V_3, which is a contradiction. Finally, suppose that $y_j \in V_1$ for some j. Consider a neighbour $z \notin \{y_1, \ldots, y_m\}$ of y_j. (Note that z is $x_{i,j}$ or $\overline{x_{i,j}}$ for some i and there are exactly three such vertices). Then z is not in V_3, since V_3 contains no $x_{i,j}$ or $\overline{x_{i,j}}$. Also, z cannot be in V_1, since V_1 is independent. Thus z must be in V_2. But z has a unique neighbour in $\{y_1, \ldots, y_m\}$, namely y_j, and hence, z does not have a neighbour in V_3, a contradiction. This proves that $V_3 = \{y_1, \ldots, y_m\}$.

Now, we define the following truth assignment $\varphi : \{v_1, \ldots, v_n\} \to \{true, false\}$. For each $i \in \{1, \ldots, n\}$, we set $\varphi(v_i) = true$ if $x_{i,j} \in V_2$ for some j, and set $\varphi(v_i) = false$ otherwise. We prove that φ is a satisfying truth assignment for the instance $\mathcal{I}$, which will conclude the proof.

Using the assignment φ, we prove that

$$V_1 = \Big\{x_{i,j} \;\Big|\; j \in J_i \wedge \varphi(v_i) = false\Big\} \cup \Big\{\overline{x_{i,j}} \;\Big|\; j \in \overline{J_i} \wedge \varphi(v_i) = true\},$$
$$V_2 = \Big\{x_{i,j} \;\Big|\; j \in J_i \wedge \varphi(v_i) = true\Big\} \cup \Big\{\overline{x_{i,j}} \;\Big|\; j \in \overline{J_i} \wedge \varphi(v_i) = false\Big\}.$$

To show this, recall that for each $i \in \{1, \ldots, n\}$, every $x_{i,j}$ is adjacent to every $\overline{x_{i,\ell}}$ where $j \in J_i$ and $\ell \in \overline{J_i}$. Thus if $\varphi(v_i) = true$, then $x_{i,j} \in V_2$ for some j which implies $\overline{x_{i,\ell}} \in V_1$ for all $\ell \in \overline{J_i}$, since V_2 is an independent set. Therefore, $x_{i,j} \in V_2$ for all $j \in J_i$, since V_1 is an independent set. Similarly, if $\varphi(v_i) = false$, then $x_{i,j} \in V_1$ for all $j \in J_i$, and hence, $\overline{x_{i,\ell}} \in V_2$ for all $\ell \in \overline{J_i}$.

Now, consider a clause C_j. Recall that $y_j \in V_3$, and hence, it has exactly one neighbour $x_{i,j}$ or $\overline{x_{i,j}}$ in V_2 corresponding to the literal v_i, resp. $\overline{v_i}$ in C_j, which φ evaluates to *true* by the above. So, all other neighbours $x_{i',j}$ or $\overline{x_{i',j}}$ of y_j belong to V_1 and thus correspond to literals $v_{i'}$, resp. $\overline{v_{i'}}$ which φ evaluates to $false$. This proves that C_j is satisfied by φ, and thus, proves that φ is a satisfying truth assignment.

That concludes the proof. □

A similar constructions also work for the two following cases:

Theorem 59. *The* STABLE-$\overline{\mathcal{M}_4^S}$ *problem is NP-complete.*

Proof. Again, we rephrase the problem as: given a graph G, decide whether the vertices of G can be partitioned into 3 sets V_1, V_2, V_3 such that V_3 is a clique, V_1 and V_2 are independent sets and the edges between V_2 and V_3 form the complement of a perfect matching.

The proof will now follow essentially the same steps as the proof of Theorem 58. We proceed by reduction from ONE-IN-THREE-3SAT.

Consider an instance $\mathcal{I}$ of the problem, namely the instance consists of m clauses $C_1, \dots, C_m$ containing variables $v_1, \dots, v_n$. Again, we may assume it satisfies the properties listed in Lemma 57, and we define J_i to be the set of indices j such that v_i appears in C_j, and define $\overline{J_i}$ to be the set of indices j such that $\overline{v_i}$ appears in C_j.

For the instance $\mathcal{I}$, consider the graph $G_{\mathcal{I}}$ constructed in the proof of Theorem 58. Let $G^+_{\mathcal{I}}$ be the graph constructed from $G_{\mathcal{I}}$ by complementing the edges between $\{y_1, \dots, y_m\}$ and the rest of the graph. Namely, for each $i \in \{1, \dots, m\}$, the vertex y_i is adjacent to $z \notin \{y_1, \dots, y_m\}$ in $G^+_{\mathcal{I}}$ if and only if y_i is not adjacent to z in $G_{\mathcal{I}}$. All other edges remain the same.

We prove that $G^+_{\mathcal{I}}$ admits a Stable-$\overline{\mathcal{M}^S_4}$ partition if and only if $\mathcal{I}$ has a satisfying truth assignment (as an instance of One-In-Three-3Sat).

For the forward direction, we note that, by the proof of Theorem 58, if $G_{\mathcal{I}}$ admits a Stable-$\mathcal{M}^S_4$ partition V_1, V_2, V_3, then $V_3 = \{y_1, \dots, y_m\}$. Thus, this is also a Stable-$\overline{\mathcal{M}^S_4}$ partition of $G^+_{\mathcal{I}}$. This proves that if $\mathcal{I}$ has a satisfying truth assignment, then $G^+_{\mathcal{I}}$ admits a Stable-$\overline{\mathcal{M}^S_4}$ partition.

Conversely, suppose that $G^+_{\mathcal{I}}$ admits a Stable-$\overline{\mathcal{M}^S_4}$ partition. Namely, let V_1, V_2, V_3 be a partition of $V(G_{\mathcal{I}})$ such that V_1, V_2 are independent sets, V_3 is a clique, and the edges between V_2 and V_3 form the complement of a perfect matching.

We shall prove that $V_3 = \{y_1, \dots, y_m\}$. By the construction of $G^+_{\mathcal{I}}$, this will imply that V_1, V_2, V_3 is also a Stable-$\mathcal{M}^S_3$ partition of $G_{\mathcal{I}}$. Thus, by the proof of Theorem 58, this will allow us to conclude that $\mathcal{I}$ has a satisfying truth assignment.

Consider a vertex y_j. By Lemma 57, there is a variable v_i such that neither v_i nor $\overline{v_i}$ appears in the clause C_j. Moreover, v_i appears as a literal in at least two clauses, say C_{j_1} and C_{j_2}, and $\overline{v_i}$ appears in two other clauses, say C_{j_3} and C_{j_4}. This implies that $G^+_{\mathcal{I}}$ contains vertices $x_{i,j_1}, x_{i,j_2}, \overline{x_{i,j_3}}, \overline{x_{i,j_4}}$ which induce a 4-cycle and are all adjacent to y_j. Suppose that $y_j \in V_1$. Since V_1 is an independent set, we conclude that $x_{i,j_1}, x_{i,j_2}, \overline{x_{i,j_3}}, \overline{x_{i,j_4}} \in V_2 \cup V_3$. However, this contradicts the fact that $G^+_{\mathcal{I}}[V_2 \cup V_3]$ is a split graph. Thus $y_j \notin V_1$. By the same argument, $y_j \notin V_2$. This proves that $V_3 \supseteq \{y_1, \dots, y_m\}$. Furthermore, note that V_3 contains no $x_{i,j}$ or $\overline{x_{i,j}}$, since each has a non-neighbour in $\{y_1, \dots, y_m\}$ and V_3 is a clique. Thus $V_3 = \{y_1, \dots, y_m\}$ as promised.

That concludes the proof. $\square$

Theorem 60. *The* Stable-$\mathcal{M}^S_2$ *problem is NP-complete.*

Proof. Once again we rephrase the problem as: given a graph G, decide if we can partition its vertex set into 3 independent sets V_1, V_2, V_3, such that the edges between V_2

and V_3 form the complement of a perfect matching. As before, we reduce from One-In-Three-3Sat.

Consider an instance $\mathcal{I}$ of the problem, namely the instance consists of m clauses $C_1, \dots, C_m$ containing variables $v_1, \dots, v_n$. Again, we may assume it satisfies the properties listed in Lemma 57. We define J_i to be the set of indices j such that v_i appears in C_j, and define $\overline{J_i}$ to be the set of indices j such that $\overline{v_i}$ appears in C_j.

For the instance $\mathcal{I}$, consider the graph $G_{\mathcal{I}}^{+}$ constructed in the proof of Theorem 59. Construct the graph $G_{\mathcal{I}}^{*}$ from $G_{\mathcal{I}}^{+}$ by removing all edges of the form $y_i y_j, i, j \in \{1, \dots, m\}$ (effectively replacing the clique on $\{y_1, \dots, y_m\}$ by an independent set). All other edges remain the same.

We claim that $G_{\mathcal{I}}^{*}$ has a Stable-$\mathcal{M}_2^S$ partition if and only if $\mathcal{I}$ has a satisfying truth assignment (as an instance of One-In-Three-3Sat).

For the forward direction, we note that, by the proof of Theorem 59, if $G_{\mathcal{I}}$ admits a Stable-$\overline{\mathcal{M}_4^S}$ partition V_1, V_2, V_3, then $V_3 = \{y_1, \dots, y_m\}$. Thus, this is also a Stable-$\mathcal{M}_2^S$ partition of $G_{\mathcal{I}}^{*}$. This proves that if $\mathcal{I}$ has a satisfying truth assignment, then $G_{\mathcal{I}}^{*}$ admits a Stable-$\mathcal{M}_2^S$ partition.

Now suppose, conversely, that $G_{\mathcal{I}}^{*}$ admits a Stable-$\mathcal{M}_2^S$ partition. In other words, $V(G_{\mathcal{I}}^{*})$ can be partitioned into three independent sets V_1, V_2, V_3, such that the edges between V_2 and V_3 form the complement of a perfect matching.

First, observe that if $a, b, c \in V_2 \cup V_3$ is an independent set then either all of them are contained in V_2 or all of them are contained in V_3. Indeed, suppose, without loss of generality that, $a, b \in V_2$ and $c \in V_3$, then c would have two non-neighbours in V_2, contradicting the fact that the edges between V_2 and V_3 form the complement of a perfect matching.

Next, we show that $y_j \in V_2$ for all $j \in \{1, \dots, m\}$ or $y_j \in V_3$ for all $j \in \{1, \dots, m\}$. By the above observation and Lemma 57, we need only show that $y_i \notin V_1$. Suppose, for contradiction, that $y_i \in V_1$. By Lemma 57, there must be vertices $x_{i_1,j_1}, x_{i_2,j_2}, x_{i_3,j_3}$ and $\overline{x_{i_1,j_4}}$ (with i_1, i_2, i_3 pairwise distinct), none of which correspond to literals in the clause C_j (i.e. $j_1, j_2, j_3, j_4 \neq j$). Since they do not correspond to these literals, y_j must be adjacent to all of these vertices, so $x_{i_1,j_1}, x_{i_2,j_2}, x_{i_3,j_3}, \overline{x_{i_1,j_4}} \in V_2 \cup V_3$. But $x_{i_1,j_1}, x_{i_2,j_2}, x_{i_3,j_3}$ and $x_{i_2,j_2}, x_{i_3,j_3}, \overline{x_{i_1,j_4}}$ are both independent sets of size 3. Thus all four of these vertices must be members of the same set V_i $(i \in \{2, 3\})$. But x_{i_2,j_2} and $\overline{x_{i_1,j_4}}$ are adjacent, contradicting the fact that V_2 and V_3 are independent sets.

Hence, we may conclude, without loss of generality, that $\{y_1, \dots, y_m\} \subseteq V_3$. Notice that, since each vertex $x_{i,j}$ or $\overline{x_{i,j}}$ corresponds to a unique occurrence of a literal in a unique clause in $\mathcal{I}$, every vertex not of the form y_j has a neighbour in V_3. Thus, since V_3 is an independent set, $V_3 = \{y_1, \ldots, y_m\}$. Finally, note that since V_1, V_2, V_3 is a STABLE-$\mathcal{M}_2^S$ partition for $G_{\mathcal{I}}^*$ and $V_3 = \{y_1, \ldots, y_m\}$, then by the construction of $G_{\mathcal{I}}^*$, V_1, V_2, V_3 must also be a STABLE-$\overline{\mathcal{M}_4^S}$-partition of $G_{\mathcal{I}}^+$. Thus, by the proof of Theorem 59, $\mathcal{I}$ must have a satisfying assignment.

This concludes the proof. □

5.6 Supplement: A superfactorial subclass of chordal bipartite graphs

It is worth noting that in addition to subfactorial and factorial classes, there are also hereditary classes Π that are superfactorial i.e. for any $c > 0$, Π has more than n^{cn} labelled graphs on n vertices if n is sufficiently large. Finding minimal such classes is a challenging open problem.

It is known that chordal bipartite graphs are superfactorial [Spinrad, 1995]. Chordal bipartite graphs are bipartite graphs with no chordless cycles of length 6 or more, i.e. they are the $(C_3, C_5, C_6, C_7, \ldots)$-free graphs. In this section we show that this class is not a minimal superfactorial class by finding a superfactorial class which is a proper subclass of bipartite chordal graphs. On the other hand, all other proper subclasses of chordal bipartite graphs that have been studied in the literature, such as forests, bipartite permutation graphs, bipartite distance-hereditary graphs and convex graphs, are factorial.

In order to derive a lower bound on the number of n-vertex chordal bipartite graphs, Spinrad counted in [Spinrad, 1995] the number of bipartite adjacency matrices representing these graphs, i.e. binary matrices whose rows correspond to one part of the graph and columns correspond to the other part. In particular, he used [Spinrad, 1995] the following construction.

Let M be a $2n$ by $2n$ binary matrix. Divide it into four n by n quadrants. Place an arbitrary perfect matching in the upper left quadrant and a matrix with all values equal to 1 in the lower right quadrant. Repeat this construction recursively within the other two quadrants. Let us denote the set of matrices constructed in this way by M^* and the set of bipartite graphs represented by these matrices by $\mathcal{Y}^*$.

Spinrad [1995] showed hat the number of matrices in M^*, and therefore the number of n-vertex graphs in $\mathcal{Y}^*$, is $\Omega(2^{\Omega(n \log^2 n)})$. He also showed that every graph in $\mathcal{Y}^*$ is chordal bipartite, which implies in particular a superfactorial lower bound for the number of n-vertex chordal bipartite graphs. However, as we show below, not every chordal bipartite graph occurs as an induced subgraph of a graph in $\mathcal{Y}^*$.

We let $2C_4$ denote the graph consisting of two disjoint copies of C_4 and let $C_4 - C_4$ denote the graph obtained from $2C_4$ by adding exactly one edge connecting vertices from different C_4's (see Figure 5.1).

(a) $2C_4$ (b) $C_4 - C_4$

Figure 5.1: The graphs $2C_4$ and $C_4 - C_4$

Lemma 61. *Let G be a graph from $\mathcal{Y}^*$ and let C^1 and C^2 be two vertex-disjoint induced C_4's in G. Then there are at least two edges between C^1 and C^2 in G.*

Proof. We prove the lemma by induction on the number of vertices in G. Clearly the lemma is true if G contains at most 7 vertices.

Let $A \cup B$ be a bipartition of G. By definition, the vertices of G can be partitioned into two parts $A = A_1 \cup A_2$ and $B = B_1 \cup B_2$ in such a way that $A_1 \cup B_1$ induces a 1-regular graph and $A_2 \cup B_2$ induces a complete bipartite graph.

The vertices of an arbitrary induced C_4 in G can be arranged within the four subsets of G in exactly one of the following ways:

(1) one vertex in A_1, one in B_1, one in A_2 and one in B_2,

(2) two vertices in A_2 and two in B_2,

(3) one vertex in A_1, two in B_2 and one in A_2,

(4) one vertex in B_1, two in A_2 and one in B_2,

(5) two vertices in A_1 and two in B_2,

(6) two vertices in B_1 and two in A_2.

If both C^1 and C^2 are located according to case 5 (or case 6), then the lemma holds by induction. In all other cases it is easy to check the existence of at least two edges between C^1 and C^2 with endpoints in $A_2 \cup B_2$. □

Corollary 62. *Every graph in $\mathcal{Y}^*$ is $(2C_4, C_4 - C_4)$-free.*

Corollary 62 and the lower bound on the number of n-vertex graphs in $\mathcal{Y}^*$ imply the following conclusion.

Theorem 63. *The number of n-vertex $(2C_4, C_4 - C_4)$-free chordal bipartite graphs is $\Omega(2^{\Omega(n \log^2 n)})$, i.e. the class of $(2C_4, C_4 - C_4)$-free chordal bipartite graphs is superfactorial.*

5.7 Conclusion

In the present chapter we proved that the STABLE-Π problem is polynomial-time solvable for all subfactorial hereditary properties Π and for seven of the nine minimal factorial hereditary properties. For $\Pi = \mathcal{M}_1$, the problem is known to be NP-complete. This leaves one final open case, namely where Π is the class of chain graphs $\mathcal{M}_3$. Clarifying the complexity status of this exception is a challenging research problem. In the Supplement, we found a proper subclass of chordal bipartite graphs which is superfactorial, showing that chordal bipartite graphs are not minimal superfactorial. Finding minimal superfactorial classes in this family is an interesting direction for future research.

Chapter 6

Efficient Edge Domination

6.1 Introduction

In this section, we study the problem of determining whether a graph G has an induced matching that dominates every edge of the graph, i.e. whether G has an induced matching M such that every edge of the graph shares at least one vertex with an edge of M. We will refer to this problem as EFFICIENT EDGE DOMINATION. In the notation of Chapter 5, this is the STABLE-$\mathcal{M}_1^S$ problem. It has also appeared in the literature under the name DOMINATING INDUCED MATCHING [Brandstädt and Mosca, 2011b; Cardoso and Lozin, 2009; Korpelainen, 2009]. The EFFICIENT EDGE DOMINATION problem is related to parallel resource allocation problems of parallel processing systems [Livingston and Stout, 1988], encoding theory and network routing [Grinstead et al., 1993].

From an algorithmic point of view, the EFFICIENT EDGE DOMINATION problem, like the MAXIMUM INDUCED MATCHING problem is hard: it is NP-complete for general graphs [Grinstead et al., 1993]. In [Cardoso et al., 2008], it was shown that if a graph has a dominating induced matching, then this induced matching is of maximum size. However, not every maximum induced matching is dominating and there are classes of graphs (e.g. line graphs) where the problems have different complexities [Cardoso and Lozin, 2009; Kobler and Rotics, 2003]. The problem is also related to several other algorithmic graph problems, such as 3-COLOURABILITY. Indeed, a graph has a dominating induced matching only if it is 3-colourable.

The problem has also been studied in many restricted graph classes. Kratochvíl [1994] proved the NP-completeness of this problem in cubic graphs, while Cardoso et al. [2008] extended this result to d-regular graphs for arbitrary $d \geq 3$. The problem was also shown to be NP-complete for bipartite graphs [Lu and Tang, 1998] and planar bipartite graphs [Lu et al., 2002], and this was strengthened to planar bipartite graphs of maximum degree 3 [Brandstädt et al., 2010]. It is also known to be NP-complete in many other classes of graphs [Cardoso et al., 2011].

The NP-completeness results for bounded degree and bipartite graphs were also strengthened in [Cardoso and Lozin, 2009] as follows.

Let $\mathcal{S}_k$ denote the class of $(C_3, \ldots, C_k, H_1, \ldots, H_k)$-free bipartite graphs of vertex degree at most 3, where C_k is a chordless cycle on k vertices and H_k is the graph represented in Figure 6.1. Associate with every graph G a parameter $\kappa(G)$, which is the maximum k such that $G \in \mathcal{S}_k$. If G belongs to no class $\mathcal{S}_k$, we define $\kappa(G)$ to be 0, and if G belongs to all classes $\mathcal{S}_k$, then $\kappa(G)$ is defined to be ∞. Finally, for a set of graphs M, define $\kappa(M) = \sup\{\kappa(G) \ : \ G \in M\}$.

Theorem 64. *[Cardoso and Lozin, 2009] Let M be a set of graphs and X the class of M-free bipartite graphs of vertex degree at most 3. If $\kappa(M) < \infty$, then the* Efficient Edge Domination *problem is NP-complete in the class X.*

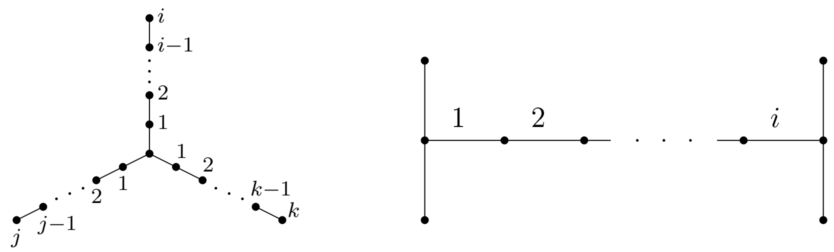


Figure 6.1: Graphs $S_{i,j,k}$ (left) and H_i (right)

Unless $P = NP$, Theorem 64 provides a necessary condition for polynomial-time solvability of the problem in classes of graphs defined by forbidden induced subgraphs. In particular, given a set M of forbidden graphs, the problem is polynomial-time solvable in the class of M-free graphs only if $\kappa(M) = \infty$. Three basic ways to make this happen is to include in the set M of forbidden graphs

(1) graphs containing arbitrarily large minimum induced cycles,

(2) graphs containing arbitrarily large minimum induced subgraphs of the form H_k,

(3) a graph G with $\kappa(G) = \infty$.

Nearly all polynomial-time results available in the literature deal with graph classes of the first type. This includes bipartite permutation graphs [Lu and Tang, 1998], convex graphs [Korpelainen, 2009], chordal graphs [Lu et al., 2002] and hole-free graphs [Brandstädt et al., 2010]. The problem is also known to be solvable in polynomial time in $(H_k, H_{k+1}, \ldots)$-free graphs of bounded degree, which is a class of type 2 and in classes of bounded clique-width [Cardoso et al., 2011] (which can be any of the three types). Nothing else is known about the complexity of the problem in classes of the second type and only a few results are available for classes of the third type. For example, the problem is known to be solvable in polynomial time in *claw*-free graphs [Cardoso et al., 2011] and in P_7-free graphs [Brandstädt and Mosca, 2011b] (see [Brandstädt and Mosca, 2011a] for full details of the proof). By definition, $\kappa(G) = \infty$ if and only if G belongs to all classes $\mathcal{S}_k$, i.e. G belongs to the intersection $\cap\mathcal{S}_k$ taken over all possible values of k. It is not difficult to see that $G \in \cap\mathcal{S}_k$ if and only if every connected component of G is of the form $S_{i,j,k}$ (see Figure 6.1), where $i, j, k \geq 0$. The smallest non-trivial graph of the from $S_{i,j,k}$ (apart from the chordless paths) is $S_{1,1,1}$, also known as the claw. It is known that the problem is solvable in polynomial time in this class [Cardoso and Lozin, 2009]. This class has received much attention in the literature due to the many attractive properties of claw-free graphs, see for example [Chudnovsky and Seymour, 2008; Favaron, 2003; Le et al., 2008; Minty, 1980]. In particular, Minty [1980] develops a polynomial-time algorithm for the Maximum Independent Set problem in claw-free graphs, which extends the celebrated Edmonds solution for the Maximum Matching problem [Edmonds, 1965]. However, very little is known about the complexity of algorithmic problems in extensions of claw-free graphs. For instance, it is known that the Maximum Independent Set problem can be solved in polynomial time for $S_{1,1,2}$-free graphs [Lozin and Milanič, 2008], but for the class of $S_{1,1,3}$-free graphs, the complexity status of the problem was an open question. It has been shown that Efficient Edge Domination is solvable in polynomial time for $S_{1,2,2}$-free graphs [Korpelainen, 2012] (these are also known as E-free graphs, since $S_{1,2,2}$ can also be drawn as a capital letter E).

To the best of our knowledge, nothing is known about the parameterized complexity of the Efficient Edge Domination problem.

In the present chapter, we first prove that Efficient Edge Domination is fixed-parameter tractable with respect to two natural parameters, namely the size of the induced matching ($|B'|$) and the size of the independent set ($|W'|$). We then go on to show that the problem is polynomial-time solvable in the class of $S_{1,1,3}$-free graphs. Generalising this result and using previously known results, we completely characterise the complexity of the problem in F-free graphs, for graphs F on at most 6 vertices.

6.2 Preliminaries

We can phrase Efficient Edge Domination as the problem of deciding whether or not the vertices of a graph G can be partitioned into two sets W' and B', such that W' is an independent set in G and $G[B']$ is an induced matching of G (i.e. every vertex in $G[B']$ is of degree exactly 1). We say that the vertices in W' are *white* and the vertices in B' are *black*.

We consider a generalisation of this problem, namely the Minimum Restricted Efficient Edge Domination problem. This is defined as follows:

Minimum Restricted Efficient Edge Domination

Instance: A graph G, a weight function $w : E(G) \to \mathbb{R}$ and two sets $B, W \subseteq V(G)$.

Output: Among all dominating induced matchings B', W' satisfying $B \subseteq B'$, $W \subseteq W'$, output one that minimises $\sum_{e \in M} w(e)$ where $M = \{xy \in E(G) : x, y \in B'\}$. If no such partition exists then output Impossible.

The unrestricted version of this problem i.e. that where $B = W = \emptyset$ is known as the Minimum Efficient Edge Domination problem. The Efficient Edge Domination problem is the case where $B = W = \emptyset$ and the weight function w is identically zero.

We say that a black vertex is *matched* if it has exactly 1 black neighbour and *unmatched* if it has no black neighbour. We say that a colouring of the vertices of a graph is *valid* if no vertex is coloured with two different colours, no two white vertices are adjacent and no black vertex has more than one black neighbour.

6.3 Simple Reduction Rules

Given an instance (G, B, W) of the Minimum Efficient Edge Domination problem, we define the following potential function: $\varphi(G, B, W) = 2|V(G)| - |B| - |W|$. Note that $\varphi(G, B, W) \geq 0$, with equality if and only if $|V(G)| = |B| = |W|$. The proofs in this chapter rely on a series of reduction rules. Applying these rules allows us to assume certain useful properties about the input instance. In this section we introduce some simple reduction rules that can be applied to general instances of the problem.

We apply the following reduction rules in the order given, i.e. before the application of any rule, we assume that all previous rules have been applied exhaustively. We will show that after any of these reduction rules is applied, the reduced problem will have a valid solution if and only if the original one did. Moreover, the value of optimal solution of the reduced problem will be the same as that for the original problem, unless otherwise stated (see comment after Rule R4). Each of these reduction rules runs in polynomial time. To see that this yields an algorithm that runs in polynomial time overall, note that each rule either solves the problem optimally or decreases the value of φ (except Rule R1, which recurses on disjoint sets of vertices).

Note that if the problem instance (G, B, W) has no valid solution for some component of G, then the problem has no valid solution for G. If every component of G has a valid solution, an optimal solution for G can be obtained by taking the union of the optimum solutions for the components of G.

Given an instance (G, B, W) of the problem with $\varphi(G, B, W) > 0$:

R1 *If G is disconnected, solve the problem componentwise.* If the problem has no solution on any of the components of G, it has no solution on G itself. Note that if $D_1, \ldots, D_k$ are the components of G then $\varphi(G, B, W) = \varphi(D_1, V(D_1) \cap B, V(D_1) \cap W) + \cdots + \varphi(D_k, V(D_k) \cap B, V(D_k) \cap W)$ and the weight of the optimal solution is the sum of the weights of the optimal solutions on each of its components.

R2 *If $B \cap W \neq \emptyset$ then output $(G, V(G), V(G))$.* Indeed, if $B \cap W \neq \emptyset$ then the problem has no valid solution, since no vertex can be both black and white. In this case we can output the instance $(G, V(G), V(G))$, which clearly has no valid solution and which φ evaluates to 0.

R3 *If $x \in W$, output $(G - x, B \cup N(x), W \setminus \{x\})$.* Indeed, all neighbours of a white vertex must be black. Once we force all the neighbours of a white vertex to be coloured black, we can remove the white vertex without loss of generality.

Note that Rule R3 allows us to assume that from now on, $W = \emptyset$ in the input instance.

R4 *If $x, y \in B, xy \in E(G)$, output $(G - \{x, y\}, B \setminus \{x, y\}, W \cup N(\{x, y\}))$.* Indeed, if xy is an edge where both x and y are black, then in any valid solution, xy must form part of the matching, and so any other neighbours of x and y must be white.

Rule R4 allows us to assume that from now on, B is an independent set in the input instance. Note that this rule reduces the value of the optimal solution in G by $w(xy)$.

R5 *If $N(x) = 0$, output $(G, B, W \cup \{x\})$.* In this case, x cannot be matched with any black vertex, so it must be white.

R6 *If* $V|(G)| = 2$, *output* $(G, V(G), W)$. In this case, G must consist of a single edge, so both its vertices must be black.

This allows us to assume that G is connected and contains at least 3 vertices.

R7 *If* $x \in B, N(x) = 1$, *output* $(G, B \cup N(x), W)$. In this case, x must be matched with its unique uncoloured neighbour.

R8 *If* $x, y \in B$, $z \in V(G)$, $xz, yz \in E(G)$, *Output* $(G, B, W \cup \{z\})$. If z has two black neighbours, it cannot be coloured black.

R9 *If G contains a diamond, colour it in the only possible way.* If $w, x, y, z \in V(G)$, with $wx, wy, wz, xz, yz \in E(G)$ and $xy \notin E(G)$, then in any valid solution w and z must be black and x and y must be white (see also Figure 6.3). We therefore output $(G, B \cup \{w, z\}, W \cup \{x, y\})$.

R10 *If* $x_1x_2, x_2x_3, x_3x_4, x_4x_1 \in E(G)$, $x_1x_3, x_2x_4 \notin G$, *and* $x_1 \in B$ *output* $(G, B \cup \{x_3\}, W \cup \{x_2, x_4\})$. The vertices of an induced C_4 must alternate in colour (see also Figure 6.2).

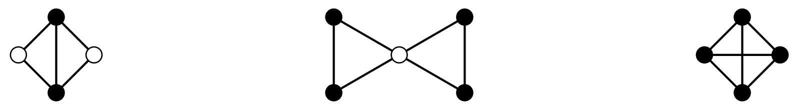
Figure 6.2: A C_4 subgraph has two possible colourings

R11 *If G contains a K_4 output* $(G, V(G), V(G))$ In this case, no valid colouring is possible (see also Figure 6.3).

R12 *If G contains a butterfly, colour it in the only way possible.* If $x, y_1, y_2, y_3, y_4 \in V(G)$ with $xy_1, xy_2, xy_3, xy_4, y_1y_2, y_3y_4 \in E(G)$ and $y_1y_3, y_1y_4, y_2y_3, y_2y_4 \notin E(G)$ then in any solution, x must be white and y_1, y_2, y_3, y_4 must be black (see also 6.3). We therefore output $(G, B \cup \{y_1, y_2, y_3, y_4\}, W \cup \{x\})$.

R13 *If G contains a paw, the leaf vertex and the central vertex must have different colours.* If $w, x, y, z \in V(G)$ with $wx, xy, xz, yz \in E(G)$, $wy, wz \notin E(G)$, then in any solution, w and x must have different colours (see also 6.4). Therefore if $x \in B$ then output $(G, B, W \cup \{w\})$ and if $w \in B$ then output $(G, B, W \cup \{x\})$.

We may now assume that G is $(butterfly, diamond, K_4)$-free. Armed with these reduction rules, we are now ready to study the parameterized complexity of the problem.



(a) diamond (b) butterfly (c) K_4

Figure 6.3: A diamond or butterfly subgraph has only one possible valid colouring. A K_4 subgraph has no valid colourings

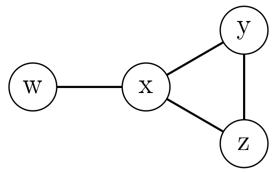


Figure 6.4: In a paw subgraph, w and x must have different colours.

6.4 Efficient Edge Domination is Fixed-Parameter Tractable

In this section, we parameterize the unweighted version of the problem by the size of both $|B|$ and $|W|$. We define the parameterized (k, l)-Restricted Efficient Edge Domination problem as follows:

(k, ℓ)-Restricted Efficient Edge Domination

Instance:	A graph G, and two sets $B, W \subseteq V(G)$.
Parameter:	A pair of integers k,l
Output:	Output a dominating induced matching B', W' satisfying $B \subseteq B'$, $W \subseteq W'$, such that $|B'| \leq k$ OR $|W'| \leq \ell$. If no such partition exists then output Impossible.

Note that if a graph G has a dominating induced matching then its size will be equal to that of a maximum induced matching in G. However, while the k-Induced Matching problem is W[1]-hard, the related $(k, 0)$-Efficient Edge Domination is fixed-parameter tractable.

Theorem 65. *The* (k, ℓ)-Restricted Efficient Edge Domination *problem is fixed-parameter tractable.*

Proof. To prove the theorem, we use the following brute-force algorithm that solves the problem.

Given an instance (G, B, W) of the problem (we define the weight function w to be identically zero), we run the following algorithm:

1. Apply the reduction rules. Note that when applying Rule R4, k must be reduced by 2 and when applying Rule R3, ℓ must be reduced by 1.

2. If $|B| > k$ AND $|W| > \ell$, stop this branch of the algorithm.

3. If (B, W) forms a partition of $V(G)$, output (B, W) and stop the algorithm.

4. As long as the reduced graph G is not fully coloured, it must contain a component of size at least 3. We can therefore find a P_3 or a K_3 in the reduced graph. None of these 3 vertices may be pre-coloured white and at most one of them may be pre-coloured black. This means that there are at most 3 ways to colour the 3 vertices in a valid way (one of them must be white, the other two black). Branch according to these possibilities and run the algorithm recursively.

If the algorithm fails to find a suitable partition, we output IMPOSSIBLE.

Since we add at least one black and at least one white vertex at each branching point, the complexity of the algorithm is bounded by $p(n)3^{\max\{k,\ell\}}$, where $p(n)$ is a polynomial in the number of vertices of the graph, independent of k and ℓ. □

6.5 Main Result

In this section, we prove the following theorem.

Theorem 66. *The* MINIMUM RESTRICTED EFFICIENT EDGE DOMINATION *problem can be solved in polynomial time in the class of $S_{1,1,3}$-free graphs (see Figure 6.5).*

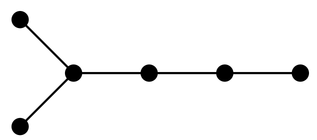


Figure 6.5: The graph $S_{1,1,3}$

As a starting point, we have the following theorem.

Theorem 67. *[Brandstädt et al., 2010] The* MINIMUM EFFICIENT EDGE DOMINATION *problem can be solved in polynomial time in the class of $(C_5, C_6, \ldots)$-free graphs.*

An easy consequence of this is the following:

Corollary 68. *The* Minimum Restricted Efficient Edge Domination *problem can be solved in polynomial time in the class of* $(C_5, C_6, \ldots)$*-free graphs.*

Proof. Let (G, B, W) be an instance of the Minimum Restricted Efficient Edge Domination problem such that G is $(C_5, C_6, \ldots)$-free. For every vertex $x \in B$, create a butterfly and attach (with an edge) x to the central vertex of the butterfly. Likewise, for every vertex $x \in W$, create a butterfly and attach one of the non-central vertices of the butterfly to x. Set the weight of all of the resulting new edges to zero. The resulting graph G', together with it's resulting weight function gives an instance of the Minimum Efficient Edge Domination problem. In any solution to this problem, since the butterfly graph can only be coloured in one valid way (see Figure 6.3), every vertex $x \in B$ must be coloured black and every vertex $x \in W$ must be coloured white. Note that G' admits a solution to the Efficient Edge Domination problem if and only if G does and, furthermore, the minimal such solution in G' has the same weight as the corresponding solution in G. □

We say that an instance (G, B, W) is *basic* if it is $(C_5, C_6, \ldots)$-free.

From now on, we may assume that the instance we are working with is non-basic and that any point of the proof, all of the reduction rules appearing earlier in the chapter have been applied exhaustively. We will construct some more reduction rules whose application will reduce the problem to a basic instance.

Recall that a set S of vertices dominates the graph G if every vertex in G is either in S or adjacent to a vertex in S. Whenever we apply one of the reduction rules defined earlier, if we remove a vertex from the graph, we colour all of it's neighbours. Thus if at any stage we find that $W \cup B$ dominates the graph G, then after applying the reduction rules exhaustively, this will still be the case. This leads us to the following useful result.

Lemma 69. *If G is an $S_{1,1,3}$-free graph and $W \cup B$ dominates G, then the problem can be solved in polynomial time.*

Proof. Suppose G is such a graph and $W \cup B$ dominates G. Without loss of generality, we may assume that the reduction rules above have been applied exhaustively. In this case G must be connected, $W = \emptyset$ and B is an independent set of vertices which dominates the graph. Note that every vertex not in B must have exactly one neighbour in B (by Rule R8 and the fact that B dominates the graph). Also, by Rules R5 and R7, every vertex in B must have at least two neighbours (which are not in B, since B is an independent set).

Consider a vertex $x \in B$. No vertex in B can be adjacent to x. Since G is $(butterfly, diamond, K_4)$-free, the neighbourhood $N(x)$ induces a graph $G[N(x)]$ containing at most one edge. Suppose that $G[N(x)]$ contains an edge. Then by Rule R13, $N(x)$ contains at most two vertices.

Now suppose that $x \in B$, $|N(x)| \geq 3$. Then $V(G) = N(x) \cup \{x\}$. Indeed, suppose that this is not the case. Then $G[N(x)]$ must be edge-less. Let y_1, y_2, y_3 be distinct neighbours of x. Some vertex $z \notin N(x)$ must be adjacent to a vertex in the neighbourhood of x. Without loss of generality, assume that z is adjacent to y_1. By Rule R8, $z \notin B$. Let $z' \in B$ be the black neighbour of z. Again, by Rule R8, z' is not a neighbour of y_1, y_2, y_3. By Rule R10, z is not a neighbour of y_2 or y_3. But now $G[x, y_2, y_3, y_1, z, z']$ is an $S_{1,1,3}$. This contradiction proves that if $|N(x)| \geq 3$ then G is a star, whose central vertex is black. In this case the problem is easy to solve optimally.

We may now assume that every vertex in B has exactly two neighbours. In any valid colouring, exactly one of these two neighbours must be black and one must be white. If $x, y \notin B$ are adjacent then they cannot both be white and they cannot both be black. The effect of this is that if we specify the colour of any vertex $\notin B$, we can immediately deduce the only possible valid colour of every other vertex in G. Thus in this case we pick an arbitrary uncoloured vertex and choose the optimum of the two resulting colourings. $\square$

Since our instance is non-basic, it must contain an induced cycle C of length at least five. Let $v_1, \ldots, v_k$ be the vertices of C, in order. We say that a vertex $x \in V(G)$ which is not in the cycle is of type d (with respect to C) if it has d neighbours on C.

We analyse this situation via a series of claims, that must be satisfied for every vertex $x \in V(G) \setminus V(C)$.

C1 *Vertex x can have at most one pair of consecutive neighbours on the cycle C.* This follows immediately from the fact that G is $(diamond, butterfly, K_4)$-free.

C2 *If $k \geq 6$ then there are no vertices of type 1.* Indeed, suppose x is of type 1, and is adjacent to v_2 (by symmetry). Then $G[v_2, x, v_1, v_3, v_4, v_5]$ is an $S_{1,1,3}$, which is a contradiction.

C3 *If $k \geq 7$ and x is a vertex of type 2, then x must be adjacent to consecutive vertices of C.* Suppose $k \geq 7$ and x is a vertex of type 2, but x is adjacent to non-consecutive vertices of C. By symmetry, we assume x is adjacent to v_1 and v_i, where $3 \leq i \leq \frac{k}{2}+1$. If $i = 3$, then $G[v_3, v_2, x, v_4, v_5, v_6]$ is an $S_{1,1,3}$. If $i = 4$ then $G[v_1, v_2, v_k, x, v_4, v_5]$ is an $S_{1,1,3}$. If $i \geq 5$ then $G[v_1, x, v_k, v_2, v_3, v_4]$ is an $S_{1,1,3}$. This contradiction proves the claim.

C4 *If $k \geq 7$, x is of type ≥ 3 then x has a pair of consecutive neighbours on C.* Suppose x is not adjacent to any pair of consecutive vertices on C. Let v_1, v_i, v_j be distinct neighbours of C with $i < j$. If $i > 3$ then $G[v_1, v_2, v_k, x, v_i, v_{i+1}]$ is an $S_{1,1,3}$. Thus $i = 3$. Repeating this argument, x must also be adjacent to v_5. But then $G[v_1, v_2, v_k, x, v_5, v_4]$ is an $S_{1,1,3}$, which is a contradiction.

C5 *If $k \geq 8$, then no vertex is type ≥ 3 with respect to C.* Suppose that $k \geq 8$ and x is a vertex of type ≥ 3. By Claim C4, x must have a pair consecutive neighbours in the cycle. By symmetry, we may assume x is adjacent to v_1, v_2, v_i, where $6 \leq i \leq k-1$. But then $G[v_i, v_{i-1}, v_{i+1}, x, v_2, v_3]$ is an induced $S_{1,1,3}$, which is a contradiction.

C6 *If $k = 7$ and x is of type ≥ 3 then x is of type 3, x must have two consecutive neighbours on C and in any dominating induced matching, x's third neighbour must be coloured white.* Suppose x is of type ≥ 4. By symmetry and Claim C1, we may assume that x is adjacent to v_1, v_2, v_4, v_6, but not v_3, v_5, v_7. But then $G[v_4, v_3, v_5, x, v_1, v_7]$ is an $S_{1,1,3}$. Thus x must have type 3. By Claim C4, x must have two consecutive neighbours on the cycle. By the same argument as for Claim C5, if x is adjacent to v_1, v_2, v_i, then $i = 5$. Suppose v_5 were coloured black. Because $G[x, v_1, v_2, v_5]$ is a paw, x would have to be white. Then v_1 and v_2 would be black, so v_3 and v_7 would be white, so v_4 and v_6 would be black. This is a contradiction since then v_5 would be black and therefore could not have 2 black neighbours. Thus v_5 must be white.

We thus get the following reduction rule:

R14 *If G contains a cycle of length seven, with a neighbour of type 3, output $(G, B, W \cup \{y\})$, where y is x's non-consecutive neighbour on the cycle.*

6.5.1 Graphs containing a cycle of length ≥ 7

We now consider the case where G contains a cycle C of length $k \geq 7$. Throughout this section, we interpret the subscripts of the vertices v_i modulo k.

Because of Rules R3 and R14 and Claims C2 and C3, from now on, we may assume that if $x \in V(G) \setminus V(C)$ has a neighbour on C then x has exactly two neighbours on C and these neighbours must be consecutive vertices of C. Note that by $(butterfly, diamond, K_4)$-freeness, it follows that any two vertices outside C cannot have a common neighbour on C. This means that the vertices with neighbours in C can be uniquely determined by which vertices in C they have as neighbours.

Now suppose that x and y (not necessarily distinct) have neighbours on C and that x and y are "consecutive neighbours" of C, i.e. that for some i, j, where $i < j$, x is adjacent to v_i and v_{i+1}, y is adjacent to v_j and v_{j+1}, but the vertices $v_{i+2}, \ldots, v_{j-1}$ have no neighbours outside C (interpreting subscripts modulo k if necessary). Note that $G[v_i, v_{i+1}, x, v_{i+2}]$ is a paw, so in any dominating induced matching, v_{i+1} and v_{i+2} must be different colours (similarly for v_{j-1} and v_j). Since G is $(butterfly, diamond, K_4)$-free, $j \geq i + 2$. We now make the following useful observation.

Observation 1. *Suppose that $x_1x_2x_3x_4$ is an induced path of length three in G, such that x_2 and x_3 have no neighbours in G outside this path. Then in any dominating induced matching, x_1 and x_4 must have the same colour.*

First, let us suppose that $j - i \equiv 0 \mod 3$. If v_{i+2} is black, then in any dominating induced matching, v_{i+1} must be white and v_{i+3} must be black. Applying Observation 1 repeatedly, we find that v_{j-1} and v_j must also be black, which is a contradiction. Thus v_{i+2} must be white.

Next, suppose $j - i \equiv 1 \mod 3$. If v_{i+1} is black, then v_{i+2} be white, so v_{i+3} must be black. Again, applying Observation 1 repeatedly, we find that both v_{j-1} and v_j must also then be black, which is a contradiction. Thus v_{i+1} must be white.

We thus have the following rule:

R15 If there is such a cycle C and if such vertices x, y exist, where $j - i \neq 2 \mod 3$, output $(G, B, W \cup \{v_{i+2}\})$ if $j - i \equiv 0 \mod 3$ and output $(G, B, W \cup \{v_{i+1}\})$ if $j - i \equiv 1 \mod 3$.

Now suppose that $j - i \equiv 2 \mod 3$ for all such consecutive pairs x, y. Let $x_1, \ldots, x_t$ be the neighbours of C, in order and define $x_{t+1} := x_1$. Let v_{i_i} and v_{i_i+1} be the neighbours of x_i (i_i's chosen such that $i_i < i_{i+1} <= i_i + k$).

For any two "consecutive neighbours" x_i, x_{i+1} of the cycle C, x_i is adjacent to v_{i_i} and v_{i_i+1} and x_{i+1} is adjacent to $v_{i_{i+1}}$ and $v_{i_{i+1}+1}$, while $v_{i_i+2}, \ldots, v_{i_{i+1}-1}$ have no neighbours outside the cycle C. We are thus reduced to the case where the size of the set $\{v_{i_i+2}, \ldots, v_{i_{i+1}-1}\}$ is a multiple of 3 for every such pair of "consecutive neighbours" x_i and x_{i+1}.

Note that for any i, $G[x_i, v_{i_i}, v_{i_i+1}]$ is a triangle, so at least one of v_{i_i}, v_{i_i+1} must be black. Suppose v_{i_i+1} is black. Then v_{i_i+2} must be white. If $i_{i+1} = i_i + 2$, then $v_{i_{i+1}+1}$ must be black. Otherwise, v_{i_i+3} and v_{i_i+4} must be black. Again, using Observation 1 repeatedly, we find that $v_{i_{i+1}}$ must be white and thus $v_{i_{i+1}+1}$ must be black. Continuing this argument, we find that v_{i_j} is white and v_{i_j+1} is black for all j. Similarly, if v_{i_i+1} is white, then v_{i_i} must be black, so by a symmetric argument we find that in this case v_{i_j} is black and v_{i_j+1} is white for all values of j.

We may thus conclude that $N(C)$ is black and there are at most two possible ways of colouring the cycle.

R16 If C has neighbours, and it is possible to colour C, colour it in the best way possible (of the at most two choices) and add $N(C)$ to B. If no such colouring is possible, output $(G, V(G), V(G))$.

If C has no neighbours outside C, then by Observation 1, the colours along the cycle must be a repeating sequence of *white*, *black*, *black*. Thus there are at most 3 possible colouring of C. We thus get the following reduction rule:

R17 If C has a valid colouring, output an optimal one (of the at most three choices). If C has no valid colouring output $(G, V(G), V(G))$.

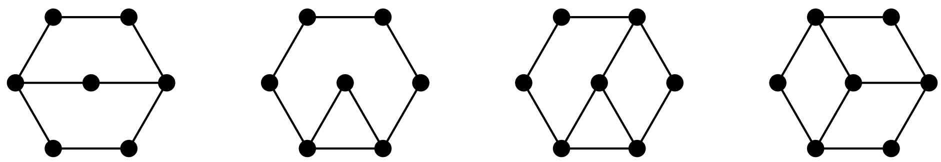
From now on, we may assume G has no induced cycles of length ≥ 7.

6.5.2 Graphs containing a cycle of length 6

We assume that all induced cycles in G are of length at most 6. Suppose C is an induced cycle on 6 vertices $v_1, \ldots, v_6$.

If x is a neighbour of the cycle, we know it cannot be of type 1 (by Claim C2). By Claim C1, it cannot be of type ≥ 4. If x is of type 3 and has two consecutive neighbours in the cycle, then it's third neighbour must be one of the two vertices not adjacent to the first two. We say that such type 3 vertices are of type 3a. If x does not have 2 consecutive neighbours in the cycle, then it must be adjacent to v_1, v_3, v_5 or v_2, v_4, v_6. We say that such vertices are of type 3b. If x is of type 2, and is adjacent to two vertices of the cycle which are of distance 2 from each other in the cycle, say x were adjacent to v_1 and v_3, then $G[v_1, x, v_2, v_6, v_5, v_4]$, would be a $S_{1,1,3}$, which would be a contradiction. Thus if x is of type 2, it is adjacent to either two opposite or two consecutive vertices of the cycle. We say such a vertex x is of type 2a and 2b respectively (see also Figure 6.6).

Suppose there is a vertex x of type 3a. Without loss of generality, x is adjacent to v_1, v_2 and v_4. If x is black then, because $G[x, v_1, v_2, v_4]$ is a paw, v_4 must be white, so v_3 and v_5 must be black, so v_2 must be white, so v_1 must be black, so v_6 must be white. If x is white, then v_1, v_2 must be black, so v_6 must be white. In both cases v_6 is white, thus in any valid partition v_6 must be white.



(a) Type 2a (b) Type 2b (c) Type 3a (d) Type 3b

Figure 6.6: The possible neighbours of a C_6 in an $S_{1,1,3}$-free graph

R18 If $C = v_1 - v_2 - v_3 - v_4 - v_5 - v_6 - v_1$ is a cycle of length 6, and x is adjacent to 3 vertices on C, two of which are adjacent, say v_1, v_2, v_4, output $(G, B, W \cup \{v_6\})$.

We may now assume there are no vertices of type 3a.

Suppose, for contradiction, that there is a vertex x of type 3b and a vertex y of type 2b. Without loss of generality, we may assume x is adjacent to v_1, v_3, v_5 and y is adjacent to v_1 and v_2. But then x must be adjacent to y otherwise $G[v_1, v_6, y, x, v_3, v_4]$ would be an $S_{1,1,3}$. But then $G[v_1, y, v_2, x]$ would be a diamond. This contradiction implies that if there is a vertex of type 3b, the only neighbours to the cycle not of type 3b must be of type 2a.

Suppose there is a vertex x of type 3b, adjacent to v_1, v_3 and v_5, say. Then $G[v_1, v_2, v_3, x]$, $G[v_3, v_4, v_5, x]$ and $G[v_5, v_6, v_1, x]$ are C_4's, so the only valid colourings are v_1, v_3, v_5 white, v_2, v_4, v_6, x black or vice-versa. This means that any type 2a neighbours must be black and there can be no two such neighbours with the same neighbourhood in the cycle. In fact, because all the black vertices in the cycle need to have a black neighbour, there must be exactly one of each of the three sorts of type 2a neighbours and they must be pairwise nonadjacent.

R19 Suppose $C = v_1 - v_2 - v_3 - v_4 - v_5 - v_6 - v_1$ is a cycle in G of length 6, and x is adjacent to 3 vertices on C, none of which are adjacent, say v_1, v_3, v_5. Check that there are exactly three vertices y_1, y_2, y_3 such that y_i's neighbours on the cycle are x_i and x_{i+3}, otherwise output $(G, V(G), V(G))$.

Any type 2a vertices must be black and must be matched to a vertex on the cycle. Thus any other neighbour (i.e. a neighbour of type 3b or of type 0) of a type 2a vertex must be white.

R20 Suppose C is a cycle in G of length 6, if there is a type 3b or type 0 vertex x adjacent to a type 2a vertex y, output $(G, B, W \cup \{x\})$.

Suppose there are type 0 vertices z, z' such that x and z' are adjacent to z. Then x must be adjacent to z' otherwise $G[v_1, v_2, v_6, x, z, z']$ would be an $S_{1,1,3}$. Thus every type 0 vertex must be adjacent to a vertex of type $3b$. In other words, the graph is dominated by the cycle, the type 3b vertices and all three type 2a vertices. However, there are at most 2 ways of colouring this set of vertices (depending on the colour of vertex v_1). Thus in this case, the problem can be solved in polynomial time by the use of Lemma 69.

R21 Suppose C is a cycle in G of length 6, if there is a type 3b vertex x. Try both possible colourings for C and apply Lemma 69. If a solution exists, output the optimal one, otherwise output $(G, V(G), V(G))$.

So we have now reduced to the case where there are no vertices of type 3. Suppose there is a type 2a vertex y and a type 2b vertex x with a common neighbour in the cycle. Without loss of generality, we may assume x is adjacent to v_1 and v_2 and y is adjacent to v_1 and v_4. The vertices x and y cannot be adjacent, otherwise $G[v_1, x, v_2, y]$ would be a diamond. But then $G[v_1, v_6, x, y, v_4, v_3]$, would be an $S_{1,1,3}$. Thus no type 2a vertex can have a common neighbour in the cycle with a type 2b vertex.

No two type 2b vertices may have a common neighbour in the cycle (since G is $(diamond, butterfly, K_4)$-free). If there are 3 type 2b vertices, without loss of generality a_1 adjacent to v_1 and v_2, a_2 adjacent to v_3 and v_4 and a_3 adjacent to v_5 and v_6, then there can be no type 2a vertices. Also, if a_1 is white, then v_1 and v_2 must be black, so v_3 and v_6 must be white, so a_2, a_3, v_4, v_5 must all be black, which is a contradiction. Thus in this case all the a_i must be black and the cycle must be coloured in alternating colours.

R22 Suppose C is a cycle in G of length 6, if there are 3 vertices a_1, a_2, a_3 of type 2b adjacent to v_1&v_2, v_3&v_4 and v_5&v_6 respectively, check if $(G, B \cup \{a_1, a_2, a_3, v_1, v_3, v_5\}, W \cup \{v_2, v_4, v_6\})$ or $(G, B \cup \{a_1, a_2, a_3, v_2, v_4, v_6\}, W \cup \{v_1, v_3, v_5\})$ violates Rule R2. If neither of them does, pick the one that had the better weight. If one of them violates the rule, output the other. If both violate the rule, output $(G, V(G), V(G))$.

If there are 2 type 2b vertices which are not opposite, without loss of generality a_1 adjacent to v_1 and v_2, a_2 adjacent to v_3 and v_4, then because $G[v_2, a_1, v_1, v_3]$ is a paw, v_2 and v_3 must have different colours. Again, in this case we cannot have any type 2a vertices. Suppose without loss of generality, that v_2 is white. Then a_1, v_1 and v_3 must be black, so v_6 is white, so v_4 and v_5 are black, which is a contradiction, since $G[a_2, v_3, v_4, v_5]$ is a paw. Thus in this case there is no valid colouring.

R23 If C is a cycle in G of length 6 and has two type 2b vertices that are not opposite, output $(G, V(G), V(G))$.

Now suppose there are 2 type 2b vertices which are opposite, without loss of generality a_1 adjacent to v_1 and v_2, a_2 adjacent to v_4 and v_5. Note that in this case we may (or may not) have some type 2a vertices adjacent to v_3 and v_6. If v_3 is black then since $G[v_2, a_1, v_1, v_3]$ is a paw, v_2 must be white. Similarly if v_6 is black then v_1 must be white. v_1 and v_2 cannot both be white, so at least one of v_3 and v_6 must be white.

If there are no vertices of type 2a, there is only one way to colour the cycle.

R24 If C is a cycle in G of length 6 and has two type 2b vertices a_1, a_2 that are opposite and are adjacent to $v_1 \& v_2$ and $v_4 \& v_5$ respectively, but C has no type 2a neighbours, output $(G, B \cup \{v_1, v_2, v_3, v_4\}, W \cup \{v_3, v_6, a_1, a_2\})$.

If there is a vertex x of type 2a, it is adjacent to v_3 and v_6, at least one of which must be white, so x must be coloured black. If x is adjacent to a_1, this uniquely determines the colouring of C.

R25 If C is a cycle in G of length 6 and has two type 2b vertices a_1, a_2 that are opposite and are adjacent to $v_1 \& v_2$ and $v_4 \& v_5$ respectively and x is a type 2a vertex adjacent to a_1, then output $(G, B \cup \{v_1, v_2, v_3, v_4\}, W \cup \{v_3, v_6, a_1, a_2\})$.

If there are two opposite type 2b vertices, there must be a type 2a vertex and no vertex of type 2a may be adjacent to to a vertex of type 2b. We now show that the cycle, together with the type 2 vertices dominates the graph. (Note that there are at most 3 ways to colour this set of vertices, depending on the colours of v_3 and v_6.) Let x be a type 2a vertex, a_1 be a type 2b vertex adjacent to v_1 and v_2 and suppose a_1 has a type 0 neighbour y. Then y must be adjacent to x otherwise $G[v_6, x, v_5, v_1, a_1, y]$ would be an $S_{1,1,3}$. Now suppose that z is a type 0 vertex adjacent to x and z' is a type 0 vertex adjacent to z, but not x. Then $G[v_6, v_1, v_5, x, z, z']$ would be an $S_{1,1,3}$. This contradiction shows that the cycle, together with the type 2 vertices dominates the graph.

R26 If C is a cycle in G of length 6 and has two type 2b vertices a_1, a_2 that are opposite and are adjacent to $v_1 \& v_2$ and $v_4 \& v_5$, the cycle and the type 2 vertices dominate the graph G. Try all 3 of the possible colourings for the cycle and vertices of type 2 and apply Lemma 69.

Now suppose there is exactly 1 type 2b vertex, without loss of generality a_1 adjacent to v_1 and v_2. Note that in this case we can again have type 2a vertices, but as before, only if they are adjacent to v_3 and v_6. If v_1 is white, then a_1, v_2 and v_6 must all be black, so v_3 must be white, so v_4 must be black, so v_5 must be black (since v_4 has no neighbours outside the cycle). This is a contradiction, since v_5 cannot be black if it has 2 black neighbours. Thus our original assumption must have been wrong and v_1 must be coloured black. Similarly, v_2 must be black. This yields a unique colouring of the cycle.

R27 If C is a cycle in G of length 6 and has exactly one type 2b vertex a_1, adjacent to $v_1 \& v_2$, say, output $(G, B \cup \{v_1, v_2, v_4, v_5\}, W \cup \{a_1, v_3, v_6\})$.

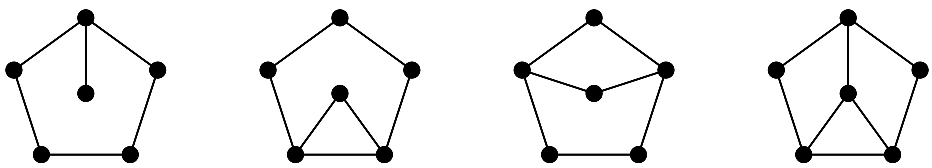
So we are now left with the case where there are only type 2a vertices. We claim that the cycle together with the type 2a vertices dominates the rest of the graph. Indeed, suppose not. Then there must be a type 2a vertex x and type 0 vertices z, z' such that x, z, z' form a path of length 3. Without loss of generality, assume that x is adjacent to v_1 and v_4. Then $G[v_1, v_2, v_6, x, z, z']$ is an $S_{1,1,3}$, which is a contradiction. Thus the cycle, together with the vertices of type 2a does indeed dominate the whole graph. Now we need to look at how many ways we can colour this set. The cycle can be coloured in 5 ways: alternately black/white (2 ways) or black, black, white, black, black, white (3 ways). Each of these gives at most one valid way of colouring all the type 2a vertices.

R28 If C is a cycle in G of length 6 and all neighbours of the cycle are of type 2a, then the cycle together with the type 2a vertices dominates the graph. Try all 5 of the possible colourings for the cycle and vertices of type 2a and apply Lemma 69.

After applying the above reduction rules, we end up with a graph which contains no induced cycles of length ≥ 6.

6.5.3 Graphs containing a cycle of length 5

We now consider the case where the graph contains a cycle C of length 5.



(a) Type 1 (b) Type 2a (c) Type 2b (d) Type 3

Figure 6.7: The possible neighbours of a C_5 in an $S_{1,1,3}$-free graph

In this case we can have vertices of type 1,2 or 3. Suppose x is a type 3 vertex. Without loss of generality, we may assume it is adjacent to v_1, v_2 and v_4. Suppose x is black. Then, since $G[x, v_1, v_2, v_4]$ is a paw, v_4 must be white. So v_3 must be black. Then, since $G[v_2, v_1, x, v_3]$ is a paw, v_2 must be white. Similarly, v_1 must be white, which is a contradiction, since no two white vertices can be adjacent. Thus x must be white.

R29 If C is a cycle in G of length 5, with a type 3 neighbour x, output $(G, B, W \cup \{x\})$.

We may assume there are no vertices of type 3.

Let us say that vertices of type 2 are of type 2a if they have consecutive neighbours on the cycle and of type 2b otherwise. Suppose there are 2 vertices a_1, a_2 of type 2a. Again, since G is $(diamond, butterfly, K_4)$-free, a_1 and a_2 cannot have a common neighbour on the cycle. Without loss of generality, let a_1 have neighbours v_1 and v_2 and a_2 have neighbours v_3 and v_4. Suppose v_5 were black. $G[v_1, a_1, v_2, v_5]$ is a paw, so v_1 would be white, so a_1 and v_2 would be black. Similarly, a_2 and v_3 would be black, which cannot happen, since v_2 cannot have two black neighbours if it is black. Therefore v_5 must be white, destroying the cycle.

R30 If C is a cycle in G of length 5, with two type 2a neighbours x, let y be the vertex of C not adjacent to a type 2a vertex and output $(G, B, W \cup \{y\})$.

Suppose that the cycle has no vertex of type 2a. We claim that the cycle, together with the type 2b and type 1 vertices dominates the graph. Indeed, suppose not, then there would be type 0 vertices x, x' and a type 2a or type 1 vertex z such that $G[z, x, x']$ is a P_3. Without loss of generality, let v_1 be a neighbour of z on the cycle, then $G[v_1, v_2, v_5, z, x, x']$ is an $S_{1,1,3}$, which is a contradiction.

There are at most 5 valid colourings of the cycle (black, black, white, black, white in order and cyclic rotations of this). Each vertex v_i can have at most one neighbour of type 1 (otherwise an $S_{1,1,3}$ would be present). Thus for any of these five colourings of the cycle, there is at most one valid colour for each of the neighbours of the cycle. This leads to the following rule.

R31 If C is a cycle in G of length 5 and it's neighbours dominates the graph, try all 5 of the possible colourings for the cycle. For each such colouring, find the unique valid colouring of its neighbours (if such a colouring exists) and then apply Lemma 69.

Note that the above rule also applies in the case where one of the neighbours of the cycle is of type 2a, say x. Note that there is at most one possible valid colouring for

x for any given colouring of the cycle. We have thus reduced to the case where the cycle, together with it's neighbours does not dominate the graph. By the above arguments, this means there must be type 0 vertices z, z' such that $G[x, z, z']$ is a P_3. Without loss of generality, assume that x is adjacent to v_1 and v_2.

Suppose, for contradiction that x can be coloured white. Then v_1 and v_2 must be black, so v_3 and v_5 must be white, so v_4 must be black and must therefore have a black neighbour y. v_4 can only have neighbours of type 1 or 2b. If y was of type 2b, without loss of generality adjacent to v_2, then $G[v_2, v_3, v_4, y]$, would form a C_4, so y would have to be white, which would be a contradiction. Thus y must be of type 1. Note that then x and y must be adjacent, otherwise $G[v_4, y, v_5, v_3, v_2, x]$ would be an $S_{1,1,3}$. Since x is white, z must be black. This means that z cannot be adjacent to y (since y is black and matched to v_4). The result of this is that $G[v_4, v_3, v_5, y, x, z]$ is an $S_{1,1,3}$. This contradiction means that x must be coloured black in any valid colouring. It must be matched with either v_1 or v_2. Thus all of the other neighbours of x must be white, yielding the following rule.

R32 If C is a cycle in G of length 5, and let x be it's neighbour of type 2a. Output $(G, B, W \cup (N(x) \setminus C)$.

After applying this rule and the Rule R3, we find that Rule R31 can be applied to the component containing the cycle, since the type 2a vertex has no more neighbours outside the cycle and so the cycle, together with its neighbours dominates its component of the graph.

We can thus reduce to the case where the graph has no induced cycles of length ≥ 5, in which case the problem can be solved in polynomial time.

Since all of the above reduction rules can be done in polynomial time, we conclude that the problem can be solved in polynomial time in the class of $S_{1,1,3}$-free graphs.

6.6 Conclusion

Using Theorem 66 we obtain the following corollary:

Corollary 70. *Let t be a non-negative integer, then the* Minimum Restricted Efficient Edge Domination *problem can be solved in polynomial time in the class of $(S_{1,1,3} + tK_2)$-free graphs.*

Proof. Given an instance (G, B, W) of the problem, we can find all dominating induced matchings consisting of less than t edges in polynomial time. If G has any induced matchings on at least t edges, we branch over all possible choices of induced matchings M on $2t$ vertices and try to solve the problem optimally on $(G, B \cup M, W)$. There are at most n^{2t} such induced matchings. Note that after the reduction rules R1-R4 have been applied, we will have removed the induced matching M and any vertices adjacent to a vertex in M. Since M forms the graph tK_2, the resulting graph must be $S_{1,1,3}$-free and we solve the instance using our usual algorithm.

If any of the branches of the algorithm give a valid solution we output the optimal one. If not, then there is no valid solution and we output Impossible. □

This allows us to draw the following conclusion:

Corollary 71. *Let F be a graph on at most 6 vertices. Then the* Efficient Edge Domination *problem can be solved in polynomial time in the class of F-free graphs if and only if every component of F is a graph of the form $S_{i,j,k}$, where $i, j, k \geq 0$. If F contains at most 6 vertices and is not of this form, the decision version of* Efficient Edge Domination *is NP-complete in the class of F-free graphs.*

Proof. If F has a component not of the form $S_{i,j,k}$, the result follows from Theorem 64. We need only prove the cases where F has exactly 6 vertices. The case where $F = S_{1,2,2}$ is solved in [Korpelainen, 2012]. Since the problem can be solved in polynomial time for P_7-free graphs [Brandstädt and Mosca, 2011b], it can also be solved for P_6-free, $P_5 + K_1$-free, $P_4 + P_2$-free and $2P_3$-free graphs. For all the remaining cases, F is an induced subgraph of $S_{1,13} + 6K_2$ (see Table 6.1), so the result follows from Corollary 71. □

In this chapter, we showed that Efficient Edge Domination is solvable in polynomial time for $S_{1,1,3}$-free graphs and completed the complexity characterisation of the problem for all classes defined by a single forbidden induced subgraph on at most 6 vertices. We also showed that the problem is fixed parameter tractable when parameterized by the size of the induced matching or by the number of vertices not in the matching.

Of results of type 3, a natural question for further study is the complexity of the problem in $S_{1,2,3}$-free graphs. Note that this class includes both $S_{1,1,3}$-free graphs and $S_{1,2,2}$-free graphs. $S_{1,2,3}$-free graphs have been previously studied in the literature, for example in [Lozin, 2002a], it was shown that $S_{1,2,3}$-free bipartite graphs have bounded clique width. Since Efficient Edge Domination can be solved in linear time for P_7-free graphs, it is also natural to ask about P_8-free graphs.

Another challenging problem is to solve Efficient Edge Domination on

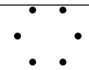
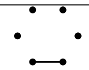
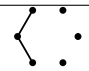
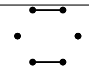
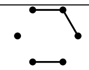
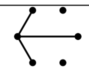
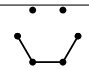
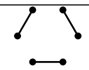
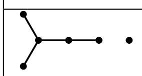
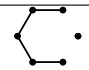
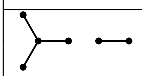
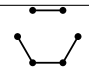
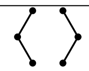
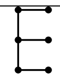
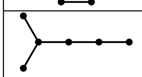
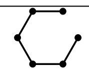
Graph	Graph Name	Reference
	Empty	Corollary 71
	$P_2 + 4K_1$	Corollary 71
	$P_3 + 3K_1$	Corollary 71
	$2P_2 + 2K1$	Corollary 71
	$P_3 + P_2 + K_1$	Corollary 71
	$K_{1,3} + 2K_1$	Corollary 71
	$P_4 + 2K_1$	Corollary 71
	$3P_2$	Corollary 71
	$S_{1,1,2} + K_1$	Corollary 71
	$P_5 + K_1$	[Brandstädt and Mosca, 2011b] (P_7-free graphs) and Corollary 71
	$K_{1,3} + P_2$	Corollary 71
	$P_4 + P_2$	[Brandstädt and Mosca, 2011b] (P_7-free graphs) and Corollary 71
	$2P_3$	[Brandstädt and Mosca, 2011b] (P_7-free graphs)
	$S_{1,2,2}$	[Korpelainen, 2012]
	$S_{1,1,3}$	Corollary 71
	P_6	[Brandstädt and Mosca, 2011b] (P_7-free graphs)

Table 6.1: The graphs F on 6 vertices, such that Efficient Edge Domination is solvable in polynomial time for F-free graphs.

$(C_k, C_{k+1}, C_{k+2}, \ldots)$-free graphs for $k > 5$, extending the recent result for $k = 5$ [Brandstädt et al., 2010].

Part III

Colouring Problems

Chapter 7

Colouring subclasses of triangle-free graphs

7.1 Introduction

A vertex colouring is an assignment of colours to the vertices of a graph G in such a way that no edge connects two vertices of the same colour. The VERTEX COLOURING problem consists of finding a vertex colouring with the minimum possible number of colours. This number is called the *chromatic number* of G and is denoted by $\chi(G)$. If G admits a vertex colouring with at most k colours, we say that G is k-colourable. The k-COLOURABILITY problem consists of deciding whether a graph is k-colourable.

Note that in the VERTEX COLOURING, rather than trying to find a single independent set of maximum size, we try to partition the vertex set of the graph into the minimum possible number of independent sets. However, in some classes, finding a maximum independent set is used as a step in finding a vertex colouring of the graph.

From a computational point of view, k-COLOURABILITY ($k \geq 3$) and the decision version of VERTEX COLOURING are hard problems, i.e. both of them are NP-complete. Moreover, the problems remain NP-complete in many restricted graph families. For instance, 3-COLOURABILITY is NP-complete for planar graphs [Dailey, 1980], 4-COLOURABILITY is NP-complete for graphs containing no induced path on 8 vertices [Broersma et al., 2010], the decision version of VERTEX COLOURING is NP-complete for line graphs [Holyer, 1981]. On the other hand, for graphs in some special classes, the problems can be solved in polynomial time. For instance, 3-COLOURABILITY is solvable for graphs containing no induced path on 6 vertices [Randerath and Schiermeyer, 2004], k-COLOURABILITY (for any value of k) is solvable for graphs containing no induced path on 5 vertices [Hoàng et al., 2010], and Vertex Colouring (and therefore also k-Colourability for any value of k) is solvable for perfect graphs [Grötschel et al., 1984]. It has also been show that in the subclass of triangle-free graphs where all subdivisions of $K_{2,3}$ are also excluded as induced subgraphs, the chromatic number is bounded by 3 and a vertex colouring can be found in polynomial time [Radovanović and Vušković, 2012].

Recently, much attention has been paid to the complexity of the problems in graph classes defined by forbidden induced subgraphs. Many results of this type were mentioned above, some others can be found in [Brandt, 2002b; Broersma et al., 2009, 2012; Kamiński and Lozin, 2007a,b; Kochol et al., 2003; Král' et al., 2001; Le et al., 2007; Maffray and Preissmann, 1996; Randerath et al., 2002; Woeginger and Sgall, 2001]. In [Král' et al., 2001], the authors systematically study Vertex Colouring on graph classes defined by a single forbidden induced subgraph, and give a complete characterisation of those for which the problem is polynomial-time solvable and those for which it is NP-complete. In particular, the problem is NP-complete for triangle-free graphs. More generally, from the results in [Kamiński and Lozin, 2007a] it follows that the problem is NP-complete in any subclass of triangle-free graphs defined by a finite collection of forbidden induced subgraphs, each of which contains a cycle. This motivates us to study the problem in subclasses of triangle-free graphs obtained by forbidding graphs without cycles, i.e. forests. In this chapter we prove polynomial-time solvability of the problem in many classes of this type. In particular, our results, combined with some previously known facts, provide a complete description of the complexity status of the problem in subclasses of triangle-free graphs obtained by forbidding a forest with at most 6 vertices.

7.2 Preliminaries

Many graph classes that are important from a practical or theoretical point of view can be described in terms of forbidden induced subgraphs. For instance, by definition, forests form the class of graphs without cycles, and due to König's Theorem, bipartite graphs are graphs without odd cycles. Bipartite graphs are precisely the 2-colourable graphs, and recognising 2-colourable graphs is a polynomially solvable task. However, the recognition of k-colourable graphs is an NP-complete problem for any $k \geq 3$.

In the present chapter, we study the computational complexity of the Vertex Colouring problem in subclasses of triangle-free graphs. The family of these classes contains both NP-hard and polynomially solvable cases of the problem. For classes defined by a single additional forbidden induced subgraph, a summary of known results is presented in the following theorem (see also Table 7.1), where we also prove one more result that can easily be derived from known results.

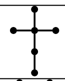
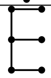
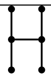
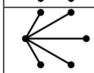
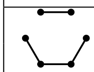
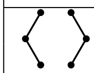
Graph	Graph Name	Complexity	Reference
	Cross	P	[Randerath and Schiermeyer, 2002]
	$S_{1,2,2}$	P	[Randerath, 2004]
	H	P	[Randerath, 2004] (see also Theorem 92 for a shorter proof)
	$K_{1,5}$	NPC	[Maffray and Preissmann, 1996]
	$P_4 + P_2$	P	[Broersma et al., 2010] (see also Theorem 89 for a more general result)
	$2P_3$	P	[Broersma et al., 2012]

Table 7.1: Forests F for which the complexity of Vertex Colouring in the class Free(K_3, F) is known.

Theorem 72. *Let F be a graph. If F contains a cycle or $F = K_{1,5}$, then the* Vertex Colouring *problem is NP-complete in the class $Free(K_3, F)$. If F is isomorphic to $S_{1,2,2}, H, cross, P_4 + P_2, 2P_3$ or P_6, then the problem is polynomial-time solvable in the class $Free(K_3, F)$.*

Proof. If F contains a cycle, then the NP-completeness of the problem follows from the fact that it is NP-complete for graphs of girth at least $k + 1$, i.e. in the class $Free(C_3, C_4, \ldots, C_k)$, for any fixed value of k (see e.g. [Kamiński and Lozin, 2007a; Král' et al., 2001]). The NP-completeness of the problem in the class of $(K_3, K_{1,5})$-free graphs was shown in [Maffray and Preissmann, 1996].

In [Randerath, 2004; Randerath and Schiermeyer, 2002; Randerath, 1998] Randerath et al. showed that every graph in the following three classes is 3-colourable and that a 3-colouring can be found in polynomial time: $Free(K_3, H)$, $Free(K_3,\ S_{1,2,2})$, $Free(K_3, cross)$. Therefore Vertex Colouring is polynomial-time solvable in these three classes.

The polynomial-time solvability of the problem in the class $Free(K_3, P_4 + P_2)$ was shown in [Broersma et al., 2010] and for the class $Free(K_3, 2P_3)$, it was proved in [Broersma et al., 2012].

The conclusion that the problem is solvable for (K_3, P_6)-free graphs can be derived from two facts. First, the clique-width of graphs in this class is bounded by a constant [Brandstädt et al., 2006]. Second, the VERTEX COLOURING problem is solvable in polynomial time on graphs of bounded clique-width [Rao, 2007]. □

A particular corollary of this theorem is that the VERTEX COLOURING problem is solvable in any subclass of triangle-free graphs defined by forbidding a forest with at most 5 vertices.

Corollary 73. *For each forest F on 5 vertices, the* VERTEX COLOURING *problem in the class $Free(K_3, F)$ is solvable in polynomial time.*

Proof. If F contains no edge, then the problem is trivial in the class of $Free(K_3, F)$, since the size of graphs in this class is bounded by a constant (by Ramsey's Theorem). If F contains at least one edge, then it is an induced subgraph of at least one of the following graphs: H, $S_{1,2,2}$, $cross$, P_6. Therefore $Free(K_3, F)$ is a subclass of one the classes $Free(K_3, H)$, $Free(K_3, S_{1,2,2})$, $Free(K_3, cross)$, $Free(K_3, P_6)$, and thus the result follows from Theorem 72. □

In the subsequent sections we study subclasses of triangle-free graphs defined by forbidding forests with more than 5 vertices and prove polynomial-time solvability of the problem in many classes of this type.

7.3 (K_3, F)-free graphs with F containing an isolated vertex

In this section we study graph classes $Free(K_3, F)$ with F being a forest on 6 vertices, at least one of which is isolated. Without loss of generality we may assume that F contains at least one edge, since otherwise there are only finitely many graphs in the class $Free(K_3, F)$ (by Ramsey's Theorem). Throughout the section, an isolated vertex in F is denoted by v and the rest of the graph is denoted by F_0, i.e. $F_0 = F - v$.

Lemma 74. *Let F be a forest on 6 vertices with at least one edge and at least one isolated vertex. Then the chromatic number of any graph G in the class $Free(K_3, F)$ is at most 4 and a 4-colouring can be found in polynomial time.*

Proof. Suppose that $F_0 \neq P_3 + P_2$. Then it is not difficult to verify that F_0 is an induced subgraph of H, $S_{1,2,2}$ or $cross$. Therefore the chromatic number of (K_3, F_0)-free graphs is at most 3 (see [Randerath, 2004; Randerath and Schiermeyer, 2002]). As a result, the chromatic number of any (K_3, F)-free graph is at most 4. To see this, observe that for any vertex x, the graph $G \setminus N(x)$ is 3-colourable, while $N(x)$ is an independent set.

Now let $F_0 = P_3 + P_2$. Let ab be an edge in a (K_3, F)-free graph G. (If G has no edges, the chromatic number is 1 and we are done.) We will show that $G_0 := G - (N(a) \cup N(b))$ is a bipartite graph. Notice that since G is K_3-free, both $N(a)$ and $N(b)$ induce an independent set. We may assume that at least one of $N(a) \setminus \{b\}, N(b) \setminus \{a\}$ is non-empty (otherwise each connected component of G has at most two vertices and thus G is trivially 2-colourable). Obviously G_0 is C_k-free for any odd $k \geq 7$, since otherwise G_0 (and therefore G) contains a $P_3 + P_2$. Therefore we may assume that G_0 contains a C_5 (otherwise G_0 is bipartite). Let $c \in N(b) \setminus \{a\}$. Since G is triangle-free, c can have at most two neighbours in the C_5, and if it has two, they must be non-consecutive vertices of the C_5. Thus c is non-adjacent to at least three vertices in C_5, say d, e, f, such that $G[d, e, f]$ is isomorphic to $P_2 + K_1$. But now $G[a, b, c, d, e, f]$ is isomorphic to $P_3 + P_2 + K_1$, which is a forbidden graph for G. This contradiction shows that G_0 has no odd cycles, i.e. G_0 is a bipartite graph. If V_0^1, V_0^2 are two colour classes of G_0, then $N(a)$, $N(b)$, V_0^1, V_0^2 are four colour classes of G. □

In view of Lemma 74 and the polynomial-time solvability of 2-Colourability, all we have to do to solve the problem in the classes under consideration is to develop a tool for deciding 3-colourability in polynomial time. For this, we use a result from [Randerath et al., 2002]. A set $D \subseteq V(G)$ is *dominating* in G if every vertex $x \in V(G) \setminus D$ has at least one neighbour in D.

Lemma 75. *[Randerath et al., 2002] For a graph $G = (V, E)$ with a dominating set D, we can decide 3-colourability and determine a 3-colouring in time $O(3^{|D|}|E|)$.*

If a graph $G \in Free(K_3, F)$ is F_0-free, then by Corollary 73, the problem is solvable for G in polynomial time. If G has an induced F_0, then the vertices of F_0 form a dominating set in G. Summarising the above discussion, we obtain the following result.

Theorem 76. *Let F be a forest on 6 vertices with at least one isolated vertex. Then the* Vertex Colouring *problem is polynomial-time solvable in the class $Free(K_3, F)$.*

All forests satisfying the conditions of Theorem 76 are listed in Table 7.2.

7.4 Graphs of bounded clique-width

In Section 7.2, we mentioned that the polynomial-time solvability of the Vertex Colouring problem in the class of (K_3, P_6)-free graphs follows from the fact that the clique-

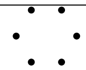
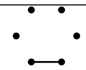
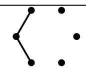
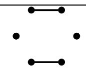
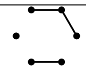
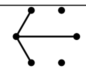
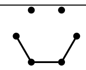
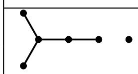
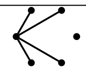
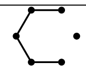
Graph	Graph Name
	Empty
	$P_2 + 4K_1$
	$P_3 + 3K_1$
	$2P_2 + 2K1$
	$P_3 + P_2 + K_1$
	$K_{1,3} + 2K_1$
	$P_4 + 2K_1$
	$S_{1,1,2} + K_1$
	$K_{1,4} + K_1$
	$P_5 + K_1$

Table 7.2: Forests F for which polynomial-time solvability of Vertex Colouring in the class Free(K_3, F) follows from Theorem 76.

width of graphs in this class is bounded by a constant. In the present section we use that same idea to solve the problem in the following two classes: $Free(K_3, S_{1,1,3})$ and $Free(K_3, K_{1,3} + K_2)$.

This means that in order to prove polynomial-time solvability of the Vertex Colouring problem in the classes $Free(K_3, S_{1,1,3})$ and $Free(K_3, K_{1,3} + K_2)$, all we have to do is to show that the clique-width of graphs in these classes is bounded. In our proofs we use the following helpful facts:

Fact 1: The clique-width of graphs with vertex degree at most 2 is bounded by 4 (see e.g. [Courcelle and Olariu, 2000]).

Fact 2: The clique-width of $S_{1,1,3}$-free bipartite graphs [Lozin, 2002a] and $(K_{1,3} + K_2)$-free bipartite graphs [Lozin et al., 2008] is bounded by a constant.

Fact 3: For a constant k and a class of graphs X, let $X_{[k]}$ denote the class of graphs obtained from graphs in X by deleting at most k vertices. Then the clique-width of graphs in X is bounded if and only if the clique-width of graphs in $X_{[k]}$ is bounded [Lozin and Rautenbach, 2004].

Fact 4: For a graph G, the *subgraph complementation* is the operation that consists of complementing the edges in an induced subgraph of G. Also, given two disjoint subsets of vertices in G, the *bipartite subgraph complementation* is the operation which consists of complementing the edges between the subsets. For a constant k and a class of graphs X, let $X^{(k)}$ be the class of graphs obtained from graphs in X by applying at most k subgraph complementations or bipartite subgraph complementations. Then the clique-width of graphs in $X^{(k)}$ is bounded if and only if the clique-width of graphs in X is bounded [Kamiński et al., 2009].

Fact 5: The clique-width of graphs in a hereditary class X is bounded if and only if it is bounded for connected graphs in X (see e.g. [Courcelle and Olariu, 2000]).

Facts 2 and 5 allow us to reduce the problem to connected non-bipartite graphs in the classes $Free(K_3, S_{1,1,3})$ and $Free(K_3, K_{1,3} + K_2)$, i.e. to connected graphs in these classes that contain an odd induced cycle of length at least five.

Lemma 77. *Let G be a connected $(K_3, S_{1,1,3})$-free graph containing an odd induced cycle C of length at least 7. Then $G = C$.*

Proof. Let $C = v_1 - v_2 - \cdots - v_{2k} - v_{2k+1} - v_1$ be an induced cycle in G, of length $2k + 1$, $k \geq 3$. Suppose that there exists a vertex $v \in V(G) \setminus V(C)$, which is adjacent to a vertex of C. Without loss of generality, we may assume that v is adjacent to v_1. We claim that v is non-adjacent to v_4. Otherwise, since G is K_3-free, it follows that v is non-adjacent to v_{2k+1}, v_2, v_3, v_5. But now $G[v_4, v_3, v_5, v, v_1, v_{2k+1}]$ is isomorphic to $S_{1,1,3}$, a contradiction. Thus v is non-adjacent to v_4. This implies that v is adjacent to v_3, since otherwise $G[v_1, v, v_{2k+1}, v_2, v_3, v_4]$ would be isomorphic to $S_{1,1,3}$. Now repeating the same argument with v_3 playing the role of v_1, we conclude that v is adjacent to v_5. But now $G[v_1, v_2, v_{2k+1}, v, v_5, v_4]$ is isomorphic to $S_{1,1,3}$. This contradiction shows that $G = C$. $\square$

Lemma 78. *Let G be a connected $(K_3, K_{1,3} + K_2)$-free graph containing an odd induced cycle C_{2k+1}, $k \geq 3$. If $k \geq 4$ then $G = C_{2k+1}$ and if $k = 3$ then $|V(G)| \leq 28$.*

Proof. Let $C = v_1 - v_2 - \cdots - v_{2k} - v_{2k+1} - v_1$ be an induced cycle of length $2k+1$ in G. First consider the case when $k \geq 4$. Suppose that there exists a vertex $v \in V(G) \setminus V(C)$ which is adjacent to some vertex of C, say v_1. Since G is K_3-free, it follows that v is non-adjacent to v_{2k+1}, v_2. We claim that for every pair of vertices $\{v_i, v_{i+1}\}$, with $i = 4, 5, \ldots, 2k-2$, vertex v is adjacent to exactly one of v_i, v_{i+1}. Clearly, since G is K_3-free, v has a non-neighbour in $\{v_i, v_{i+1}\}$. If v has no neighbours in $\{v_i, v_{i+1}\}$, then $G[v_1, v_2, v, v_{2k+1}, v_i, v_{i+1}]$ is isomorphic to $K_{1,3} + K_2$, a contradiction. Now suppose that v is adjacent to v_4. Then it follows that v is complete to $\{v_4, v_6, \ldots, v_{2k-2}\}$ and anticomplete to $\{v_5, v_7, \ldots, v_{2k-1}\}$. But then $G[v_{2k-2}, v, v_{2k-3}, v_{2k-1}, v_2, v_3]$ is isomorphic to $K_{1,3} + K_2$, a contradiction. Thus we may assume that v is adjacent to v_5. This implies that v is complete to $\{v_5, v_7, \ldots, v_{2k-1}\}$ and anticomplete to $\{v_4, v_6, \ldots, v_{2k-2}\}$. It follows that v is non-adjacent to v_{2k}, since G is K_3-free. But now $G[v_5, v_4, v_6, v, v_{2k}, v_{2k+1}]$ is isomorphic to $K_{1,3} + K_2$. This contradiction shows that $G = C$.

Now consider the case where $k = 3$ and let $v \in V(G) \setminus V(C)$ be adjacent to v_1. As before, v has exactly one neighbour in $\{v_4, v_5\}$. By symmetry, we may assume that v is adjacent to v_4. Hence v has no neighbours in $\{v_2, v_3, v_5, v_7\}$. Finally, observe that v is non-adjacent to v_6, since otherwise $G[v_6, v_5, v_7, v, v_2, v_3]$ would be isomorphic to $K_{1,3}+K_2$. Therefore we conclude that each vertex $v \in V(G) \setminus V(C)$ that is adjacent to some vertex $v_i \in V(C)$, is either complete to $\{v_i, v_{i+3}\}$ and anticomplete to $V(C) \setminus \{v_i, v_{i+3}\}$, or complete to $\{v_i, v_{i+4}\}$ and anticomplete to $V(C) \setminus \{v_i, v_{i+4}\}$ (here subscripts are taken modulo 7).

Let U_j denote the set of vertices at distance j from the cycle. We claim that:

- $|U_1| \leq 7$. Indeed, if $|U_1| > 7$, then there exist two vertices $z, z' \in U_1$ that are complete to $\{v_i, v_{i+3}\}$ (and thus anticomplete to $V(C) \setminus \{v_i, v_{i+3}\}$) for some value of i.

Since G is K_3-free, z, z' must be non-adjacent. But then $G[v_i, z, z', v_{i+1}, v_{i+4}, v_{i+5}]$ is isomorphic to $K_{1,3} + K_2$, a contradiction.

- *Each vertex of U_1 has at most one neighbour in U_2.* Indeed, suppose a vertex $x \in U_1$ has two neighbours $y, z \in U_2$, and without loss of generality let x be complete to $\{v_i, v_{i+3}\}$ (and thus anticomplete to $V(C) \setminus \{v_i, v_{i+3}\}$). Since G is K_3-free, it follows that y, z are non-adjacent. But then $G[x, y, z, v_i, v_{i+4}, v_{i+5}]$ is isomorphic to $K_{1,3} + K_2$, a contradiction.

- *Each vertex of U_2 has at most one neighbour in U_3*, which can be proved by analogy with the previous claim.

- *For each $i \geq 4$, U_i is empty.* Indeed, assume without loss of generality that $U_4 \neq \emptyset$ and let u_4, u_3, u_2, u_1 be a path from U_4 to C with $u_j \in U_j$ and u_1 being adjacent to v_i. Then $G[v_i, v_{i-1}, v_{i+1}, u_1, u_3, u_4]$ is isomorphic to $K_{1,3} + K_2$, a contradiction.

From the above claims we conclude that $V(G) = V(C) \cup U_1 \cup U_2 \cup U_3$, $|U_3| \leq |U_2| \leq |U_1| \leq 7 = |V(C)|$, and therefore $|V(G)| \leq 28$. □

Thus Lemmas 77 and 78 and Fact 2 further reduce the problem to graphs containing a C_5.

Lemma 79. *If G is a connected $(K_3, S_{1,1,3})$-free graph containing a C_5, then the clique-width of G is bounded by a constant.*

Proof. Let G be a connected $(K_3, S_{1,1,3})$-free graph and let $C = v_1 - v_2 - v_3 - v_4 - v_5 - v_1$ be an induced cycle of length five in G. If $G = C$ then the clique-width of G is at most 4 (Fact 1). Therefore we may assume that there exists at least one vertex $v \in V(G) \setminus V(C)$. Since G is K_3-free, v can be adjacent to at most two vertices of C, and if v has two neighbours in C, they must be non-consecutive vertices of the cycle. We denote the set of vertices in $V(G) \setminus V(C)$ that have exactly i neighbours in C by N_i, $i \in \{0, 1, 2\}$. Also, for $i = 1, \ldots, 5$, we let V_i denote the set of vertices in N_2 adjacent to $v_{i-1}, v_{i+1} \in V(C)$ (throughout the proof subscripts i are taken modulo 5). We call two different sets V_i and V_j *consecutive* if v_i and v_j are consecutive vertices of C, and *opposite* otherwise. Finally, we call V_i *large* if $|V_i| \geq 2$, and *small* otherwise. The proof of the lemma will be given through a series of claims.

(1) *Each V_i is an independent set.* This immediately follows from the fact that G is K_3-free.

(2) N_0 *is an independent set.* Indeed, suppose xy is an edge connecting two vertices $x, y \in N_0$, and, without loss of generality, let y be adjacent to a vertex $z \in N_1 \cup N_2$. Let $v_i \in V(C)$ be a neighbour of z. Since G is K_3-free, z is non-adjacent to x, v_{i-1}, v_{i+1}. But then $G[v_i, v_{i-1}, v_{i+1}, z, y, x]$ is isomorphic to $S_{1,1,3}$, a contradiction.

(3) *Any vertex* $x \in N_1 \cup N_2$ *has at most one neighbour in* N_0. Suppose $x \in N_1 \cup N_2$ is adjacent to $z, z' \in N_0$, and let $v_i \in V(C)$ be a neighbour of x. Since G is K_3-free, it follows that x is non-adjacent to v_{i-1}, v_{i+1}. Furthermore, x is adjacent to at most one of v_{i-2}, v_{i+2}. By symmetry we may assume that x is non-adjacent to v_{i-2}. But now $G[x, z, z', v_i, v_{i-1}, v_{i-2}]$ is isomorphic to $S_{1,1,3}$, a contradiction.

(4) $|N_1| \leq 5$. Indeed, if there are two vertices $x, x' \in N_1$ which are adjacent to the same vertex $v_i \in V(C)$, then $G[v_i, x, x', v_{i+1}, v_{i+2}, v_{i+3}]$ is isomorphic to $S_{1,1,3}$, a contradiction.

(5) *If* V_i *and* V_j *are opposite sets, then no vertex of* V_i *is adjacent to a vertex of* V_j. This immediately follows from the fact that G is K_3-free.

(6) *If* V_i *and* V_j *are consecutive, then every vertex* $x \in V_i$ *has at most one non-neighbour in* V_j. Suppose $x \in V_i$ has two non-neighbours $y, y' \in V_j$. By symmetry, we may assume that $j = i + 1$. But now, by Claim (1), $G[v_{i-3}, y, y', v_{i-2}, v_{i-1}, x]$ is isomorphic to $S_{1,1,3}$, a contradiction.

(7) *If* V_i *and* V_j *are two opposite large sets, then no vertex in* N_0 *has a neighbour in* $V_i \cup V_j$. Without loss of generality assume that $i = 1$ and $j = 4$, and suppose for a contradiction that a vertex $x \in N_0$ has a neighbour $y \in V_1$. If x is non-adjacent to some vertex $z \in V_4$, then $G[v_3, v_4, z, v_2, y, x]$ is isomorphic to $S_{1,1,3}$, a contradiction. Therefore x is complete to V_4. But now, by Claim (1), $G[x, z, z', y, v_2, v_1]$ with $z, z' \in V_4$ is isomorphic to $S_{1,1,3}$, a contradiction.

Since G is connected and N_0 is an independent set, every vertex of N_0 has a neighbour in $N_1 \cup N_2$. Let V_0 be the set of vertices in N_0, all of whose neighbours belong to the large sets V_i. Let G_0 be the subgraph of G induced by V_0 and the large sets. From Claims (2),(3) and (4), it follows that at most 25 vertices of G do not belong to G_0. Therefore, by Fact 3, the clique-width of G is bounded if and only if it is bounded for G_0. We may assume that G has at least one large set, since otherwise G_0 is empty. We will show that G_0 has bounded clique-width by examining all possible combinations of large sets.

Case 1: Suppose that for every large set V_i there is an opposite large set V_j. Then it follows from Claim (7) that $V_0 = \emptyset$. In order to see that G_0 has bounded clique-width, we complement the edges between every pair of consecutive large sets. By Claims (5) and (6), the resulting graph has maximum degree at most 2. From Fact 1 it follows that this graph is of bounded clique-width, and therefore, applying Fact 4, G_0 has bounded clique-width.

Case 1 allows us to assume that G contains a large set such that the opposite sets are small. Without loss of generality we let V_1 be large, and V_3 and V_4 be small. The rest of the proof is based on the analysis of the size of the sets V_2 and V_5.

Case 2: V_2 and V_5 are large. Then, by Claims (1), (2), (5), and (7), G_0 is a bipartite graph with bipartition $(V_1, V_2 \cup V_5 \cup V_0)$. Therefore by Fact 2, G_0 has bounded clique-width.

Case 3: V_2 and V_5 are small. Then by Claims (1) and (2), G_0 is a bipartite graph with bipartition (V_1, V_0), and therefore, by Fact 2, G_0 has bounded clique-width.

Case 4: V_2 is large and V_5 is small, i.e. G_0 is induced by $V_0 \cup V_1 \cup V_2$. Consider a vertex $x \in V_0$ that has a neighbour $y \in V_1$ and a neighbour $z \in V_2$. Then y and z are non-adjacent (since G is K_3-free) and therefore, by Claim (6), y is complete to $V_2 \setminus \{z\}$ and z is complete to $V_1 \setminus \{y\}$. From the K_3-freeness of G it follows that x is anticomplete to $(V_1 \cup V_2) \setminus \{y, z\}$.

Let V_0' denote the vertices of V_0 that have neighbours both in V_1 and V_2, and let V_i' $(i = 1, 2)$ denote the vertices of V_i that have neighbours in V_0'. Also, let $V_i'' = V_i - V_i'$ for $i = 0, 1, 2$, and $G_0' = G_0[V_0' \cup V_1' \cup V_2']$, $G_0'' = G_0[V_0'' \cup V_1'' \cup V_2'']$.

By Claim (3), V_0'' is anticomplete to $V_1' \cup V_2'$. Also, it follows from the above discussion that V_0' is anticomplete to $V_1'' \cup V_2''$, that V_1' is complete to V_2'', and that V_2' is complete to V_1''. Therefore by complementing the edges between V_1' and V_2'', and between V_2' and V_1'', we disconnect G_0' from G_0''. The graph G_0'' is a bipartite graph, since every vertex of V_0'' has neighbours either in V_1'' or in V_2'' but not in both. Thus it follows from Fact 2 that G_0'' has bounded clique-width. To see that G_0' has bounded clique-width, we complement the edges between V_1' and V_2'. This operation transforms G_0' into a collection of disjoint triangles. Therefore the clique-width of G_0' is bounded. Now it follows from Fact 4 that G_0 has bounded clique-width. □

Similarly to Lemma 79, one can prove the following result.

Lemma 80. *If G is a connected $(K_3, K_{1,3} + K_2)$-free graph containing a C_5, then the clique-width of G is bounded by a constant.*

Proof. The proof is similar to the proof of Lemma 79. Let G be a connected $(K_3, K_{1,3}+K_2)$-free graph and let $C = v_1 - v_2 - v_3 - v_4 - v_5 - v_1$ be an induced cycle of length five in G. If $G = C$ then the clique-width of G is at most 4 (Fact 1). Therefore we may assume that there exists at least one vertex $v \in V(G) \setminus V(C)$. Since G is K_3-free, v can be adjacent to at most two vertices in C, and if v has two neighbours in C, they must be non-consecutive vertices of C. We denote the set of vertices in $V(G) \setminus V(C)$ that have exactly i neighbours in C by N_i, $i \in \{0, 1, 2\}$. Also, for $i = 1, \ldots, 5$, we let V_i denote the set of vertices in N_2 adjacent to $v_{i-1}, v_{i+1} \in V(C)$ (throughout the proof subscripts i are taken modulo 5). We call two different sets V_i and V_j *consecutive* if v_i and v_j are consecutive vertices of C, and *opposite* otherwise. Finally, we call V_i *large* if $|V_i| \geq 7$, and *small* otherwise. The proof of the lemma will be given through a series of claims.

(1) *Each V_i is an independent set.* This immediately follows from the fact that G is K_3-free.

(2) $|N_1| \leq 10$. Indeed, if there are three vertices $x, x', x'' \in N_1$ which are adjacent to the same vertex $v_i \in V(C)$, then $G[v_i, x, x', x'', v_{i+2}, v_{i+3}]$ is isomorphic to $K_{1,3} + K_2$, a contradiction (notice that x, x', x'' are pairwise non-adjacent since G is K_3-free).

(3) *If V_i and V_j are opposite sets, then no vertex of V_i is adjacent to a vertex of V_j.* This immediately follows from the fact that G is K_3-free.

(4) *If V_i and V_j are consecutive, then every vertex of V_i has at most two non-neighbours in V_j.* By symmetry, we may assume $j = i + 1$. Suppose $x \in V_i$ has three non-neighbours $y, y', y'' \in V_j$. Then by Claim (1), $G[v_{i+2}, y, y', y'', v_{i-1}, x]$ is isomorphic to $K_{1,3} + K_2$, a contradiction.

(5) *Each vertex $w \in N_0$ is adjacent to at most two vertices in a set V_i.* Indeed, if $w \in N_0$ were adjacent to three vertices $z, z', z'' \in V_i$, then by Claim (1), $G[w, z, z', z'', v_{i+2}, v_{i+3}]$ would be isomorphic to $K_{1,3} + K_2$, a contradiction.

(6) *N_0 induces a graph of vertex degree at most two. Moreover, if there exists at least one large set, then N_0 is an independent set.* If a vertex $w \in N_0$ has three neighbours $z, z', z'' \in N_0$, then $G[w, z, z', z'', v_1, v_2]$ is isomorphic to $K_{1,3}+K_2$, since G is K_3-free. This contradiction proves the first part of the claim. To prove the second part, assume V_i is a large set and suppose that two vertices $w, w' \in N_0$ are adjacent. Since V_i is large, it follows from Claim (5) that there exist at least three vertices $z, z', z'' \in V_i$ which are anticomplete to $\{w, w'\}$. But now, by Claim (1), $G[v_{i-1}, z, z', z'', w, w']$ is isomorphic to $K_{1,3} + K_2$, a contradiction.

(7) *If V_i and V_j are two opposite large sets, then no vertex in N_0 has a neighbour in $V_i \cup V_j$.* Without loss of generality, assume that $i = 1$ and $j = 4$, and suppose for contradiction, that a vertex $w \in N_0$ has a neighbour $y \in V_1$. Since V_4 is large and since w is adjacent to at most two vertices in V_4 (Claim (5)), it follows that w has two non-neighbours $z, z' \in V_4$. But now, by Claim (1), $G[v_3, v_4, z, z', w, y]$ is isomorphic to $K_{1,3} + K_2$, a contradiction.

(8) *Any vertex $x \in N_1 \cup N_2$ has at most two neighbours in N_0.* Indeed, for any vertex $x \in N_1 \cup N_2$ there exist at least two consecutive vertices of C non-adjacent to x. These two vertices together with x and any three neighbours of x in N_0 would induce a $K_{1,3} + K_2$.

From Claim (6) and Fact 1 we know that the clique-width of $G[N_0]$ is at most 4. Therefore, if all sets V_i are small, then G has bounded clique-width, which follows from Claim (2) and Fact 3.

From now on, we assume that there exists at least one large set V_i. This implies that N_0 is an independent set (Claim (6)). Since G is connected, every vertex of N_0 has a neighbour in $N_1 \cup N_2$. Let V_0 be the set of vertices in N_0, all of whose neighbours belong to the large sets V_i. Let G_0 be the subgraph of G induced by V_0 and the large sets. From Claims (2) and (8), it follows that the size of $V(G) \setminus V(G_0)$ is bounded. Therefore, by Fact 3, the clique-width of G is bounded if and only if it is bounded for G_0. We will show that G_0 has bounded clique-width by examining all possible combinations of large sets.

Case 1: Suppose that for every large set V_i there is an opposite large set V_j. Then it follows from Claim (7) that $V_0 = \emptyset$. Let V_{i-1} and V_{i+1} be large sets. We claim that every vertex $x \in V_i$ is complete to $V_{i-1} \cup V_{i+1}$. For suppose not: let $y \in V_{i+1}$ be a non-neighbour of x. Since V_{i-1} is large, it follows from Claim (4) that x has at least two neighbours $z, z' \in V_{i-1}$. But now, by Claims (1) and (3), $G[x, z, z', v_{i-1}, v_{i+2}, y]$ is isomorphic to $K_{1,3} + K_2$, a contradiction. In order to see that G_0 is of bounded clique-width, we complement the edges between every pair of consecutive large sets. From Claim (4) and the discussion above, it follows that the resulting graph is of vertex degree at most 2. From Fact 1 it follows that this graph has bounded clique-width, and therefore applying Fact 4, G_0 has bounded clique-width.

Case 1 allows us to assume that G contains a large set such that the opposite sets are small. Without loss of generality we let V_1 be large, and V_3 and V_4 be small. The rest of the proof is based on the analysis of the size of the sets V_2 and V_5.

Case 2: V_2 and V_5 are large. Then by Claims (1),(3),(6) and (7), G_0 is a bipartite graph with bipartition $(V_1, V_2 \cup V_5 \cup V_0)$. Therefore by Fact 2, G_0 has bounded clique-width.

Case 3: V_2 and V_5 are small. Then, by Claims (1) and (6), G_0 is a bipartite graph with bipartition (V_1, V_0), and therefore, by Fact 2, G_0 has bounded clique-width.

Case 4: V_2 is large and V_5 is small, i.e. G_0 is induced by $V_0 \cup V_1 \cup V_2$. Consider a vertex $w \in V_0$ that is adjacent to some vertex $x \in V_1$ (resp. $y \in V_2$). We claim that

(9) *w is complete to all the non-neighbours of x in V_2 (resp. of y in V_1).* By symmetry we let x belong to V_1 and for contradiction, suppose that w is non-adjacent to a non-neighbour $z \in V_2$ of x. Since V_1 is large, it follows from Claims (4) and (5) that V_1 contains three vertices x_1, x_2, x_3 adjacent to z and non-adjacent to w. But now, by Claim (1), $G_0[z, x_1, x_2, x_3, x, w]$ is isomorphic to $K_{1,3} + K_2$, a contradiction.

In order to see that G_0 has bounded clique-width, we complement the edges between V_1 and V_2. Let us denote the resulting graph by G_0'. From Facts 4 and 5, it follows that it is enough to show that each connected component of G_0' has bounded clique-width. Let C^* be a component of G_0'. If C^* has maximum vertex degree at most two, then C^* has bounded clique-width by Fact 1. So we may assume that C^* contains a vertex x of degree at least three.

First suppose that $x \in V_1 \cup V_2$. By symmetry, we may assume $x \in V_1$. We know that in the graph G_0', vertex x has at most two neighbours in V_0 (Claim (8)) and at most two neighbours in V_2 (Claim (4)). Therefore, x is adjacent to some vertex $y \in V_2$ and to some vertex $w \in V_0$ in the graph G_0'. Since in the graph G_0 vertex y is a non-neighbour of x, it follows from Claim (9) that y, w are adjacent. Repeating this argument, we conclude that w is complete to $V(C^*) \cap (V_1 \cup V_2)$. By Claim (5), we obtain that $|V(C^*) \cap (V_1 \cup V_2)| \leq 4$. Since each vertex in $V_1 \cup V_2$ has at most two neighbours in V_0 (Claim (8)), we finally conclude that $|V(C^*)| \leq 12$ and therefore the clique-width of C^* is at most 12.

Now suppose that $x \in V_0$ and all vertices of C^* in $V_1 \cup V_2$ have degree at most 2. Since V_0 is an independent set, all neighbours of x are in $V_1 \cup V_2$. Let z, z', z'' denote three neighbours of x. Without loss of generality we may assume that $z, z' \in V_1$ and $z'' \in V_2$ (Claim (5)). Since G is K_3-free, it follows that in C^*, vertex z'' is adjacent to both z and z'. But now $z'' \in V_2$ has degree at least three, contradicting our assumption. $\square$

From Lemmas 77, 78, 79, and 80, we derive the main result of this section.

Theorem 81. *The clique-width of $(K_3, S_{1,1,3})$-free graphs and $(K_3, K_{1,3}+K_2)$-free graphs is bounded by a constant and therefore the* Vertex Colouring *problem is polynomial-time solvable in these classes of graphs.*

7.5 $(K_3, S_{1,2,3}, S_{1,1,2} + P_2)$-free graphs

In this section we prove polynomial-time solvability of the problem in the class of $(K_3, S_{1,2,3}, S_{1,1,2} + P_2)$-free graphs. It is not difficult to see that both $S_{1,2,3}$ and $S_{1,1,2} + P_2$ contain P_4+P_2 as an induced subgraph. Therefore, our result generalises a recent solution of the problem in the class of (K_3, P_4+P_2)-free graphs [Broersma et al., 2010]. Our result is based on a sequence of lemmas.

(a) $S_{1,2,3}$ (b) $S_{1,1,2} + P_2$

Figure 7.1: The graphs $S_{1,2,3}$ and $S_{1,1,2} + P_2$.

Lemma 82. *Let G be a $(K_3, S_{1,2,3}, S_{1,1,2} + P_2)$-free graph. Then the chromatic number of G is at most 4 and a 4-colouring of G can be found in polynomial time.*

Proof. We may assume G is connected and contains an edge ab. Note that since G is K_3-free, $G[N(a) \cup N(b)]$ is a bipartite graph. Let $X = V(G) \setminus (N(a) \cup N(b))$. We will now show that $G[X]$ is bipartite, in which case G is 4-colourable. Indeed, suppose for contradiction that $G[X]$ is not bipartite. Then, since it is K_3-free, it must contain an induced odd cycle $v_1 - \cdots - v_{2k+1} - v_1$ with $k \geq 2$.

Let $w_1, w_2, \ldots, w_q$ be a shortest path from this cycle to a, with $w_q = a$ and $w_1 = v_i$ for some $i \in \{1, \ldots, 2k + 1\}$. If $q = 3$ then $w_2 \in N(a) \setminus \{b\}$. In this case let $w_4 = b$.

Vertex w_2 cannot be adjacent to v_{i-1} or v_{i+1} since G is K_3-free. But now w_2 must be adjacent to v_{i+2}, otherwise $G[v_i, v_{i-1}, v_{i+1}, v_{i+2}, w_2, w_3, w_4]$ would be isomorphic to $S_{1,2,3}$. Since vertex v_i was chosen arbitrarily, we can repeat this argument k times to find that w_2 must be adjacent to 2 consecutive vertices in the cycle. But this cannot happen, since G is K_3-free. This contradiction completes the proof. □

Lemma 82 reduces VERTEX COLOURING in the class of $(K_3, S_{1,2,3}, S_{1,1,2}+P_2)$-free graphs to 3-COLOURABILITY. We now prove some lemmas to help solve this problem.

Lemma 83. *Let G be a connected $(K_3, S_{1,2,3}, S_{1,1,2} + P_2)$-free graph containing an odd induced cycle C of length at least 9. Then $G = C$.*

Proof. Let $C = v_1 - v_2 - \cdots - v_{2k+1}$ be an induced odd cycle of length at least 9 in G. Let x be adjacent to some vertex v_i on C. Then obviously it is adjacent to neither v_{i-1} nor v_{i+1}, since the graph is K_3-free. If in addition it is non-adjacent to v_{i-2}, then the subgraph of G induced by v_i, v_{i+1}, v_{i-1}, v_{i-2}, x, v_{i+3}, v_{i+4} is either isomorphic to $S_{1,2,3}$ (if x has a neighbour in $\{v_{i+3}, v_{i+4}\}$) or to $S_{1,1,2}+P_2$ (if x has no neighbour in $\{v_{i+3}, v_{i+4}\}$). Therefore, x is adjacent to v_{i-2}. But v_i was an arbitrary vertex of the cycle, so as in the proof of Lemma 82, by iterating this argument k times, we find that G must contain a K_3, which is a contradiction. $\square$

Lemma 84. *Let G be a connected $(K_3, S_{1,2,3}, S_{1,1,2}+P_2)$-free graph containing an induced cycle C of length 7. Then C is dominating.*

Proof. Suppose G is connected and contains an induced cycle $C = v_1 - v_2 - v_3 - v_4 - v_5 - v_6 - v_7 - v_1$. If C is not dominating then there must exist vertices x and y such that y is not adjacent to any vertex of the cycle and x is adjacent to both y and some vertex of the cycle, say v_1. x is non-adjacent to v_2 and v_7 since G is K_3-free. So x must be adjacent to v_4 or v_5, otherwise $G[v_1, v_2, v_7, x, y, v_4, v_5]$ would be isomorphic to $S_{1,1,2} + P_2$. Without loss of generality, assume that x is adjacent to v_4. Since G is K_3-free, x is non-adjacent to v_3 and v_5. Now, x must be adjacent to v_6, otherwise $G[v_1, x, v_2, v_3, v_7, v_6, v_5]$ would be isomorphic to $S_{1,2,3}$. But then $G[v_6, v_5, v_7, x, y, v_2, v_3]$ is isomorphic $S_{1,1,2} + P_2$. This contradiction leads to the conclusion that such vertices x and y cannot exist and thus C is dominating. $\square$

Let B be a connected bipartite induced subgraph of a graph G with at least 3 vertices. We say that the vertices in one part of B are *odd* and those in the other part are *even*. If two vertices are in the same part of B, we say they have the same *parity*. The following lemma is an easy observation.

Lemma 85. *Suppose a graph G has a connected bipartite induced subgraph B, $|V(B)| \geq 3$, and that for every vertex $x \notin B$, x is either complete or anticomplete to the odd vertices in B and is either complete or anticomplete to the even vertices in B. Then all vertices of B except any two adjacent vertices can be deleted from G and the new graph has a 3-colouring if and only if G does.*

Lemma 86. *Let G be a connected $(K_3, S_{1,2,3}, S_{1,1,2}+P_2)$-free graph containing an induced cycle C of even length $k \geq 8$. If a vertex x has a neighbour on the cycle, then x is adjacent to all vertices of the same parity with respect to C.*

Proof. Let x be adjacent to a vertex v_i on the cycle. Then obviously it is adjacent to neither v_{i-1} nor v_{i+1}, since the graph is K_3-free. If it is also non-adjacent to v_{i-2}, then the subgraph of G induced by $v_i, v_{i+1}, v_{i-1}, v_{i-2}, x, v_{i+3}, v_{i+4}$ is either isomorphic to $S_{1,2,3}$ (if x has a neighbour in $\{v_{i+3}, v_{i+4}\}$) or to $S_{1,1,2} + P_2$ (if x has no neighbour in $\{v_{i+3}, v_{i+4}\}$). Therefore, x is adjacent to v_{i-2}. Since vertex v_i was chosen arbitrarily, x must be adjacent to all vertices which have the same parity as v_i. □

Notice that we may assume that G satisfies the following property:

(*) for any two non-adjacent vertices u and v, there exists a neighbour of u which is non-adjacent to v and there exists a neighbour of v which is non-adjacent to u.

Indeed if a pair of vertices does not satisfy Property (*), then the neighbourhood of one of the vertices u, v is included in the neighbourhood of the other. In this case the first vertex can be deleted from the graph G and it is easy to see that the new graph has a 3-colouring if and only if the original graph does.

Lemma 87. *Let G be a $(K_3, S_{1,2,3}, S_{1,1,2} + P_2)$-free graph with Property (*) and let P a be an induced path in G with at least 8 vertices. If a vertex x is adjacent to a vertex of degree 2 in P, then x is adjacent to all vertices of the same parity in P.*

Proof. Let P be the path $v_1 - v_2 - \cdots - v_k$ with $k \geq 8$. Suppose, for contradiction, that x has a neighbour v_i with $2 < i \leq k-1$, such that x is not adjacent to v_{i-2} (the case where x is not adjacent to v_{i+2} is symmetric). Clearly x cannot be adjacent to v_{i-1} or v_{i+1} since G is K_3-free.

If $i < k-3$, then $G[v_i, x, v_{i+1}, v_{i-1}, v_{i-2}, v_{i+3}, v_{i+4}]$ is either isomorphic to $S_{1,2,3}$ (if x has a neighbour in $\{v_{i+3}, v_{i+4}\}$) or to $S_{1,1,2}+P_2$ (if x has no neighbour in $\{v_{i+3}, v_{i+4}\}$). Thus we may assume $i \geq k-3$.

But now if $k \geq 9$ or $k = 8, i \geq k-2$, then $G[v_i, x, v_{i+1}, v_{i-1}, v_{i-2}, v_{i-4}, v_{i-5}]$ is either isomorphic to $S_{1,2,3}$ (if x has a neighbour in $\{v_{i-5}, v_{i-4}\}$) or to $S_{1,1,2}+P_2$ (if x has no neighbour in $\{v_{i-5}, v_{i-4}\}$). This contradiction proves that if $k \geq 9$ or $k = 8, i \neq k-3$, then x must be adjacent to v_{i-2}.

Now let us analyse the case when $k = 8$ and $i = k-3 = 5$. By the above argument for $k = 8, i = 3$, we conclude that x is adjacent to v_7. Since G satisfies Property (*), vertex v_6 must have a neighbour y which is non-adjacent to x. From the first part of the proof, we know that y must be adjacent to v_8 and v_4 and therefore to v_2. But x cannot be adjacent to v_2, since then it would have to be adjacent to v_4, contradicting the fact that G is K_3-free. If x is adjacent to v_1, then $G[y, v_6, v_8, v_4, v_3, v_1, x]$ is an $S_{1,1,2} + P_2$. If x is non-adjacent to v_1, then $G[y, v_4, v_2, v_1, v_6, v_7, x]$ is an $S_{1,2,3}$. This final contradiction completes the proof of the lemma. $\square$

We may also assume that G satisfies the following property (otherwise we can apply Lemma 85):

(**) For any induced path P in G on 6 or 7 vertices, there is a vertex $x \in V(G) \setminus V(P)$ which has both a neighbour and a non-neighbour of the same parity in P.

Let $\mathcal{G}$ denote the subclass of $(K_3, S_{1,2,3}, S_{1,1,2} + P_2, C_7, C_8, P_8)$-free graphs with Properties (*) and (**).

Lemma 88. *Any connected graph $G \in \mathcal{G}$ containing an induced P_6 has chromatic number at most 3 and a 3-colouring of G can be found in polynomial time.*

Proof. Let Q denote the graph obtained from a C_6 by adding a vertex which has exactly one neighbour on the cycle (see Figure 7.2). We split the proof into two cases.

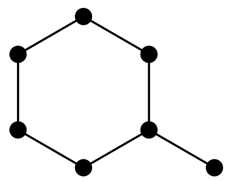


Figure 7.2: The graph Q

Case 1: G contains an induced subgraph isomorphic to Q. Say Q is induced by vertices $a, b, c, d, e, f, g \in V(G)$ where $a - b - c - d - e - f - a$ is a chordless cycle and the only neighbour of g on the cycle is e. The vertices of G outside the set $\{a, b, c\}$ can be partitioned into at most 5 non-empty subsets in the following way:

V_a is the set of vertices adjacent to a and non-adjacent to b and c,

V_b and V_c are defined by analogy with V_a,

V_{ac} is the set of vertices adjacent to a and c and non-adjacent to b,

W is the set of vertices anticomplete to $\{a, b, c\}$.

Note that V_a, V_b, V_c and V_{ac} are independent sets, since G is K_3-free. We will split W into independent sets. We will investigate the possible edges between all these independent sets and finally, we will show how to obtain a 3-colouring of G.

(i) *For any edge uv in $G[W \setminus \{e, g\}]$, at least one of u, v has a neighbour in $\{e, g\}$.* Suppose not. Then since $G[e, d, g, f, a, u, v]$ cannot be isomorphic to $S_{1,1,2} + P_2$, it follows that at least one of u, v is adjacent to one of d, f. Without loss of generality, we may assume that u is adjacent to f. But then $G[f, u, e, g, a, b, c]$ would be an $S_{1,2,3}$, a contradiction.

We may now partition W into two sets W_0 and W_1, where $G[W_1]$ is the connected component of $G[W]$ containing e and g. Notice that $W_0 = W \setminus W_1$ is an independent set (by (i)).

(ii) *For every edge uv in $G[W_1]$, exactly one of u, v has a neighbour in $\{d, f\}$.* This is trivially true for every edge incident to e. Now consider an edge ug in $G[W_1]$, where $u \neq e$. Notice that g is non-adjacent to d, f. If u is non-adjacent to d, f, then $G[e, f, g, u, d, c, b]$ is isomorphic to $S_{1,2,3}$, a contradiction. Thus u is adjacent to at least one of d, f. Now consider an edge uv in $G[W_1]$ such that $u, v \neq e, g$. Since G is (K_3, C_7)-free, at most one of u, v can have a neighbour in $\{d, f\}$. Suppose that u, v are non-adjacent to d, f. From the previous case, we may assume that u, v are non-adjacent to g. It follows from (i) that one of u, v is adjacent to e. Without loss of generality we may assume that u is adjacent to e. But then $G[e, g, u, v, f, a, b]$ would be isomorphic to $S_{1,2,3}$, which is a contradiction.

(iii) *$G[W_1]$ is complete bipartite.* First let us show that every vertex $u \in W_1 \setminus \{e, g\}$ is adjacent to exactly one of e, g. Clearly no vertex can be adjacent to both e and g since G is K_3-free. Now let $u \in W_1 \setminus \{e, g\}$ and suppose that u is non-adjacent to e, g. If u is adjacent to f (resp. d) then $G[f, u, e, g, a, b, c]$ (resp. $G[d, u, e, g, c, b, a]$) is isomorphic to $S_{1,2,3}$, a contradiction. Now let v be a neighbour of u in W_1. It follows from (ii) that v is adjacent to at least one of d, f. We may assume that v is adjacent to f. But now $G[f, e, v, u, a, b, c]$ is isomorphic to $S_{1,2,3}$, a contradiction. Thus every vertex $u \in W_1 \setminus \{e, g\}$ is indeed adjacent to exactly one of e, g. Let $W_1(g)$ be the vertices in W_1 which are adjacent to e and let $W_1(e)$ be the vertices adjacent to g. Notice that $e \in W_1(e)$ and $g \in W_1(g)$. Now we only need to show that $W_1(e)$ is complete to $W_1(g)$. Suppose not. Let $w \in W_1(g)$ and $w' \in W_1(e)$ be non-adjacent. Since g is non-adjacent to d, f, it follows from (ii) that w' is adjacent to at least one of d, f. Without loss of generality we may assume that w' is adjacent to f. But now $G[f, w', e, w, a, b, c]$ is isomorphic to $S_{1,2,3}$, a contradiction.

Notice that since e is adjacent to d, f, (ii) implies that $W_1(g)$ must be anticomplete to $\{d, f\}$ and that every vertex in $W_1(e)$ is adjacent to at least one of d, f.

(iv) *Let $v \in V_a \cup V_c$ with $v \neq d, f$. Then for every edge ww' in $G[W_1]$, exactly one of w, w' is adjacent to v.* Suppose not. Without loss of generality, assume $v \in V_c$, $w \in W_1(e)$ and $w' \in W_1(g)$. But then $G[c, v, b, a, d, w, w']$ is isomorphic to $S_{1,2,3}$ (if $dw \in E(G)$) or to $S_{1,1,2} + P_2$ (if $dw \notin E(G)$), which is a contradiction.

(v) *There exist no two vertices $u, v \in W_1(e)$ such that $uf, vd \in E(G)$ and $ud, vf \notin E(G)$.* Suppose, for contradiction, that two such vertices exist. Notice that $u, v \neq e$. But then $G[d, v, c, b, e, f, u]$ is isomorphic to $S_{1,2,3}$, a contradiction.

Thus either d or f is complete to $W_1(e)$. Without loss of generality, we may assume f is complete to $W_1(e)$. Then by (iii) and (iv) it follows that we may partition V_a into $V_a = V_a^1 \cup V_a^2$ such that V_a^1 is complete to $W_1(e)$ and anticomplete to $W_1(g)$ and V_a^2 is complete to $W_1(g)$ and anticomplete to $W_1(e)$. From (iii) and (iv) it also follows that we may partition V_c into $V_c = V_c^1 \cup V_c^2$ such that every vertex in V_c^1 has a neighbour in $W_1(e)$ and is anticomplete to $W_1(g)$ and every vertex in V_c^2 has a neighbour in $W_1(g)$ and is anticomplete to $W_1(e)$. Since G is K_3-free, V_a^1 must be anticomplete to V_c^1 and V_a^2 must be anticomplete to V_c^2.

(vi) *W_0 is anticomplete to $V_a \cup V_c$.* Let $u \in W_0$ and suppose that u is adjacent to some vertex v in $V_a \cup V_c$. Consider an edge ww' in $G[W_1]$. It follows from (iv) that exactly one vertex of w, w' is adjacent to v. We may assume without loss of generality that w is adjacent to v. But now $G[v, u, w, w', a, b, c]$ is isomorphic to $S_{1,2,3}$, a contradiction.

(vii) *$W_1(g)$ and W_0 have no common neighbours in V_{ac}.* Suppose that $w \in W_1(g)$ and $u \in W_0$ have a common neighbour $v \in V_{ac}$. Since G is K_3-free, e is non-adjacent to v. But then $G[v, u, a, b, w, e, d]$ is isomorphic to $S_{1,2,3}$, a contradiction.

Let X denote the subset of vertices of V_{ac} that have a neighbour in $W_1(g)$ and let Y denote the remaining vertices of V_{ac}. Notice that X is anticomplete to $W_1(e)$ since G is K_3-free. From the above and the fact that G is K_3-free, we conclude that each of the following three sets is independent: $V_a^2 \cup V_c^2 \cup W_1(e) \cup W_0 \cup \{b\} \cup X$, $V_a^1 \cup V_c^1 \cup W_1(g) \cup Y$, $V_b \cup \{a, c\}$. Therefore G is 3-colourable and such a colouring can be found in polynomial time.

Case 2: G contains no induced subgraph isomorphic to Q. Suppose that the vertices a, b, c, d, e, f induce a P_6 with edges $\{ab, bc, cd, de, ef\}$ (we know that G contains an induced P_6). The vertices outside the set $\{b, c, d, e\}$ can be partitioned into at most 8 non-empty sets as follows:

V_b is the set of vertices adjacent to b and non-adjacent to c, d, e,

V_c, V_d, V_e are defined by analogy with V_b,

V_{bd} is the set of vertices adjacent to b and d and non-adjacent to c and e,

V_{ce} and V_{be} are defined by analogy with V_{bd},

W is the set of vertices anticomplete to $\{b, c, d, e\}$.

(i) *V_b is anticomplete to V_e.* Note that $a \in V_b$ and $f \in V_e$. We know that $af \notin E(G)$. Suppose a has a neighbour $u \in V_e \setminus \{f\}$. Then $G[a, b, c, d, e, u, f]$ is isomorphic to Q, a contradiction. Therefore a is anticomplete to V_e. Now suppose that there exist two adjacent vertices $u \in V_b \setminus \{a\}, v \in V_e$. Then $G[b, c, d, e, v, u, a]$ is isomorphic to Q. This contradiction shows that V_b is anticomplete to V_e.

(ii) *Every vertex in W is either complete to V_b (resp. V_e) or anticomplete to V_b (resp. V_e).* Suppose there exists a vertex $w \in W$ which is adjacent to some vertex $u \in V_b$ and non-adjacent to some other vertex $v \in V_b$. Then $G[b, v, u, w, c, d, e]$ is isomorphic to $S_{1,2,3}$, a contradiction. Thus the claim holds for V_b and by symmetry we conclude that it holds for V_e as well.

(iii) *No vertex in W is complete to both V_b and V_e.* Suppose a vertex $w \in W$ is complete to $V_b \cup V_e$. Then $G[a, b, c, d, e, f, w]$ is isomorphic to C_7, a contradiction.

It follows from the above that we may partition W into three sets W_b, W_e, W_0, where W_b is complete to V_b and anticomplete to V_e, W_e is complete to V_e and anticomplete to V_b, and W_0 is anticomplete to $V_b \cup V_e$. Notice that W_b and W_e are both independent sets.

(iv) *At most one of W_b, W_e is non-empty.* Indeed if both W_b and W_e are non-empty, say $u \in W_b$ and $v \in W_e$, then $G[u, a, b, c, d, e, f, v]$ is either isomorphic to C_8 or P_8, a contradiction.

It follows from (iv) that we may assume without loss of generality that $W_e = \emptyset$. Thus W is anticomplete to V_e. Furthermore, $|W_b| \leq 1$, since if $u, v \in W_b$, then $G[a, u, v, b, c, e, f]$ is isomorphic to $S_{1,1,2} + P_2$, a contradiction.

(v) *W is an independent set.* Suppose W contains an edge uv and that $u \in W_b$. Since G is K_3-free, it follows that v is non-adjacent to a. But now $G[v, u, a, b, c, d, e, f]$ is isomorphic to P_8. This contradiction shows that neither u nor v has neighbours in V_b, hence $u, v \in W_0$.

We let P denote either the induced path $P_6 = \{ab, bc, cd, de, ef\}$ (if $W_b = \emptyset$) or the induced path $P_7 = \{ya, ab, bc, cd, de, ef\}$ (if $W_b = \{y\}$). We label the vertices of P by natural numbers $1, 2, \ldots, 6$ or $1, 2, \ldots, 7$ and let k be the number of vertices in P.

Suppose a vertex z outside P has a neighbour in P. Then it must be adjacent to a vertex i of degree 2 in P. Note that W_0 and P are anticomplete, so $z \neq u, v$.

This implies that z is adjacent to $i-2$ (if $i > 2$), since otherwise $G[i, i+1, i-1, i-2, z, u, v]$ induces either an $S_{1,2,3}$ (if z has a neighbour in $\{u, v\}$) or an $S_{1,1,2} + P_2$ (if z has no neighbour in $\{u, v\}$). Similarly z must be adjacent to $i+2$ if $i < k-1$. As a result z is adjacent to all vertices of the same parity in P. Therefore, if W is not an independent set, then G does not have Property (**). This contradiction implies that W is an independent set.

(vi) *W_b is anticomplete to V_d.* Let $W_b = \{y\}$. Suppose that y is adjacent to $u \in V_d$. Then $G[a, b, c, d, u, y, e]$ is isomorphic to Q, a contradiction.

(vii) *W_0 is anticomplete to $V_c \cup V_d$.* By symmetry it is enough to show that W_0 is anticomplete to V_c. Suppose that a vertex $w \in W_0$ is adjacent to some vertex

$u \in V_c$. Then u must be adjacent to f otherwise $G[c, b, u, w, d, e, f]$ would be isomorphic to $S_{1,2,3}$, a contradiction. Now we claim that u is adjacent to a. Suppose not, then $G[u, w, f, e, c, b, a]$ would be isomorphic to $S_{1,2,3}$, a contradiction. But now $G[u, w, a, b, f, e, d]$ is isomorphic to $S_{1,2,3}$, a contradiction.

(viii) *One of W_b, V_{be} is empty.* Indeed, suppose $W_b = \{y\}$ and $u \in V_{be}$. If y is non-adjacent to u then $G[b, c, a, y, u, e, f]$ is isomorphic to $S_{1,2,3}$, a contradiction. On the other hand, if y is adjacent to u, then $G[e, f, d, c, u, y, a]$ is isomorphic to $S_{1,2,3}$, a contradiction.

(ix) *If $W_b = \emptyset$, then G is 3-colourable.* First, suppose that W_0 is anticomplete to V_{be}. Then it is easy to see that the following are independent sets: $W_0 \cup V_b \cup V_e \cup V_{be} \cup \{c\}$, $V_{bd} \cup V_d \cup \{e\}$, $\{b, d\} \cup V_{ce} \cup V_c$. So we may now assume that there exists a vertex $w \in W_0$ which has a neighbour $v \in V_{be}$. We claim that v must be complete to $V_c \cup V_d$. Suppose that v is non-adjacent to some vertex $u \in V_c$. Then f is adjacent to u, since otherwise $G[v, w, e, f, b, c, u]$ would be isomorphic to $S_{1,2,3}$, a contradiction. But now $G[c, d, u, f, b, v, w]$ is isomorphic to $S_{1,2,3}$, a contradiction. Thus v is complete to V_c and by symmetry we conclude that v is complete to V_d as well. Hence V_c and V_d are anticomplete. Now we obtain a 3-colouring as follows: $V_b \cup V_{be} \cup V_{bd} \cup \{c\}$, $\{b, e\} \cup V_c \cup V_d \cup W_0$, $\{d\} \cup V_e \cup V_{ce}$.

It follows from (ix) that we may now assume that $W_b = \{y\}$ and hence $V_{be} = \emptyset$. We claim that V_e is complete to V_d. Suppose some vertex $u \in V_d$ is non-adjacent to some vertex $v \in V_e$. Then u must be adjacent to a, otherwise $G[d, u, e, v, c, b, a]$ is isomorphic to $S_{1,2,3}$, a contradiction. But now $G[d, c, e, v, u, a, y]$ is isomorphic to $S_{1,2,3}$, a contradiction. Thus V_e is complete to V_d. This implies that V_b is anticomplete to V_d. Indeed if a vertex $u \in V_b$ is adjacent to some vertex $v \in V_d$, then $G[u, y, b, c, v, f, e]$ is isomorphic to $S_{1,2,3}$, a contradiction. Now we obtain a 3-colouring as follows: $V_b \cup V_{bd} \cup V_d \cup \{c, e\}$, $\{b, d\} \cup V_e \cup W$, $V_{ce} \cup V_c$.

This completes the proof that any connected graph $G \in \mathcal{G}$ containing an induced P_6 has chromatic number at most 3. From the above, it is easy to see that a 3-colouring of G can be found in polynomial time. □

Theorem 89. *The* VERTEX COLOURING *problem is solvable in polynomial time in the class of $(K_3, S_{1,2,3}, S_{1,1,2} + P_2)$-free graphs.*

Proof. Since we can solve the problem component-wise in G, we may assume that G is connected. It follows from Lemmas 75, 82, 83 and 84 that the problem reduces to

3-Colourability of $(K_3, S_{1,2,3}, S_{1,1,2} + P_2)$-free graphs which contain no odd induced cycle of length at least 7. Also, we only need to consider graphs that satisfy Property (*). Lemmas 85, 86 and 87 further reduce the problem in polynomial time to those graphs that contain no induced paths or induced even cycles of length at least 8. The reduction is as follows:

- Check if G contains a P_8 or C_8. If G contains a C_8 apply Lemmas 85 and 86. If G contains a P_8 extend it to a maximal (with respect to set inclusion) induced path P. This can obviously be done in polynomial time. If there is a vertex which creates a cycle with P, by Lemma 86, we can apply Lemma 85. Otherwise, every vertex of G which has a neighbour on P must be adjacent to a vertex of degree 2 in P, in which case Lemma 87 tells us we can apply Lemma 85.

The above procedure further reduces the problem to 3-Colourability of $(K_3, S_{1,2,3}, S_{1,1,2} + P_2)$-free graphs with Property (*) that are (C_7, C_8, P_8)-free. Finally, if G does not satisfy Property (**), we can find a suitable path on 6 or 7 vertices and apply Lemma 85. We may therefore assume G satisfies Property (**).

Note that all of the above reductions work in polynomial time and either solve the 3-Colourability problem or delete vertices from the graph, so at most $|V(G)|$ such reductions can be applied. We may now assume that G is a connected $(K_3, S_{1,2,3}, S_{1,1,2} + P_2, C_7, C_8, P_8)$-free graph satisfying Properties (*) and (**), i.e. $G \in \mathcal{G}$.

Now if G is P_6-free, we can solve the 3-Colourability problem in polynomial time by Theorem 72 and if G is not P_6-free, we can solve the problem in polynomial time using Lemma 88. This completes the proof. $\square$

7.6 Further results

In this section we prove a few additional results. The first two results deal with graph classes $Free(K_3, F)$ where F is a "big" forest of simple structure.

Theorem 90. *For every fixed m, the* Vertex Colouring *problem is polynomial-time solvable in the class $Free(K_3, mK_2)$.*

Proof. Obviously, if a graph G is k-colourable, then it admits a k-colouring in which one of the colour classes is a maximal independent set.

It is known that for every fixed m the number of maximal independent sets in the class $Free(mK_2)$ is bounded by a polynomial [Balas and Yu, 1989] and all of them can be found in polynomial time [Tsukiyama et al., 1977]. Therefore, given an mK_2-free graph G, we can solve the 3-Colourability problem for G by generating all maximal independent sets and solving 2-Colourability for the remaining vertices of the graph. Then by induction on k, we conclude that for any fixed k the k-Colourability problem can be solved in the class $Free(mK_2)$ in polynomial time. Since the chromatic number of (K_3, mK_2)-free graphs is bounded by $2m-2$ (see e.g. [Brandt, 2002a]), the Vertex Colouring problem is polynomial-time solvable in the class $Free(K_3, mK_2)$ for any fixed m. □

Theorem 91. *For every fixed m, the* Vertex Colouring *problem is polynomial-time solvable in the class $Free(K_3, P_3 + mK_1)$.*

Proof. To prove the theorem, we will show that for any fixed m, graphs in the class $Free(K_3, P_3+mK_1)$ are either bounded in size, or they are 3-colourable and a 3-colouring can be found in polynomial time.

Let G be a $(K_3, P_3 + mK_1)$-free graph. We start by finding a maximum independent set in G. For each fixed m, this problem is solvable in polynomial time, which can easily be seen by induction on m. Let S be a maximum independent set in G. Let R denote the remaining vertices of G, i.e. $R = V(G) - S$. We may assume that R contains an induced odd cycle $C = v_1 - v_2 - \cdots - v_p - v_1$ with $p \geq 5$. Since S is a maximum independent set, each vertex of C has at least one neighbour in S. Let us call a vertex $v_i \in V(C)$ *strong* if it has at least 2 neighbours in S and *weak* otherwise. Since C is an odd cycle, it has either two consecutive weak vertices or two consecutive strong vertices.

If C has two consecutive weak vertices, say v_1, v_2, then jointly they are adjacent to two vertices of S, say v_1 is adjacent to s_1, and v_2 is adjacent to s_2, and therefore, they have $|S| - 2$ common non-neighbours in S. If $|S| - 2 \geq m$, then s_1, v_1, v_2 together with m vertices in $S \setminus \{s_1, s_2\}$ induce a subgraph isomorphic to $P_3 + mK_1$, a contradiction. Therefore $|S| < m + 2$. But then the number of vertices in G is bounded by the Ramsey number $R(3, m + 2)$, since G is K_3-free and contains no independent set of size $m + 2$.

Now suppose C has two consecutive strong vertices, say v_1, v_2. Since the graph is $(P_3 + mK_1)$-free, every strong vertex has at most $m - 1$ non-neighbours in S, and since the graph is K_3-free, consecutive vertices of C cannot have common neighbours. Therefore each of v_1 and v_2 has at most $m - 1$ neighbours in S. But then $|S| < 2m - 1$ and hence the number of vertices of G is bounded by the Ramsey number $R(3, 2m - 1)$ by the same argument as before.

Thus, if R has an odd cycle, then the number of vertices in G is bounded by a constant. If R has no odd cycles, then $G[R]$ is bipartite, and hence G is 3-colourable. Finding a maximum independent set in a (P_3+mK_1)-free graph can be done in polynomial time, so any (K_3, P_3+mK_1)-free graph is either bounded in size, or can be 3-coloured in this way in polynomial time. Thus VERTEX COLOURING of (K_3, P_3+mK_1)-free graphs can be solved in polynomial time. $\square$

We now present an alternative proof of the fact that every (K_3, H)-free graph is 3-colourable which is much shorter than the original proof in [Randerath, 2004].

Theorem 92. *Every (K_3, H)-free graph is 3-colourable and a 3-colouring can be found in polynomial time.*

Proof. Let G be a (K_3, H)-free graph and S be any maximal (with respect to set inclusion) independent set in G. We assume that S admits no augmenting $K_{1,2}$ (i.e. a triple x, y, z such that x and y are non-adjacent vertices outside S with $N(x) \cap S = N(y) \cap S = \{z\}$), since finding an augmenting $K_{1,2}$ can be done in polynomial time. (If such an augmenting $K_{1,2}$ exists, we can just replace S by $\{x, y\} \cup S \setminus \{z\}$, which increases the size of S.)

Assume that the graph $G[V \setminus S]$ is not bipartite, and let vertices $x_1, \dots, x_k$ induce a cycle C of odd length $k \geq 5$ in $G[V \setminus S]$. By maximality of S, every vertex outside S has a neighbour in S.

Suppose that each vertex of C has exactly one neighbour in S, and let $y_2 \in S$ and $y_3 \in S$ be the neighbours of x_2 and x_3, respectively. Then $x_1, x_2, x_3, x_4, y_2, y_3$ induce a copy of the graph H (by lack of triangles and augmenting $K_{1,2}$s). Thus, C must contain vertices with at least two neighbours in S. Assume without loss of generality that x_2 is of this type. If C has two consecutive vertices each of which has at least two neighbours in S, then an induced H can be easily found. Therefore, each of x_1 and x_3 has exactly one neighbour in S. If $y_2 \in S$ is a neighbour of x_2 and $y_3 \in S$ is a neighbour of x_3, then x_4 is adjacent to y_2, since otherwise $x_1, x_2, y_2, x_3, y_3, x_4$ would induce a copy of H. Therefore, $N(x_2) \cap S \subseteq N(x_4) \cap S$, and by symmetry, $N(x_4) \cap S \subseteq N(x_2) \cap S$, i.e. x_2 and x_4 have the same neighbourhood in S. This in turn implies that x_5 has exactly one neighbour in S. Continuing inductively, we conclude that the even-indexed vertices of C have the same neighbourhood in S consisting of at least two vertices, and each of the odd-indexed vertices of C has exactly one neighbour in S. But then $x_1, x_2, x_k, x_{k-1}, y_1, y_k$ induce a copy of the graph H, where $y_1 \in S$ and $y_k \in S$ are the neighbours of x_1 and x_k, respectively. $\square$

The VERTEX COLOURING problem is equivalent to the problem of finding a partition of the complement of G into minimum number of cliques, which we call CLIQUE

Partition. Vertex Colouring is known to be intractable not just from the classical, but also from the parameterized point of view (see e.g. [Downey and Fellows, 1999; Flum and Grohe, 2006]). We prove that in the class of complements of line graphs of triangle-free graphs, the Vertex Colouring problem is NP-hard but fixed-parameter tractable.

Theorem 93. *In the class of complements of line graphs of triangle-free graphs,* Vertex Colouring *is NP-hard and fixed-parameter tractable.*

Proof. The reduction is from Minimum Vertex Cover in triangle-free graphs, which is an NP-hard [Poljak, 1974] problem in this class and is fixed-parameter tractable (see e.g. [Downey and Fellows, 1999; Flum and Grohe, 2006]). Let G be a triangle-free graph and let $H = L(G)$ be its line graph. It is not difficult to see that every clique K in H corresponds to a set of edges of G incident to the same vertex, which we denote $v(K)$. Let $K^1, \ldots, K^p$ be a set of cliques that partition $V(H)$. Then $\{v(K^1), \ldots, v(K^p)\}$ is a vertex cover of G. This cover is minimum if and only if $K^1, \ldots, K^p$ is a minimum clique partition of $V(H)$. Since Minimum Vertex Cover is fixed-parameter tractable, we conclude that Clique Partition is fixed-parameter tractable in the class of line graphs of triangle-free graphs. Therefore, Vertex Colouring is fixed-parameter tractable in the class of complements of line graphs of triangle-free graphs. By the same argument, the problem is NP-hard in this class. □

7.7 Conclusion

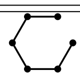
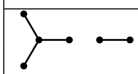
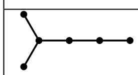
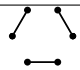
Graph	Graph Name	Complexity	Reference
	P_6	P	Theorem 72
	$K_{1,3} + P_2$	P	Theorem 81
	$S_{1,1,3}$	P	Theorem 81
	$3P_2$	P	Theorem 90

Table 7.3: Forests F on six vertices, none of which is isolated, for which the complexity of Vertex Colouring in the class $\text{Free}(K_3, F)$ is contributed in this chapter.

In this chapter we studied the complexity of the Vertex Colouring problem in subclasses of triangle-free graphs obtained by forbidding forests and proved polynomial-time solvability of the problem in many classes of this type. In particular our contribution, combined with some previously known results listed in Table 7.1, provides a complete description of the complexity status of the problem in subclasses of triangle-free graphs obtained by forbidding a forest with at most 6 vertices (Tables 7.2 and 7.3 summarise results of this type obtained in the present chapter). Very little is known about the status of the problem in subclasses of triangle-free graphs defined by forbidding forests with more than 6 vertices, and this creates a challenging research direction.

One more natural direction of research is investigation of the problem in extensions of triangle-free graphs. Let us observe that all results on triangle-free graphs can be extended, with no extra work, to so-called paw-free graphs, where a paw is the graph obtained from a triangle by adding a pendant edge. This follows from two facts: first, the problem can obviously be reduced to connected graphs, and second, according to [Olariu, 1988], a connected paw-free graph is either complete multipartite (i.e. $\overline{P}_3$-free), in which case the problem is trivial, or triangle-free.

Further extensions of these classes make the problem much harder. For instance, by adding a pendant edge to each vertex of a triangle, we obtain a graph known in the literature as a net, and according to [Schindl, 2005] the problem is NP-hard even for $(net, 2K_2)$-free graphs and $(net, 4K_1)$-free graphs. An interesting intermediate class between paw-free and net-free graphs is the class of bull-free graphs, where a bull is the graph obtained by adding a pendant edge to two vertices of a triangle. Recently, the class of bull-free graphs received much attention in the literature (see e.g. [Chudnovsky, 2012; Chudnovsky and Safra, 2008; de Figueiredo and Maffray, 2004; Lévêque and Maffray, 2008]). In particular, [Chudnovsky, 2012] provides a structural characterisation of bull-free graphs which may be helpful in designing algorithms for various graph problems, including the vertex colouring problem in subclasses of bull-free graphs.

Chapter 8

Dynamic Edge-Choosability

8.1 Introduction

We define a *dynamic edge-colouring* to be a (possibly improper) colouring of the edges of a graph G such that the set of edges incident to any vertex of degree greater than one contains at least two distinct colours (i.e. it is not monochromatic). Let the *dynamic edge-choice number* $ch_2^*(G)$ denote the smallest integer k such that if every edge is given a list of k colours then G has a dynamic edge-colouring with the additional property that every edge is assigned a colour from its list. This parameter was introduced in [Esperet, 2010] to help construct counterexamples to a conjecture proposed in [Akbari et al., 2009].

Using a greedy algorithm, it can be shown that $ch_2^*(G) \leq 3$ for all graphs G [Esperet, 2010]. Indeed, given a graph G and a list assignment of 3 colours to each edge of G, we proceed as follows. Taking each edge uv in turn, pick a colour for uv that is different from one of the colours already-chosen for the edges incident with u and from one of the colours already-chosen for the edges incident with v (if such already-coloured edges exist, otherwise pick a colour for uv arbitrarily). Since the list for uv contains three colours, it is always possible to do this and the resulting colouring has the property that every vertex of degree greater than one has at least two distinct colours among the edges incident with it. It is easy to see that $ch_2^*(G)$ is 0 or 1 if and only if the graph G has maximum degree 0 or 1, respectively.

Characterising the graphs G with $ch_2^*(G) = 2$ was an open problem [Esperet, 2010]. We solve this problem in the next section.

8.2 Graphs with Dynamic Edge-choice Number 2

In this section we show that the graphs with dynamic edge-choice number at most two are precisely those that have no component which is an odd cycle. To help us show this, we start by introducing a useful graph transformation.

Let G be a graph and let X be a set of vertices in G. For each $e \in E(G)$, let $L(e) \subset \mathbb{N}$ be the list of colours assigned to the edge e. We define the graph G_X as follows: Start with the graph $G[V(G) \setminus X]$, then for each edge $xy \in E(G)$ such that $x \in X, y \in V(G) \setminus X$, create a new vertex x_y whose only neighbour in G_X is y (see Figures 8.1 and 8.2 for examples). We associate these new edges $x_y y$ with the original edges xy in the sense that we let $L(x_y y) = L(xy)$ and if we choose to colour the edge $x_y y$ with a colour c in its list, we also implicitly do so for xy. (Similarly, we associate the remaining edges in G_X with the corresponding edges of G in the obvious way.)

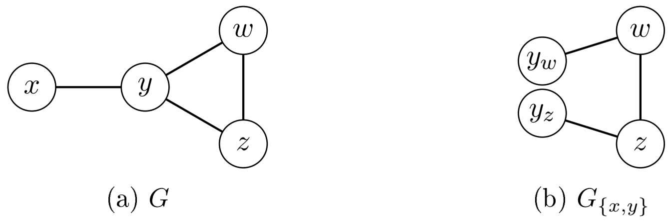


Figure 8.1: An example of G and G_X when X contains a vertex of degree 1

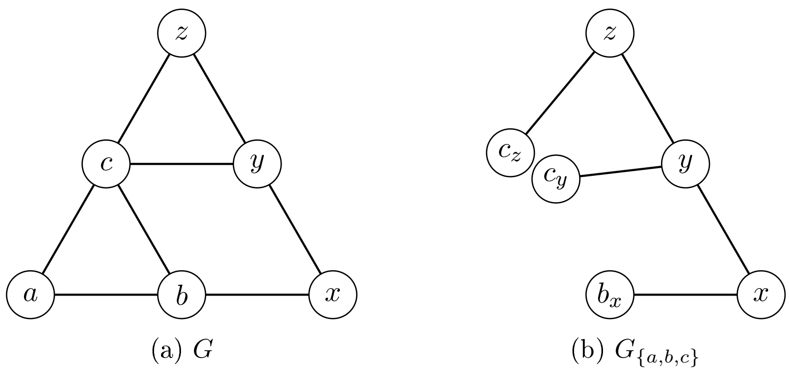


Figure 8.2: An example of G and G_X when X is a cycle

Note that, by construction, if G is connected and X is non-empty, then every component of G_X must contain a vertex of degree 1. If $G[X]$ contains an edge then G_X contains less edges than G. Also, for any vertex $y \in V(G) \setminus X$, the degree of y in G and

G_X are equal.

If we colour some (but not necessarily all) of the edges of a graph G, we say that a vertex x in G has the *dynamic property* if it is incident to two edges which have been coloured with different colours or if it of degree at most 1.

We now prove a lemma that deals with a special case of the problem.

Lemma 94. *Let G be a connected graph such that every edge e has a list of colours $L(e)$ of size 2 assigned to it. If G contains a vertex x of degree 1, then G has a dynamic edge-colouring respecting the lists and this colouring can be found in $O(n+m)$ time.*

Proof. We prove the lemma by induction on the number of edges in G. Suppose that for some k, the lemma holds for all connected graphs G on at most k edges containing a vertex of degree 1.

Let G be a connected graph on $k+1$ edges, containing a vertex x of degree 1. Let y be the unique neighbour of x in G and let $X = \{x, y\}$. Note that since G is connected, every component of G_X must contain a vertex of degree 1. Also, every component of G_X can have at most k edges, so we can find a dynamic edge-colouring of G_X respecting the lists (since the problem can be solved componentwise). Now, if we consider the associated colouring on G, we find that every vertex in $V(G) \setminus X$ has the dynamic property.

If G_X is edgeless, then G must consist of just the vertices x and y, in which case colouring xy with either colour in its list yields a dynamic edge-colouring of G. If G_X contains an edge (see Figure 8.1 for an example), then by the connectivity of G, y must have a neighbour z in G. This means that the vertex y_z is present in G_X and adjacent to z in G_X. This means that in the associated colouring of G, the edge yz must have an assigned colour. We now colour the edge xy with a colour from its list, different from that of yz. This causes both x and y to have the dynamic property, yielding a dynamic edge-colouring for the whole graph G.

By considering recursive transformations of the form G to G_X and keeping track of the vertices of degree 1 that are created, we can find a dynamic edge-colouring respecting the lists in $O(n+m)$ time. $\square$

We are now ready to characterise the graphs with dynamic edge-choice number 2.

Theorem 95. *A graph G has $ch_2^*(G) \leq 2$ if and only if G has no component that is an odd cycle. Furthermore, if every edge e in G is assigned a list $L(e) \subset \mathbb{N}$ of two colours, then a dynamic edge-colouring respecting these lists can be found in $O(n+m)$ time, if such an edge-colouring exists.*

Proof. Suppose that each edge e of G is assigned a list $L(e)$ of two colours. Since the problem can be solved componentwise, we may assume that G is connected and contains at least one edge. We may also assume that G contains no vertices of degree 1, otherwise Lemma 94 can be applied.

Every vertex in G must be of degree at least 2, so G must contain a chordless cycle C. Since G is connected, either G_C is edgeless (in which case the graph G consists only of the cycle C) or every component of G_C must contain a vertex of degree 1 (see Figure 8.2 for an example). Therefore, by Lemma 94, we can find a dynamic edge-colouring of G_C which respects the lists. Note that in the partial edge-colouring of G associated with this colouring, every vertex of $V(G) \setminus C$ has the dynamic property in G.

Let $e_1, \ldots, e_k$ be the vertices in the cycle C in order. We now have three cases to consider:

- *If not all the edges of C have the same list of colours*, say e_1 has a colour c in its list which is not in the list of e_k. We can colour e_1 with colour c and then for $i = 2, \ldots, k$ colour e_i with a colour different from that of e_{i-1}. Combining this with the colouring for G_C, we obtain a colouring for G in which every vertex has the dynamic property.

- *If all the edges of C have the same list and the cycle is of even length*, colour the edges of the cycle by alternating between the two colours. Again, combining this with the colouring for G_C ensures that every vertex in G has the dynamic property.

- *If C is an odd cycle and every edge of C has the same list of colours.* If G_C is edgeless, then C must contain all the vertices of G. In this case there is no solution to the problem since it is impossible to colour the edges of C with alternating colours. If G_C contains an edge, then there must be an edge of G incident with a vertex of C, say xy is incident with the edges e_1 and e_k (where $x \in C, y \notin C$). In this case, we colour G_C as before, so that every vertex outside of C has the dynamic property in G. Let c_1 be a colour from the list assigned to the edge e_1 which is different from the colour of xy. Now set e_1 and e_k to be of colour c_1 and then for $i = 2, \ldots, k-1$ colour e_i to be different from e_{i-1} (i.e. alternate colours). This will ensure that all the vertices of C also have the dynamic property.

Thus the only conditions under which a (not necessarily connected) graph G can fail to have a dynamic edge-colouring is in the case where it has a component which is an odd cycle and where every edge in this cycle is assigned the same list of two colours.

To summarise, we can solve the problem by the following algorithm:

Algorithm CHOOSE(G)
Input: A graph G and lists $L(e) \subset \mathbb{N}$ of size 2, for every $e \in E(G)$
Output: A dynamic edge-colouring of G respecting the lists, or "IMPOSSIBLE" if no such edge-colouring exists

1. Find all components on at least 2 vertices in G
2. If G contains a component which is an odd cycle and every edge in this cycle has the same list return "IMPOSSIBLE"
3. For every component H containing a vertex of degree 1, apply Lemma 94 to it
4. For every component H not containing a vertex of degree 1, find a chordless cycle C in the component, apply Lemma 94 to H_C and colour the edges of C so that all the vertices of C have the dynamic property.
5. Return the resulting colouring

Note that a graph can be split into components in $O(n+m)$ time, a chordless cycle or a vertex of degree 1 can be found in $O(n+m)$ time and Lemma 94 can be applied in $O(n+m)$ time. Thus the above algorithm runs in $O(n+m)$ time. □

8.3 Conclusion

In this chapter we fully characterised the graphs with every possible dynamic edge-choice number and gave a linear-time algorithm to find a dynamic edge-colouring respecting the lists assigned to each edge of the input graph when each such list is of size 2 (if such an edge-colouring exists).

Chapter 9

Conclusion

In this thesis we started with the questions "How far must we restrict the structure of our graph to be able to solve our problem efficiently?" We studied several problems which are computationally hard to solve. We found various non-trivial classes of graphs where these problems can be solved efficiently and discussed various techniques (such as augmenting graphs and modular decomposition) that can be used to achieve this goal.

It is often the case that as problems are solved, they suggest new problems that could be investigated. I hope that the results in this thesis will stimulate more work in the field of algorithmic graph theory. I conclude by listing the four open problems mentioned in the thesis that I would most like to see solved:

1. The complexity of the Maximum Independent Set problem in P_5-free graphs.
2. The complexity of the Stable-$\mathcal{M}_3$ problem.
3. The complexity the Efficient Edge Domination problem in $S_{1,2,3}$-free graphs.
4. The parameterized complexity of the Weighted Independent Set problem in C_4-free graphs.

Bibliography

S. Akbari, M. Ghanbari, and S. Jahanbekam. On the list dynamic coloring of graphs. *Discrete Applied Mathematics*, 157(14):3005–3007, 2009.

V.E. Alekseev. On lower layers of a lattice of hereditary classes of graphs (in russian). *Diskretnyĭ Analiz i Issledovanie Operatsiĭ, Ser. 1*, 4:3–12, 1997.

V.E. Alekseev, A. Farrugia, and V.V. Lozin. New results on generalized graph coloring. *Discrete Mathematics and Theoretical Computer Science*, 6(2):215–222, 2004.

P. Alimonti and V. Kann. Hardness of approximating problems on cubic graphs. *Lecture Notes in Computer Science*, 1203:288–298, 1997.

H. Assiyatun. Maximum induced matchings of random regular graphs. *Lecture Notes in Computer Science*, 3330:44–57, 2005.

G. Ausiello, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, and V. Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties.* Springer-Verlag, 1st edition, 1999.

E. Balas and C.S. Yu. On graphs with polynomially solvable maximum-weight clique problem. *Networks*, 19(2):247–253, 1989.

J. Balogh, B. Bollobás, and D. Weinreich. The speed of hereditary properties of graphs. *Journal of Combinatorial Theory, Series B*, 79(2):131–156, 2000.

C. Berge. Two theorems in graph theory. *Proceedings of the National Academy of Sciences of the United States of America*, 43(9):842–844, 1957.

R. Boliac and V.V. Lozin. An augmenting graph approach to the stable set problem in P_5-free graphs. *Discrete Applied Mathematics*, 131(3):567–575, 2003.

A. Brandstädt and C.T. Hoàng. On clique separators, nearly chordal graphs, and the maximum weight stable set problem. *Theoretical Computer Science*, 389(1–2):295–306, 2007.

A. Brandstädt and C.T. Hoàng. Maximum induced matchings for chordal graphs in linear time. *Algorithmica*, 52:440–447, 2008.

A. Brandstädt and R. Mosca. Dominating induced matchings for P_7-free graphs in linear time. *CoRR*, abs/1106.2772, 2011a.

A. Brandstädt and R. Mosca. Dominating induced matchings for P_7-free graphs in linear time. *Lecture Notes in Computer Science*, 7074:100–109, 2011b.

A. Brandstädt, V.B. Le, and T. Szymczak. The complexity of some problems related to graph 3-colorability. *Discrete Applied Mathematics*, 89(1–3):59–73, 1998.

A. Brandstädt, V.B. Le, and J.P. Spinrad. *Graph classes: a survey.* SIAM, 1999.

A. Brandstädt, P.L. Hammer, V.B. Le, and V.V. Lozin. Bisplit graphs. *Discrete Mathematics*, 299(1–3):11–32, 2005.

A. Brandstädt, T. Klembt, and S. Mahfud. P_6- and triangle-free graphs revisited: structure and bounded clique-width. *Discrete Mathematics and Theoretical Computer Science*, 8(1):173–188, 2006.

A. Brandstädt, E.M. Eschen, and R. Sritharan. The induced matching and chain subgraph cover problems for convex bipartite graphs. *Theoretical Computer Science*, 381 (1–3):260–265, 2007.

A. Brandstädt, C. Hundt, and R. Nevries. Efficient edge domination on hole-free graphs in polynomial time. *Lecture Notes in Computer Science*, 6034:650–661, 2010.

S. Brandt. Triangle-free graphs and forbidden subgraphs. *Discrete Applied Mathematics*, 120(1–3):25–33, 2002a.

S. Brandt. A 4-colour problem for dense triangle-free graphs. *Discrete Mathematics*, 251 (1–3):33–46, 2002b.

H. Broersma, F.V. Fomin, P.A. Golovach, and D. Paulusma. Three complexity results on coloring P_k-free graphs. *Lecture Notes in Computer Science*, 5874:95–104, 2009.

H. Broersma, P.A. Golovach, D. Paulusma, and J. Song. Narrowing down the gap on the complexity of coloring P_k-free graphs. *Lecture Notes in Computer Science*, 6410: 63–74, 2010.

H. Broersma, P.A. Golovach, D. Paulusma, and J. Song. Determining the chromatic number of triangle-free p_3-free graphs in polynomial time. *Theoretical Computer Science*, 423:1–10, 2012.

L. Cai and D. Corneil. A generalization of perfect graphs–i-perfect graphs. *Journal of Graph Theory*, 23(1):87–103, 1996.

K. Cameron. Induced matchings. *Discrete Applied Mathematics*, 24(1–3):97–102, 1989.

K. Cameron. Induced matchings in intersection graphs. *Discrete Mathematics*, 278(1–3): 1–9, 2004.

K. Cameron, R. Sritharan, and Y. Tang. Finding a maximum induced matching in weakly chordal graphs. *Discrete Mathematics*, 266(1–3):133–142, 2003.

D.M. Cardoso and V.V. Lozin. Dominating induced matchings. *Lecture Notes in Computer Science*, 5420:77–86, 2009.

D.M. Cardoso, J.O. Cerdeira, C. Delorme, and P.C. Silva. Efficient edge domination in regular graphs. *Discrete Applied Mathematics*, 156(15):3060–3065, 2008.

D.M. Cardoso, N. Korpelainen, and V.V. Lozin. On the complexity of the dominating induced matching problem in hereditary classes of graphs. *Discrete Applied Mathematics*, 159(7):521–531, 2011.

J-M. Chang. Induced matchings in asteroidal triple-free graphs. *Discrete Applied Mathematics*, 132(1–3):67–78, 2003.

M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation*, 206(11):1264–1275, 2008.

M. Chudnovsky. The structure of bull-free graphs II and III–a summary. *Journal of Combinatorial Theory, Series B*, 102(1):252–282, 2012.

M. Chudnovsky and S. Safra. The Erdős–Hajnal conjecture for bull-free graphs. *Journal of Combinatorial Theory, Series B*, 98(6):1301–1310, 2008.

M. Chudnovsky and P. Seymour. Claw-free graphs. V. global structure. *Journal of Combinatorial Theory, Series B*, 98(6):1373–1410, 2008.

D.G. Corneil, H. Lerchs, and L.S. Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3(3):163–174, 1981.

B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1–3):77–114, 2000.

B. Courcelle, J.A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33:125–150, 2000.

K. Dabrowski, V. Lozin, R. Raman, and B. Ries. Colouring vertices of triangle-free graphs. *Lecture Notes in Computer Science*, 6410:184–195, 2010.

K. Dabrowski, V. Lozin, H. Müller, and D. Rautenbach. Parameterized algorithms for the independent set problem in some hereditary graph classes. *Lecture Notes in Computer Science*, 6460:1–9, 2011.

K. Dabrowski, V. Lozin, H. Müller, and D. Rautenbach. Parameterized complexity of the weighted independent set problem beyond graphs of bounded clique number. *Journal of Discrete Algorithms*, 14:207–213, 2012a.

K. Dabrowski, V. Lozin, and J. Stacho. Stable-π partitions of graphs. *(submitted)*, 2012b.

K. Dabrowski, V.V. Lozin, and V. Zamaraev. On factorial properties of chordal bipartite graphs. *Discrete Mathematics*, 312(16):2457–2465, 2012c.

K. Dabrowski, M. Demange, and V. Lozin. New results on maximum induced matchings in bipartite graphs and beyond (to appear). *Theoretical Computer Science*, 2013.

K.K. Dabrowski, V. Lozin, R. Raman, and B. Ries. Colouring vertices of triangle-free graphs without forests. *Discrete Mathematics*, 312(7):1372–1385, 2012d.

D.P. Dailey. Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete. *Discrete Mathematics*, 30(3):289–293, 1980.

C.M.H. de Figueiredo and F. Maffray. Optimizing bull-free perfect graphs. *SIAM Journal on Discrete Mathematics*, 18(2):226–240, 2004.

E.D. Demaine, M. Hajiaghayi, and D.M. Thilikos. Exponential speedup of fixed-parameter algorithms for classes of graphs excluding single-crossing graphs as minors. *Algorithmica*, 41:245–267, 2005.

M. Demange, T. Ekim, and D. de Werra. Partitioning cographs into cliques and stable sets. *Discrete Optimization*, 2(2):145–153, 2005.

R.G. Downey and M.R. Fellows. *Parameterized complexity.* Monographs in computer science. Springer, 1999.

W. Duckworth, N.C. Wormald, and M. Zito. Maximum induced matchings of random cubic graphs. *Journal of Computational and Applied Mathematics*, 142(1):39–50, 2002.

W. Duckworth, N.C. Wormald, and M. Zito. A PTAS for the sparsest 2-spanner of 4-connected planar triangulations. *Journal of Discrete Algorithms*, 1(1):67–76, 2003.

W. Duckworth, D.F. Manlove, and M. Zito. On the approximability of the maximum induced matching problem. *Journal of Discrete Algorithms*, 3(1):79–91, 2005.

J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

A. Ehrenfeucht and G. Rozenberg. Primitivity is hereditary for 2-structures. *Theoretical Computer Science*, 70(3):343–358, 1990.

T. Ekim and J. Gimbel. Partitioning graphs into complete and empty graphs. *Discrete Mathematics*, 309(19):5849–5856, 2009.

L. Esperet. Dynamic list coloring of bipartite graphs. *Discrete Applied Mathematics*, 158 (17):1963–1965, 2010.

R.J. Faudree, A. Gyárfas, R.H. Schelp, and Z. Tuza. Induced matchings in bipartite graphs. *Discrete Mathematics*, 78(1–2):83–87, 1989.

O. Favaron. Independence and upper irredundance in claw-free graphs. *Discrete Applied Mathematics*, 132(1–3):85–95, 2003.

T. Feder, P. Hell, S. Klein, and R. Motwani. List partitions. *SIAM Journal on Discrete Mathematics*, 16(3):449–478, 2003.

M.R. Fellows, D. Hermelin, F. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1): 53–61, 2009.

J. Flum and M. Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 1 edition, 2006.

R. Forcade. Smallest maximal matchings in the graph of the d-dimensional cube. *Journal of Combinatorial Theory, Series B*, 14(2):153–156, 1973.

J-L. Fouquet, V. Giakoumakis, J-M. Vanherpe, and S. Olariu. Bipartite graphs totally decomposable by canonical decomposition. *International Journal of Foundations of Computer Science*, 10(4):513, 1999.

T. Gallai. Transitiv orientierbare graphen. *Acta Mathematica Hungarica*, 18:25–66, 1967.

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

M.U. Gerber and V.V. Lozin. On the stable set problem in special P_5-free graphs. *Discrete Applied Mathematics*, 125(2–3):215–224, 2003.

M.U. Gerber, A. Hertz, and D. Schindl. P_5-free augmenting graphs and the maximum stable set problem. *Discrete Applied Mathematics*, 132(1–3):109–119, 2003.

V. Giakoumakis and J.M. Vanherpe. Linear time recognition and optimizations for weak-bisplit graphs, bi-cographs and bipartite P_6-free graphs. *International Journal of Foundations of Computer Science*, 14(1):107, 2003.

M.C. Golumbic. Trivially perfect graphs. *Discrete Mathematics*, 24(1):105–107, 1978.

M.C. Golumbic and R.C. Laskar. Irredundancy in circular arc graphs. *Discrete Applied Mathematics*, 44(1–3):79–89, 1993.

M.C. Golumbic and M. Lewenstein. New results on induced matchings. *Discrete Applied Mathematics*, 101(1–3):157–165, 2000.

Z. Gotthilf and M. Lewenstein. Tighter approximations for maximum induced matchings in regular graphs. *Lecture Notes in Computer Science*, 3879:270–281, 2006.

D.L. Grinstead, P.J. Slater, N.A. Sherwani, and N.D. Holmes. Efficient edge domination problems in graphs. *Information Processing Letters*, 48(5):221–228, 1993.

M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, second corrected edition, 1993.

M. Grötschel, L. Lovász, and A. Schrijver. Polynomial algorithms for perfect graphs. In *Topics on Perfect Graphs*, volume 88 of *North-Holland Mathematics Studies*, pages 325–356. North-Holland, 1984.

M. Habib and M.C. Maurer. On the X-join decomposition for undirected graphs. *Discrete Applied Mathematics*, 1(3):201–207, 1979.

F. Harary, J.P. Hayes, and H-J. Wu. A survey of the theory of hypercube graphs. *Computers & Mathematics with Applications*, 15(4):277–289, 1988.

P. Hell, S. Klein, L.T. Nogueira, and F. Protti. Partitioning chordal graphs into independent sets and cliques. *Discrete Applied Mathematics*, 141(1–3):185–194, 2004.

A. Hertz and V.V. Lozin. The maximum independent set problem and augmenting graphs. In *Graph Theory and Combinatorial Optimization*, pages 69–99. Springer, 2005.

I. Holyer. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10(4): 718–720, 1981.

C.T. Hoàng and V.B. Le. On P_4-transversals of perfect graphs. *Discrete Mathematics*, 216(1–3):195–210, 2000.

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

Y. Huang and Y. Chu. A note on the computational complexity of graph vertex partition. *Discrete Applied Mathematics*, 155(3):405–409, 2007.

J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

Y. Jing-Ho, C. Jer-Jeong, and G.J. Chang. Quasi-threshold graphs. *Discrete Applied Mathematics*, 69(3):247–255, 1996.

P.C. Kainen. A lower bound for crossing numbers of graphs with applications to $K_n, K_{p,q}$, and $Q(d)$. *Journal of Combinatorial Theory, Series B*, 12(3):287–298, 1972.

M. Kamiński and V. Lozin. Coloring edges and vertices of graphs without short or long cycles. *Contributions to Discrete Mathematics*, 2(1):61–66, 2007a.

M. Kamiński and V. Lozin. Vertex 3-colorability of claw-free graphs. *Algorithmic Operations Research*, 2(1), 2007b.

M. Kamiński, V.V. Lozin, and M. Milanič. Recent developments on graphs of bounded clique-width. *Discrete Applied Mathematics*, 157(12):2747–2761, 2009.

R.J. Kang, M. Mnich, and T. Müller. Induced matchings in subcubic planar graphs. *Lecture Notes in Computer Science*, 6347:112–122, 2010.

C.W. Ko and F.B. Shepherd. Bipartite domination and simultaneous matroid covers. *SIAM Journal on Discrete Mathematics*, 16(4):517–523, 2003.

D. Kobler and U. Rotics. Finding maximum induced matchings in subclasses of claw-free and P_5-free graphs, and in graphs with matching and induced matching of equal maximum size. *Algorithmica*, 37:327–346, 2003.

M. Kochol, V. Lozin, and B. Randerath. The 3-colorability problem on graphs with maximum degree four. *SIAM Journal on Computing*, 32(5):1128–1139, 2003.

N. Korpelainen. A polynomial-time algorithm for the dominating induced matching problem in the class of convex graphs. *Electronic Notes in Discrete Mathematics*, 32: 133–140, 2009.

N. Korpelainen. *Boundary properties of graphs*. PhD thesis, University of Warwick, 2012.

J. Kratochvil and I. Schiermeyer. On the computational complexity of (O, P)-partition problems. *Discussiones Mathematicae Graph Theory*, 17(2):253–258, 1997.

J. Kratochvíl. Regular codes in regular graphs are difficult. *Discrete Mathematics*, 133 (1–3):191–205, 1994.

D. Král', J. Kratochvíl, Z. Tuza, and G. Woeginger. Complexity of coloring graphs without forbidden induced subgraphs. *Lecture Notes in Computer Science*, 2204:254–262, 2001.

J. Kára and J. Kratochvíl. Fixed parameter tractability of independent set in segment intersection graphs. *Lecture Notes in Computer Science*, 4169:166–174, 2006.

V.B. Le, B. Randerath, and I. Schiermeyer. On the complexity of 4-coloring graphs without long induced paths. *Theoretical Computer Science*, 389(1–2):330–335, 2007.

V.B. Le, R. Mosca, and H. Müller. On stable cutsets in claw-free graphs and planar graphs. *Journal of Discrete Algorithms*, 6(2):256–276, 2008.

M. Livingston and Q.F. Stout. Distributing resources in hypercube computers. In *Proceedings of the third conference on Hypercube concurrent computers and applications: Architecture, software, computer systems, and general issues - Volume 1*, C3P, pages 222–231, New York, NY, USA, 1988. ACM.

L. Lovász and M.D. Plummer. *Matching theory*, volume 121 of *North-Holland Mathematics Studies*. North-Holland Publishing Co., Amsterdam, 1986. Annals of Discrete Mathematics, 29.

V. Lozin and R. Mosca. Maximum independent sets in subclasses of P_5-free graphs. *Information Processing Letters*, 109(6):319–324, 2009.

V. Lozin and D. Rautenbach. On the band-, tree-, and clique-width of graphs with bounded vertex degree. *SIAM Journal on Discrete Mathematics*, 18(1):195–206, 2004.

V.V. Lozin. Bipartite graphs without a skew star. *Discrete Mathematics*, 257(1):83–100, 2002a.

V.V. Lozin. On maximum induced matchings in bipartite graphs. *Information Processing Letters*, 81(1):7–11, 2002b.

V.V. Lozin. Between 2- and 3-colorability. *Information Processing Letters*, 94(4):179–182, 2005.

V.V. Lozin. Stability preserving transformations of graphs. *Annals of Operations Research*, 188:331–341, 2011.

V.V. Lozin and M. Milanič. A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. *Journal of Discrete Algorithms*, 6(4):595–604, 2008.

V.V. Lozin, J. Volz, and S. Moran. The clique-width of bipartite graphs in monogenic classes. *International Journal of Foundations of Computer Science*, 19(2):477–494, 2008.

C.L. Lu and C.Y. Tang. Solving the weighted efficient edge domination problem on bipartite permutation graphs. *Discrete Applied Mathematics*, 87(1–3):203–211, 1998.

C.L. Lu, M-T. Ko, and C.Y. Tang. Perfect edge domination and efficient edge domination in graphs. *Discrete Applied Mathematics*, 119(3):227–250, 2002.

B. Lévêque and F. Maffray. Coloring bull-free perfectly contractile graphs. *SIAM Journal on Discrete Mathematics*, 21(4):999–1018, 2008.

F. Maffray and M. Preissmann. On the NP-completeness of the k-colorability problem for triangle-free graphs. *Discrete Mathematics*, 162(1–3):313–317, 1996.

N.V.R. Mahadev and U.N. Peled. Threshold graphs and related topics. In *Threshold Graphs and Related Topics*, volume 56 of *Annals of Discrete Mathematics*, page 543. Elsevier, 1995.

R.M. McConnell and J.P. Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201(1–3):189–241, 1999.

O. McNoleg. The integration of gis, remote sensing, expert systems and adaptive co-kriging for environmental habitat modeling of the highland haggis using object-oriented, fuzzy-logic and neural-network techniques. *Computers & Geosciences*, 22(5): 585–588, 1996.

G.J. Minty. On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory, Series B*, 28(3):284–304, 1980.

R. Mosca. Stable sets in certain P_6-free graphs. *Discrete Applied Mathematics*, 92(2–3): 177–191, 1999.

H. Moser and S. Sikdar. The parameterized complexity of the induced matching problem. *Discrete Applied Mathematics*, 157(4):715–727, 2009.

H. Moser and D.M. Thilikos. Parameterized complexity of finding regular induced subgraphs. *Journal of Discrete Algorithms*, 7(2):181–190, 2009.

O.J. Murphy. Computing independent sets in graphs with large girth. *Discrete Applied Mathematics*, 35(2):167–170, 1992.

R.H. Möhring. Algorithmic aspects of comparability graphs and interval graphs. In *Graphs and order: the role of graphs in the theory of ordered sets and its applications*, number v. 1984 in NATO ASI series: Mathematical and physical sciences, pages 41–101. D. Reidel Pub. Co., 1985.

S. Olariu. Paw-free graphs. *Information Processing Letters*, 28(1):53–54, 1988.

S. Olariu. On the homogeneous representation of interval graphs. *Journal of Graph Theory*, 15(1):65–80, 1991.

Y. Orlovich, G. Finke, V. Gordon, and I. Zverovich. Approximability results for the maximum and minimum maximal induced matching problems. *Discrete Optimization*, 5(3):584–593, 2008.

C.H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

S. Poljak. A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 15:307–309, 1974.

F. Protti, M.D. da Silva, and J.L. Szwarcfiter. Applying modular decomposition to parameterized cluster editing problems. *Theory of Computing Systems*, 44:91–104, 2009.

M. Radovanović and K. Vušković. A class of three-colorable triangle-free graphs (to appear). *Journal of Graph Theory*, 2012.

V. Raman and S. Saurabh. Triangles, 4-cycles and parameterized (in-)tractability. *Lecture Notes in Computer Science*, 4059:304–315, 2006.

B. Randerath. 3-colorability and forbidden subgraphs. I: Characterizing pairs. *Discrete Mathematics*, 276(1–3):313–325, 2004.

B. Randerath and I. Schiermeyer. A note on Brooks' theorem for triangle-free graphs. *The Australasian Journal of Combinatorics*, 26:3–9, 2002.

B. Randerath and I. Schiermeyer. 3-colorability $\in P$ for P_6-free graphs. *Discrete Applied Mathematics*, 136(2–3):299–313, 2004.

B. Randerath and I. Schiermeyer. On maximum independent sets in P_5-free graphs. *Discrete Applied Mathematics*, 158(9):1041–1044, 2010.

B. Randerath, I. Schiermeyer, and M. Tewes. Three-colourability and forbidden subgraphs. II: polynomial algorithms. *Discrete Mathematics*, 251(1–3):137–153, 2002.

H. Randerath. *The vizing bound for the chromatic number based on forbidden pairs*. Shaker, 1998.

M. Rao. MSOL partitioning problems on graphs of bounded treewidth and clique-width. *Theoretical Computer Science*, 377(1–3):260–267, 2007.

M. Rao. Solving some NP-complete problems using split decomposition. *Discrete Applied Mathematics*, 156(14):2768–2780, 2008.

I. Rusu. Maximum weight edge-constrained matchings. *Discrete Applied Mathematics*, 156(5):662–672, 2008.

D. Schindl. Some new hereditary classes where graph coloring remains NP-hard. *Discrete Mathematics*, 295(1–3):197–202, 2005.

J.P. Spinrad. Nonredundant 1's in Γ-free matrices. *SIAM Journal on Discrete Mathematics*, 8(2):251–257, 1995.

J. Stacho. On P_4-transversals of chordal graphs. *Discrete Mathematics*, 308(23):5548–5554, 2008.

L.J. Stockmeyer and V.V. Vazirani. NP-completeness of some generalizations of the maximum matching problem. *Information Processing Letters*, 15(1):14–19, 1982.

M. Tedder, D. Corneil, M. Habib, and C. Paul. Simpler linear-time modular decomposition via recursive factorizing permutations. *Lecture Notes in Computer Science*, 5125: 634–645, 2008.

S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505–517, 1977.

D. Upper. The unsuccessful self-treatment of a case of "writer's block". *Journal of Applied Behavioural Analysis*, 4(3):497, 1974.

G.J. Woeginger and J. Sgall. The complexity of coloring graphs without long induced paths. *Acta Cybernetica*, 15(1):107–117, 2001.

M. Yannakakis. Node-deletion problems on bipartite graphs. *SIAM Journal on Computing*, 10(2):310–327, 1981.

M. Zito. Induced matchings in regular graphs and trees. *Lecture Notes in Computer Science*, 1665:89–101, 1999.