Algorithmic and combinatorial problems on graph classes

Reza SaeiDinvar

Dissertation for the degree of Philosophiae Doctor (PhD)

Department of Informatics
University of Bergen

May 2015

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Pinar Heggernes, for all the help and support I have received from her during the past four years. This thesis would not have been completed without her guidance. I would like to thank her for her patience in reviewing many inferior drafts and guiding me with almost every single detail. I also want to thank her for giving me the opportunities to travel to many conferences and to collaborate with other great researchers. I will forever be beholden to you Pinar. Another person to whom I owe a lot is Pim van 't Hof. I cannot emphasize enough the importance of his advices in my advancements. Thank you Pim for being such a nice friend and colleague.

I would like to thank my evaluation committee, Prof. Hajo Broersma, Prof. Vadim Lozin, and Prof. Øyvind Ytrehus for accepting to review and evaluate this thesis.

My endless thanks to all my coauthors: Manu Basavaraju, Rémy Belmonte, Henning Fernau, Peter A. Golovach, Pinar Heggernes, Pim van 't Hof, Dieter Kratsch, Erik Jan van Leeuwen, Daniel Meister, Arash Rafiey, and Yngve Villanger. I would like to thank especially Henning Fernau, Daniel Meister, Dieter Kratsch and Erik Jan van Leeuwen for hosting me during my research visits to Trier, Metz and Saarbrucken.

My deepest thanks go to all my colleagues at the Algorithms group. The unique scientific and social atmosphere of the Algorithms group which has made my study a great and unforgettable experience is due to all of you. I want to thank in particular, my former and present officemates Sadia Sharmin, Md Naim and Markus Dregi for making our office a nice place to work, Sigve Sæther for teaching me Metapost tricks, as well as my other fellow Ph.D. students Pål Drange, Mithilesh Kumar, and Akanksha Agrawal.

I would like to extend my sincerest thanks to the administrative staff at the department of Informatics for easing all the administrative works and ensuring a nice educational environment. I appreciate University of Bergen for financing my study and for supporting many interesting and useful travels. I would also like to thank the Meltzer Research Fund for the grant I received from in 2013.

Words cannot describe how thankful I am to my family and especially my parents for always supporting me and my choices. I thank all my friends most warmly, in particular Naser and Maryam, a couple from whom I have received a lot of kindness, help and support within the past four years.

Bergen, May 2015
Reza SaeiDinvar

Contents

Part I
Background

Chapter 1

Introduction

Graphs are mathematical structures to describe pairwise relationships between objects, such as friendship between people, or communication networks. They provide a natural model to describe a wide range of problems emerging from industry and scientific fields like biology, chemistry, physics, informatics, sociology, and linguistics. Examples of such problems are in particular VLSI layouts, molecular biology and DNA mappings, phylogenetic trees, sparse systems of linear equations, relational databases, and resource allocation. Unfortunately, the solutions of most of the interesting graph problems are still far beyond today's combinatorial knowledge and computational tools. Determining the Ramsey numbers of graphs is such a problem. Out of infinitely many numbers to be determined, the number of known nontrivial Ramsey numbers can be counted on the fingers of two hands after 80 years. In addition, most of the important graph problems are NP-hard, meaning that algorithms to solve all their instances in reasonable time are highly unlikely.

Discovering more structural properties of the instances of a hard problem grants stronger combinatorial statements and more efficient algorithms so that one can hopefully manage to solve the problem on these instances. Structural graph theory is the branch of graph theory that classifies and studies graphs based on the properties that they share. A *graph class* is a collection of graphs that share a property, e.g., the class of acyclic graphs. Study of graph classes has long been an interesting line of research in graph theory and it has flourished since the introduction of perfect graphs by Berge in early 1960s [7]. This development has opened new paths to cope with graph problems which are hard on general graphs. Trying to exploit structural properties of graph classes to cope with intractability of hard problems resulted in designing algorithms particularly tailored for graph classes. However some hard problems remain intractable even for very restricted classes of graphs.

In the present thesis, our aim is to exploit properties of graphs with particular structure to improve the existent results on problems that are hard on general graphs. In addition to algorithmic results, we also deal with combinatorial problems. For these problems, we have two objectives from restricting ourselves to structured graphs, improving the existing combinatorial results and managing to solve problems whose answers are not known for general graphs.

This thesis consists of two parts. We are currently in Part I, and the following six papers form its second part:

Paper I: [48]
Computing the metric dimension for chain graphs.
Henning Fernau, Pinar Heggernes, Pim van 't Hof, Daniel Meister, and Reza Saei.

Paper II: [47]
Covering the edges of bipartite graphs by overlapping C_4's.
Henning Fernau, Pinar Heggernes, Daniel Meister, and Reza Saei.

Paper III: [71]
Finding Disjoint Paths in Split Graphs.
Pinar Heggernes, Pim van 't Hof, Erik Jan van Leeuwen, and Reza Saei.

Paper IV: [62]
Subset Feedback Vertex Sets in Chordal Graphs.
Petr A. Golovach, Pinar Heggernes, Dieter Kratsch, and Reza Saei.

Paper V: [4]
Maximal induced matchings in triangle-free graphs.
Manu Basavaraju, Pinar Heggernes, Pim van 't Hof, Reza Saei, and Yngve Villanger.

Paper VI: [5]
Graph Classes and Ramsey numbers.
Remy Belmonte, Pinar Heggernes, Pim van't Hof, Arash Rafiey, and Reza Saei.

The first two papers of Part II concern two hard optimization problems that we have managed to solve in polynomial time on some classes of graphs. METRIC DIMENSION is a problem that is NP-complete even on very restricted classes of graphs like bipartite graphs. We solve this problem in linear time on chain graphs which form a subclass of bipartite graphs. As a corollary, in the case of twin-free chain graphs this yields a formula to calculate the metric dimension. The problem studied in Paper II is the problem of covering the edge set of graphs by cycles of length 4 also called C_4. This problem is also known to be NP-complete on bipartite graphs. We determine subclasses of bipartite graphs on which the problem becomes tractable. We give an exact formula to calculate the size of the smallest C_4-covers of complete bipartite graphs. In addition, we present an algorithm that computes C_4-cover number of trees in linear-time.

Paper III is on a well-known decision problem, DISJOINT PATHS, which is NP-complete on general graphs. We show that both vertex disjoint and edge disjoint versions of the problem remain NP-hard on split graphs. Furthermore, although both versions of the problem are known to be fixed-parameter tractable, it was shown that vertex disjoint version does not admit any kernel of polynomial size on general graphs. We show that both versions of the problem, vertex disjoint and edge disjoint, admit polynomial kernels on split graphs.

Enumerating and determining the number of various objects in graphs have found interest and applications in computer science, especially in the area of exact algorithms. The most famous example of this kind is the theorem of Moon and Moser [87] which states that the maximum number of maximal cliques and hence maximal independent sets in any graph on n vertices is $3^{n/3}$. In Paper IV we study minimal subset feedback vertex sets. Both weighted and unweighted versions of SUBSET FEEDBACK VERTEX SET are NP-complete on general graphs, and remain NP-complete on split graphs [50]. We enumerate all minimal subset feedback vertex sets of chordal graphs. As a consequence, we obtain an exponential-time algorithm for SUBSET FEEDBACK VERTEX

SET on chordal graphs which runs faster than the best known algorithm for general graphs. We also obtain an upper bound on the number of minimal subset feedback vertex sets of chordal graphs.

In Paper V, we determine the maximum number of maximal induced matchings in triangle-free graphs. In general graphs, the maximum number was known [69] and the graph that has the maximum number of maximal induced matchings contains many triangles. Motivated by this, we study triangle-free graphs; and we obtain a smaller number for these graphs than for general graphs. There is a polynomial-delay algorithm that lists all maximal induced matchings of graphs [20, 73, 103]. Combining our result and this polynomial-delay algorithm yields an algorithm that lists all maximal induced matchings and returns a maximum induced matching of triangle-free graphs. Hence we get an algorithm which runs faster than the best known algorithm for this purpose for general graphs.

In Paper VI, we study Ramsey numbers of graph classes. Extremal graph theory is a branch of graph theory that includes interesting problems of pure combinatorial flavor. Ramsey theory on graphs is one of the most famous extremal graph theory problems which studies the size of a system in order to ensure that it contains some particular structure. The fundamental theorem of this theory on graphs states that for every pair of positive integers i and j, there exists a finite integer $R(i,j)$ such that every graph on at least $R(i,j)$ vertices contains either a clique of size i or an independent set of size j. Finding these numbers for general graphs is very difficult and we do not know the exact value of $R(4,6)$ or $R(3,10)$ after eight decades. In paper VI, we define the Ramsey number of graph classes, and we identify classes of graphs for which we are able to give a formula for all Ramsey numbers in terms of i and j.

This first part of the thesis comprises preliminaries, definitions and concepts in a broader perspective to provide the required background for the second part. Chapter 2 provides all the definitions and notation used throughout the first part. In Chapter 3, we briefly survey the methods to cope with NP-hard problems. In Chapter 4, we study enumeration problems and problems that aim to upper bound the number of various objects in graphs. Chapter 5 provides the required background on extremal graphs and Ramsey theory. Chapter 6 is dedicated to conclusion and future works.

Chapter 2

Notation and terminology

In the current chapter we give most of the definitions and notation required throughout the first part of the thesis. Additional notation and definitions are presented when they are needed. However, we assume that the reader is familiar with the fundamental definitions of set theory. We start with the basic definitions of graph theory and then we define the required classes of graphs. In the last section of this chapter we survey some well known graph problems and their computational complexity.

2.1 Graph theory

A *graph* G is defined as a pair of sets $V(G)$ and $E(G)$ where each element of $E(G)$ is a subset of $V(G)$ of size 2. Sets $V(G)$ and $E(G)$ are respectively called the *vertex set* and the *edge set* of G. The best way to illustrate G is embedding it on the plane by putting a dot for each element of $V(G)$ and drawing a line between two dots when the corresponding vertices form an element of $E(G)$. A graph G is *finite* if $V(G)$ has finitely many elements. All the graphs in this thesis are finite.

An edge $\{u,v\}$ is usually written as uv. Let G be a graph with vertex set $V(G)$ and edge set $E(G)$. Two vertices u and v in $V(G)$ are called *adjacent* if $uv \in E(G)$. For every vertex $v \in V(G)$, the set $\{u \,|\, uv \in E(G)\}$ is called the *open neighborhood* of v and is denoted by $N_G(v)$. The *degree* of a vertex $v \in V(G)$, denoted by $d_G(v)$, is the size of its open neighborhood. The *closed neighborhood* of v is defined as $N_G[v] = N_G(v) \cup \{v\}$. If A is a subset of $V(G)$, then $N_G(A) = \bigcup_{v \in A} N_G(v)$. We say that an edge $e \in E(G)$ is *incident* with a vertex $v \in V(G)$ if $uv \in E(G)$ for some $u \in V(G)$; vertices u and v are called the *endpoints* of e. Two edges are also called *adjacent* if they share one of their endpoints. Two vertices u and v are *false twins*, or simply *twins*, if $N_G(u) = N_G(v)$. The binary relation of being twins is an equivalence relation on $V(G)$. The equivalence classes of this relation are called the *twin classes* of G. We call the corresponding partition of $V(G)$ the *twin partition* of G.

A graph is called *r-regular* if all of its vertices have degree r. A *complete graph* on n vertices, denoted by K_n, is a graph in which all of the vertices are pairwise adjacent. Let G_1 and G_2 be two graphs such that $V(G_1) \cap V(G_2) = \emptyset$. The *disjoint union* of G_1 and G_2 is the graph G with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. The *join* of G_1 and G_2 is the graph G with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2) \cup \{uv \,|\, u \in V(G_1), v \in V(G_2)\}$. A graph with vertex set $\{v_1, v_2, \ldots, v_n\}$ and edge set $\{v_i v_{i+1} \,|\, 1 \leq i \leq n-1\}$ is called a *path* of length $n-1$ or a path on n vertices and is denoted by P_n. The disjoint union of m copies of P_2 is called an *m-stripe*. A *cycle* C_n of length n (on n vertices) is obtained from a path of length $n-1$ by adding v_nv_1 to the edge set. A graph on n vertices $x_1, x_2, \ldots, x_n$ in which vertices $x_1, x_2, \ldots, x_{n-1}$ form a cycle and all of them are adjacent to x_n is called a *wheel* graph and is denoted by W_n. The graph $\overline{G}$ is the *complement graph* of a G if its vertex set is $V(G)$ and two vertices in $\overline{G}$ are adjacent if and only if they are not adjacent in G.

An *isomorphism* between two graphs G and H is a bijective function ϕ from $V(G)$ to $V(H)$ such that any pair of vertices $u, v \in V(G)$ are adjacent in G if and only if $\phi(u)$ and $\phi(v)$ are adjacent in H. If there exists an isomorphism between two graphs then they are called *isomorphic*. A Graph H is a *subgraph* of a G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Then graph G is called a *supergraph* of H. An *induced subgraph* H of a graph G is a subgraph of G with the property that every two vertices u and v of $V(H)$ are adjacent in H if and only if they are adjacent in G. Let X be a subset of vertices of a graph G. The subgraph induced by X which is denoted by $G[X]$, is an induced subgraph of G whose vertex set is X. We denote the induced subgraph $G[V(G) \setminus X]$ by $G - X$. When X has only one element, say $X = \{v\}$, the induced subgraph $G[V(G) \setminus \{v\}]$ is simply denoted by $G - v$. A subgraph H of a graph G with a specified property is called *maximal* if there is no subgraph H' of G with the same property such that H is a proper subgraph of H'.

An *independent set* in a graph G is a subset I of its vertices such that $G[I]$ has no edges. A graph on n vertices $x_1, x_2, \ldots, x_n$ in which vertices $x_1, x_2, \ldots, x_{n-1}$ form an independent set and all of them are adjacent to x_n is called a *star* graph and is denoted by S_n. A *clique* in a graph G is a subset C of its vertices such that all vertices of $G[C]$ are pairwise adjacent. The *clique number* of a graph G, denoted by $\omega(G)$, is the size of a largest clique in G. A *path* in a graph G is a subgraph of G which is isomorphic to P_k for some $1 \leq k \leq |V(G)|$. A *chord* of a path P_k in a graph G is an edge of G whose endpoints belong to $V(P_k)$ and it does not belong to $E(P_k)$. The *length of a chord* of a path P_k is defined to be the number of edges of P_k between the endpoints of the chord. Cycles in a graph, chords of a cycle, and the length of a chord of a cycle are defined similarly. A chord-less cycle of length at least 4 is called *hole*. An *anti-hole* is the complement of a hole.

A graph G is *connected* if every pair of its vertices are linked by a path in G. A maximal connected subgraph of G is called a *connected component* of G. The *distance* between two vertices u and v in a graph G is denoted by $d_G(u, v)$ and it is defined to be the length of a shortest path in G linking the two vertices.

A *matching* M in G is a subset of its edges such that no two edges in M are adjacent. For a vertex $v \in V(G)$, we say that a matching M *covers* or *saturates* v if v is an endpoint of an edge in M. The size of a largest matching in G is denoted by $\nu(G)$. A matching is *induced* if the endpoints of its edges induce a 1-regular subgraph.

An assignment of labels or colors to the vertex set of a graph G in a way that there are no two adjacent vertices with the same label or color is called a *vertex coloring* of G. A graph is *k-colorable* if it has a vertex coloring with k labels or colors. The fewest number of colors required to color the vertex set of a graph G is called the *chromatic number* of G and is denoted by $\chi(G)$. An *edge coloring* of a graph G is assigning labels or colors to the edge set of G such that no two adjacent edges get the same color.

2.2 Graph classes

This section is dedicated to definitions of the classes of graphs that will be mentioned later on in Part I. Generally, a *graph class* $\mathcal{G}$ is a collection of graphs with some common properties. Let $\mathcal{G}$ and $\mathcal{G}'$ be two graphs classes. Class $\mathcal{G}$ is a *subclass* of class $\mathcal{G}'$ if every graph in $\mathcal{G}$ is a graph in $\mathcal{G}'$. Class $\mathcal{G}'$ is then called a *superclass* of $\mathcal{G}$. This containment relation defines a hierarchy of graph classes. This hierarchy is depicted in figure 3.2 for the classes of graphs defined in this section.

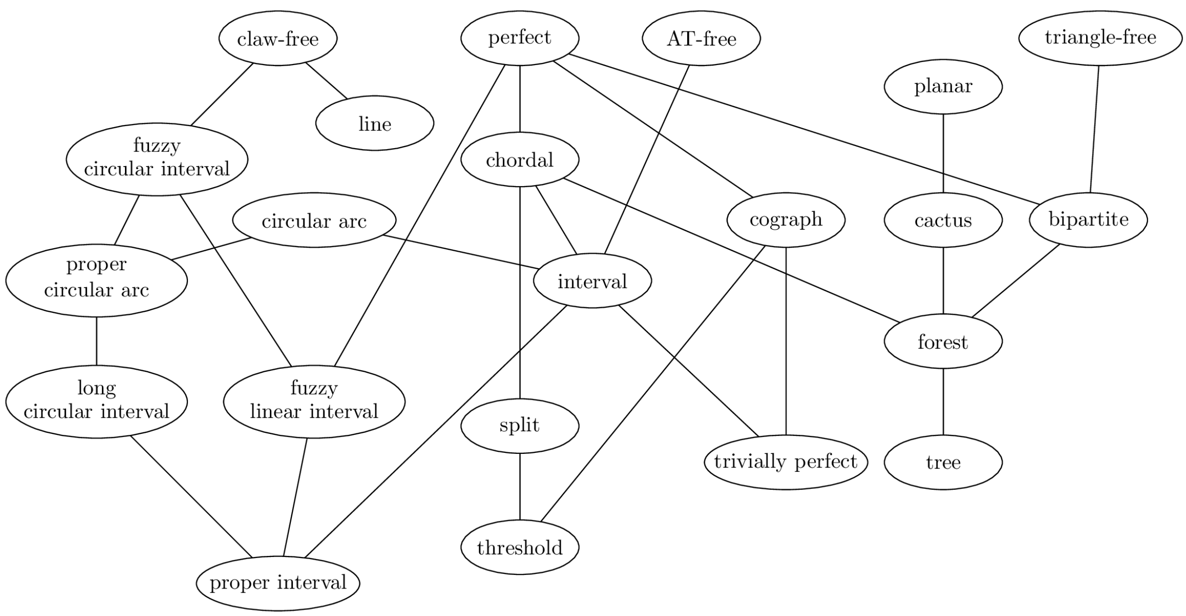


Figure 2.1: An overview of the graph classes mentioned in the first part of the thesis. A path between two classes $\mathcal{G}$ and $\mathcal{H}$ where $\mathcal{G}$ is above $\mathcal{H}$ indicates that $\mathcal{G}$ is a proper superclass of $\mathcal{H}$.

Perfect graphs: For any graph G, it is obvious that $\chi(G) \geq \omega(G)$ as vertices of a clique should receive different colors. However the equality does not hold in general. For instance for a cycle C_k, where $k \geq 5$ and k is odd, $\chi(C_k) = 3$ and $\omega(C_k) = 2$.

A *perfect graph* is a graph in which the chromatic number of every induced subgraph equals the clique number of that subgraph. A graph is *Berge* if it contains neither an odd hole nor an odd anti-hole. In the early 1960s, Claude Berge formulated the following two conjectures:

(A) the complement of a perfect graph is perfect, and

(B) a graph is perfect if and only if it contains neither an odd hole nor an odd anti-hole.

Conjecture A was proved by Lovász in 1972 [81] and it is known as *the perfect graph theorem*. The Perfect graph theorem contends that the class of perfect graphs is *self-complementary*. Conjecture B was one of the most challenging conjectures in graph theory for more than four decades and finally it was proved in 2002 by Chudnovsky, Robertson, Seymour, and Thomas [22]. Now it is known as *the strong perfect* *graph theorem* which asserts that a graph is perfect if and only if it is a Berge graph.

Chordal graphs: A *chordal graph* is a graph in which every cycle of length at least 4 has a chord. A vertex v of a graph G is called *simplicial* if $N(v)$ is a clique. A *perfect elimination order* of a graph G is an ordering $v_1, \ldots, v_n$ of its vertices, such that v_i is simplicial in $G[\{v_i, \ldots, v_n\}]$ for all $1 \leq i \leq n$. A graph is chordal if and only if it has a perfect elimination order [57]. Chordal graphs are perfect.

Interval graphs: A graph $G = (V, E)$ is an *interval graph* if there exists a family $\mathscr{I}$ of closed intervals of the real line such that each vertex $v \in V$ is associated with an interval in $\mathscr{I}$, and two vertices of G are adjacent if and only if their associated intervals intersect. The pair $(G, \mathscr{I})$ is called an *interval model* of G. Interval graphs are chordal.

Proper interval graphs: An interval graph G with an interval model $(G, \mathscr{I})$ is called a *proper interval graph* if no interval of $\mathscr{I}$ properly contains another.

Split graphs: A *split graph* is a graph whose vertex set can be partitioned into a clique C and an independent set I, either of which might be empty. Such a partition (C, I) is called a *split partition*. Note that, in general, a split graph can have more than one split partition. Observe that every split graph is chordal. If a split graph is disconnected, then only one of its components can be on more than one vertices and the other components are isolated vertices.

Threshold graphs: A vertex in a graph G is *dominating* if it is adjacent to all other vertices of G. A graph is a *threshold graph* if it can be constructed from the empty graph by repeatedly adding either an isolated vertex or a dominating vertex. Threshold graphs are split.

$\mathcal{F}$-free graphs: Let $\mathcal{F}$ be a family of graphs. The class of *$\mathcal{F}$-free graphs* consists of all graphs that contain no induced subgraph isomorphic to a member of $\mathcal{F}$. Several important classes of graphs are characterized by a single forbidden induced subgraph. If $\mathcal{F}$ contains only a single graph H, then we write H-free. The complete bipartite graph $K_{1,3}$ is also called the *claw*. Therefore the class of *claw-free* graphs is the class of $\mathcal{F}$-free graphs where $\mathcal{F} = \{K_{1,3}\}$. When the single forbidden induced subgraph is K_3 then the class is called *triangle-free*. Three vertices form an *asteroidal triple* (*AT*) for a graph if any pair of these vertices are linked by a path that does not contain any neighborhood of the third vertex. An *AT-free* graph is a graph that contains no AT. For every positive integer k, the class of *P_k-free graphs* is the class of all graphs that do not contain P_k as an induced subgraph.

Cographs: A graph is a *cograph* if it can be constructed from single vertices by disjoint union and join operations. Cographs are equivalent to the class of P_4-free graphs.

Trivially perfect graphs: A *trivially perfect graph* is a graph in which the size of a largest independent set of every induced subgraph equals the number of maximal cliques of that subgraph. Every trivially perfect graph is a cograph as it does not contain P_4 or C_4 as an induced subgraph. A trivially perfect graph is also an interval graph.

Bipartite graphs: Bipartite graphs form an important subclass of perfect graphs. A graph $G = (A, B, E)$ is *bipartite* if its vertex set can be partitioned into two non-empty subsets A and B such that there is no edge of G with both endpoints in one of the subsets. It is easy to see that a graph is bipartite if and only if it is 2-colorable. A graph is bipartite if and only if it contains no cycle of odd length [75]. This implies that all acyclic graphs are bipartite. A bipartite graph $G = (A, B, E)$ is called *complete bipartite* if $|N(u)| = |B|$ for every $u \in A$ and $|N(v)| = |A|$ for every $v \in B$. We denote this graph by $K_{|A|,|B|}$.

Forest: A *forest* is a graph that has no cycle. The vertices of degree 1 in a forest are called *leaves*. Forests form a subclass of bipartite graphs and also form a subclass of chordal graphs.

Tree: A *tree* is a connected forest.

Chain graphs: A bipartite graph $G = (A, B, E)$ is called a *chain graph* if the vertices in A can be ordered as $\langle a_1, \dots, a_p \rangle$ such that $N_G(a_1) \subseteq \dots \subseteq N_G(a_p)$. It is easy to see that this implies an ordering $\langle b_1, \dots, b_q \rangle$ on the vertices in B such that $N_G(b_1) \supseteq \dots \supseteq N_G(b_q)$ [14]. If a chain graph is disconnected, then it has at most one connected component which is not an isolated vertex. Let $G = (A, B, E)$ be a connected chain graph on at least two vertices, and consider its twin partition. Since each twin class is either a subset of A or a subset of B, this is a partition of A and B into twin classes $A_1, A_2, \dots, A_k$ and $B_1, B_2, \dots, B_{k'}$, respectively, such that $N_G(A_1) \subseteq \dots \subseteq N_G(A_k)$ and $N_G(B_1) \supseteq \dots \supseteq N_G(B_{k'})$. From the definition of chain graphs, it is easy to see that $k = k'$. A chain graph in which no two vertices are twins is called a *twin-free* chain graph.

Planar graphs: Let G be a graph. An embedding of G on the plane is called *crossing-free* if no two curves representing edges cross each other. Note that two curves having a common endpoint are not considered crossing each other. A graph is *planar* if it has a crossing-free embedding. For instance K_4 is a planar graph but K_5 and $K_{3,3}$ are not. A *direct subdivision* of a graph $G = (V, E)$ is the graph $G' = (V', E')$ where $V' = V \cup \{w\}$ and $E' = (E \setminus \{uv\}) \cup \{uw, wv\}$ for some edge $uv \in E$ and a newly added vertex $w \notin V$.

A graph which is obtained from G by a sequence of direct subdivisions is called a *subdivision* of G. A graph is planar if and only if it does not contain a subdivision of K_5 or $K_{3,3}$ [76].

Line graphs: Let G be a graph with vertex set $V(G)$ and edge set $E(G)$. The *line graph* of G which is denoted by $L(G)$ is a graph such that its vertex set is $E(G)$ and two vertices of $L(G)$ are adjacent if and only if the corresponding edges in G are adjacent. Graph G is called the *preimage* graph of $L(G)$. A graph is a *line graph* if it is the line graph of some graph. Except the cycle on three vertices, the preimage graph of every line graph is unique [106]. Line graphs are claw-free.

Cactus graphs: A graph G is a *cactus graph* if every edge of G is contained in at most one cycle. Every acyclic cactus graph is a forest. The class of cactus graphs forms a subclass of planar graphs and contains all forests.

Circular-arc graphs: A graph $G = (V, E)$ is a *circular-arc graph* if there exists a family $\mathscr{A}$ of arcs of a circle such that each vertex $v \in V$ is associated with an arc in $\mathscr{A}$, and two vertices of G are adjacent if and only if their associated arcs intersect. The pair $(G, \mathscr{A})$ is called a *circular-arc model* of G.

Proper circular-arc graphs: A circular-arc graph G with a circular-arc model $(G, \mathscr{A})$ in called a *proper circular-arc graph* if no arc of $\mathscr{A}$ properly contains another.

Circular interval graphs: Let C be a circle. A *closed interval* of C is a proper subset of C which is homeomorphic to the closed unit interval $[0, 1]$; this means in particular that every closed interval of C has two distinct endpoints. Circular interval graphs has been identified by Chudnovsky and Seymour as one of the two basic subclasses of claw-free graphs [23].

A graph $G = (V, E)$ is *circular interval* if the following conditions hold:

- there is an injective mapping φ from V to a circle C;
- there is a set $\mathscr{I}$ of closed intervals of C, none containing another, such that two vertices $u, v \in V$ are adjacent if and only if $\varphi(u)$ and $\varphi(v)$ belong to a common interval in $\mathscr{I}$.

It is known that the class of circular interval graphs is equivalent to the class of proper circular-arc graphs [23]. We call the triple $(V, \varphi, \mathscr{I})$ a *circular interval model* of G.

Linear interval graphs: Another basic subclass of claw-free graphs identified by Chudnovsky and Seymour is the class of linear interval graphs [23]. A *linear interval graph* is defined the way that circular interval graphs are defined except that we substitute "circle" by "line". Linear interval graphs are equivalent to proper interval graphs [23].

As mentioned, circular interval graphs and linear interval graphs are respectively equivalent to proper circular arc graphs and proper interval graphs, which we had already defined. However, the latter two classes are necessary because we will study fuzzy versions of circular interval graphs and linear interval graphs. These are not equivalent to any other graph classes.

Long circular interval graphs: A graph $G = (V, E)$ is a *long circular interval graph* if it has a circular interval model $(V, \varphi, \mathscr{I})$ such that no three intervals in $\mathscr{I}$ cover the entire circle C. Circular interval graphs form a superclass of both long circular interval graphs and linear interval graphs.

Fuzzy circular interval graphs: A graph $G = (V, E)$ is *fuzzy circular interval* if the following conditions hold:

- there is a mapping φ from V to a circle C (not necessarily injective); and
- there is a set $\mathscr{I}$ of closed intervals of C, none containing another, such that no point of C is an endpoint of more than one of the intervals in $\mathscr{I}$, so that
  - ▷ if two vertices $u, v \in V$ are adjacent, then $\varphi(u)$ and $\varphi(v)$ belong to a common interval in $\mathscr{I}$;
  - ▷ if two vertices $u, v \in V$ are not adjacent, then either no interval in $\mathscr{I}$ contains both $\varphi(u)$ and $\varphi(v)$, or there is exactly one interval in $\mathscr{I}$ whose endpoints are $\varphi(u)$ and $\varphi(v)$.

We call the triple $(V, \varphi, \mathscr{I})$ a *fuzzy circular interval model* of G. Fuzzy circular interval graphs form a superclass of circular interval graphs. Let us remark that *circular-arc graphs* form neither a subclass nor a superclass of fuzzy circular interval graphs: the claw is an example of a circular-arc graph that is not a fuzzy circular interval graph, whereas the complement of C_6 is known not to be a circular-arc graph.

Fuzzy linear interval graphs: A *fuzzy linear interval graph G* is defined the way that we define fuzzy circular interval graphs except that we substitute "circle" by "line". Triple $(V, \varphi, \mathscr{I})$ is called a *fuzzy linear interval model* of G where $\mathscr{I}$ is a set of intervals of a line.

The class of fuzzy linear interval graphs is clearly a subclass of fuzzy circular interval graphs. A fuzzy circular interval graph for which the mapping φ is injective is a circular interval graph, and hence a proper circular-arc graph [23]. Therefore fuzzy circular interval graphs form a superclass of circular interval graphs. Similarly, a fuzzy linear interval graph for which the mapping φ is injective is a linear interval graph.

2.3 Algorithmic framework

We assume that the reader is familiar with the basic concepts of complexity theory. We begin with the definition of decision problems and we briefly introduce the complexity classes P and NP. Then we introduce other types of problems that we will be dealing with in this thesis.

2.3.1 Decision problems and their complexity

A *problem* is an infinite collection of instances with a solution for each instance. For every problem $\mathcal{Q}$, there is a finite alphabet such that every instance of $\mathcal{Q}$ is encoded as a string over that alphabet.

Definition 1. *A* decision problem *is a problem for which the answer is YES or NO.*

For instance, the classical CLIQUE problem is represented as follows:

CLIQUE
Instance: A graph G and an integer k.
Question: Does G contain a clique of size at least k?

An *algorithm* is a finite set of step-by-step instructions to solve all instances of a problem. An algorithm $\mathcal{A}$ that solves a decision problem $\mathcal{Q}$ needs to determine whether x is a YES-instance of $\mathcal{Q}$ for each input x. Computational complexity of $\mathcal{A}$ is measured in terms of the input size[1] $|x|$.

P and NP

Definition 2. P *is the class of all decision problems which can be solved in polynomial time.*

For every problem $\mathcal{Q}$ in P, there exists an algorithm that decides whether each instance x is a YES-instance of $\mathcal{Q}$ in time $O(|x|^c)$ for some constant c. Therefore, problems in P are said to be *polynomial-time solvable*.

Definition 3. *A* polynomial-time algorithm *is an algorithm that runs in time $O(|x|^c)$ for some constant c, on input x.*

Definition 4. *A* certificate *of a problem $\mathcal{Q}$ is an additional information to check that an instance x is a YES-instance of $\mathcal{Q}$.*

For example, a subset C of vertices of size at least k is given as a certificate for an instance x of the CLIQUE problem. Then it is claimed that C is a clique and therefore x is a YES-instance.

Definition 5. *A* verifier *for $\mathcal{Q}$ is an algorithm that checks whether a certificate is a witness to the fact that x is a YES-instance of $\mathcal{Q}$.*

In our example above, a verifier is an algorithm that checks whether or not C is actually a clique.

Definition 6. NP *is the class of all decision problems which have polynomial time verifiers.*

Definition 7. *Let $\mathcal{Q}$ and $\mathcal{Q}'$ be two decision problems. Problem $\mathcal{Q}$ is* polynomial-time reducible *to problem $\mathcal{Q}'$ if there exists a polynomial-time algorithm $\mathcal{A}$ that transforms any instance x of $\mathcal{Q}$ to an instance x' of Q' such that*

$$x \text{ is a YES-instance of } \mathcal{Q} \iff x' \text{ is a YES-instance of } \mathcal{Q}'.$$

Algorithm $\mathcal{A}$ is called a polynomial-time reduction *of $\mathcal{Q}$ to $\mathcal{Q}'$.*

Definition 8. *A problem $\mathcal{Q}$ is* NP-hard *if every problem in NP is polynomial-time reducible to $\mathcal{Q}$.*

Definition 9. *A problem is* NP-complete *if it belongs to NP and it is NP-hard.*

The Cook-Levin theorem proves that SATISFIABILITY is an NP-complete problem [96]. To prove that a problem $\mathcal{Q}$ is NP-hard it suffices to prove that there is a polynomial-time reduction of an NP-complete problem to $\mathcal{Q}$, because of the following:

If problem $\mathcal{Q}$ is polynomial-time reducible to problem $\mathcal{R}$, and $\mathcal{R}$ is polynomial-time reducible to problem $\mathcal{S}$, then $\mathcal{Q}$ is polynomial-time reducible to $\mathcal{S}$. Therefore polynomial-time reductions are closed under composition.

[1] The size of an input is the length of the encoding of the input. For a graph as input, normally, the number of vertices plus the number of edges is considered as its size.

Theorem 2.3.1. *[96] If a problem $\mathcal{Q}$ is polynomial-time reducible to a problem $\mathcal{Q}'$ and $\mathcal{Q}'$ belongs to P, then $\mathcal{Q}$ belongs to P.*

Theorem 2.3.1 guarantees that if an NP-hard problem is solved in polynomial time then P = NP.

We saw that the answer for instances of decision problems is simply YES or NO. However, there are other types of problems whose solutions are not that simple.

2.3.2 Optimization problem

Optimization problems are problems whose solutions need to minimize or maximize a desired value. In an optimization problem, the output is either the optimal value or a solution with the optimal value. For instance, the following are considered as two variants of the MAXIMAL CLIQUE problem where in the first one we ask for the size of a maximum clique and in the second we ask for a clique of maximum size.

MAXIMUM CLIQUE
Instance: A graph G.
Output: The size of a maximum clique in G.

MAXIMUM CLIQUE
Instance: A graph G.
Output: A clique of maximum size in G.

Notice that both problems CLIQUE and MAXIMUM CLIQUE are about cliques in graphs and therefore they are decision and optimization variations of the same problem. It is easy to see that the decision version of every problem is reducible to its optimization version in polynomial time. Therefore if the decision version of a problem is NP-complete then its optimization version is NP-hard.

2.3.3 Enumeration problem

Listing all elements of a family of objects satisfying a specified property without any repetition is called *enumeration*. Usually, an infinite collection F of finite families F_i, where i belongs to an infinite index set I, is given and we are asked to list all elements of F_i for every $i \in I$. An *enumeration algorithm* for F is an algorithm that enumerates all elements of F_i, for every $i \in I$. A problem whose solution is a list of elements of a finite family of objects is called an *enumeration problem*. The following represents an example of an enumeration problem.

Enumerate all maximal cliques
Instance: A graph G.
Output: All maximal cliques in G.

For the enumeration problem above, we define the index set I to be the set of all graphs; and for every $G \in I$, we define F_G to be the family of all maximal cliques of

G. An enumeration algorithm that solves this problem enumerates elements of F_G for every $G \in I$.

Enumeration algorithms are also useful in solving the appropriate decision and optimization problems. The best known algorithms for many optimization problems are enumeration algorithms. Let $\mathcal{Q}$ be an optimization problem. Again define the index set I to be the set of all graphs. Let F_G be the family of all feasible solutions of G, for every $G \in I$. Suppose $\mathcal{A}$ is an enumeration algorithm that lists all elements of F_G for all $G \in I$. Indeed, $\mathcal{A}$ lists all feasible solutions of G for every input graph G of Q. An algorithm $\mathcal{A}'$ that solves Q, for every input graph G, devises $\mathcal{A}$ as subroutine to list all feasible solutions of G and then picks a best solution among the listed items. We will study enumeration algorithms in Chapter 4.

2.3.4 Some examples of problems

In this section we introduce some of the well-known graph problems that we mention within Part I.

MAXIMUM INDEPENDENT SET
Instance: A graph G.
Output: The size of a largest independent set in G.

Dominating set: A *dominating set* of a graph G is a subset S of its vertices such that every vertex which is not in S has a neighbor in S. The following defines the optimization variant of the dominating problem which is a classical NP-hard problem [59].

MINIMUM DOMINATING SET
Instance: A graph G.
Output: The size of a dominating set of minimum size in G.

Set cover: Let $U = \{x_1, x_2, \ldots, x_n\}$ be a set and let $S = \{S_1, S_2, \ldots, S_m\}$ be a collection of subsets of U. A sub-collection C of S is called a *set cover* of U if the union of the elements of C covers U. The problem of determining a set cover with the minimum number of subsets is an NP-hard problem [74].

MINIMUM SET COVER
Instance: A subset U together with a collection S of its subsets.
Output: The size of a set cover of U of minimum size.

Vertex cover: A *vertex cover* in a graph G is a subset S of its vertices such that every edge of G is incident with at least one vertex in S. The problem of finding the size of a minimum vertex cover is known to be NP-hard [74]. The optimization variant of the problem is as follows:

MINIMUM VERTEX COVER
Instance: A graph G.
Output: The size of a vertex cover of minimum size in G.

Metric dimension: Let G be a graph. A subset R of vertices of G is called a *resolving set* for G if for every pair of vertices u and v in G, there is a vertex w in R such that $d(w,u) \neq d(w,v)$. The cardinality of a smallest resolving set for G is called the *metric dimension* of G. The METRIC DIMENSION problem is NP-complete on general graphs [59]. It is also NP-complete on split graphs and bipartite graphs [41].

METRIC DIMENSION
Instance: A graph G.
Output: The cardinality of a smallest resolving set for G.

Hamiltonian cycle: A *Hamiltonian cycle* in a graph G is a cycle that visits each vertex of G exactly once. Deciding whether or not a given graph has a Hamiltonian cycle is one of the most famous NP-complete problems in graph theory [74].

HAMILTONIAN CYCLE
Instance: A graph G.
Question: Does G contain a Hamiltonian cycle?

Traveling salesman: Assume that a list of cities together with the distances between each pair of the cities are given. What is the length of the shortest route that visits all the cities and each city exactly once? TRAVELING SALESMAN is one of the most famous optimization problems in graph theory which has many practical applications. NP-completeness of HAMILTONIAN CYCLE implies the NP-hardness of the TRAVELING SALESMAN problem.

TRAVELING SALESMAN
Instance: A graph G.
Output: A shortest cycle that visits each vertex once.

Graph coloring: By a simple modification of depth first search, 2-color-ability of a graph can be recognized in polynomial time. However, determining the chromatic number of a given graph is an NP-hard problem [58]. Even 3-color-ability of a graph is known to be an NP-complete problem; and it remains NP-complete even on planar 4-regular graphs [35].

GRAPH COLORING:
Instance: A graph G.
Output: Determine the chromatic number of G.

Satisfiability (SAT): A *Boolean variable* is a variable with two possible values such as TRUE and FALSE, YES and NO, or 0 and 1. A *binary logical operator* is an operator that assigns a Boolean value to two operands with Boolean values. *AND* or *conjunction* is a binary logical operator that assigns TRUE if and only if both variables have the value TRUE. Normally $\wedge$ is used to denote the conjunction operator. *OR* or *disjunction* is another binary logical operator that assigns FALSE if and only if both variables are FALSE. We use $\vee$ to indicate the disjunction operation. Negation, which is denoted normally by $\neg$, is a unary operation that negates the value of a Boolean variable, i.e., if the variable is YES it returns NO and vice versa. Negation of a variable x is also denoted by $\bar{x}$.

Definition 10 (Boolean formula)**.** *[17] Consider the alphabet* $\{\wedge, \vee, \neg, 0, 1, (,)\}$. *The* Boolean formulas *are defined as follows:*

- 0 *and* 1 *are Boolean formulas.*
- *If* α *and* β *are Boolean formulas, then* $\neg\alpha$*,* $\alpha \wedge \beta$ *and* $\alpha \vee \beta$ *are Boolean formulas.*

A *truth assignment* of a Boolean formula is an assignment of values TRUE and FALSE to the variables of the formula. A Boolean formula is *satisfiable* if there is a truth assignment that makes the formula TRUE; and the assignment is called a *satisfying truth assignment*.

A *literal* is a Boolean variable or negation of a Boolean variable, e.g., x or $\bar{x}$. A *clause* is disjunctions of several literals; for instance $(\bar{x_1} \vee x_2 \vee x_3)$ is a clause with three literals. A Boolean formula is said to be in *conjunctive normal form* if it is conjunctions of several clauses. Such a formula is simply called a *CNF* formula. The following is an example of a CNF formula

$$(\bar{x_1} \vee x_2 \vee x_3) \wedge (x_3 \vee \bar{x_4} \vee \bar{x_5} \vee x_6 \vee \bar{x_7}) \wedge (x_2 \vee x_3).$$

A CNF formula is called *3-CNF* if each clause contains three literals.

SATISFIABILITY (SAT)
Instance: A CNF formula φ.
Question: Is φ satisfiable?

SATISFIABILITY is a well-known NP-complete problem [96].

Chapter 3

Coping with NP-hardness

The *running time* of an algorithm is the maximum number of steps required by the algorithm to solve any instance of a problem. Problems in P are considered efficient, whereas for NP-hard problems we have so far only algorithms that run in exponential time. For instance, MAXIMUM INDEPENDENT SET is one of those problems. A brute-force way to solve this problem is to check all subsets of vertices. Since every graph on n vertices has 2^n such subsets, this algorithm takes $O(n^2 \cdot 2^n)$ time to return the solution. We normally replace O with O^* to suppress the polynomial factor. So we write $O^*(2^n)$ instead of $O(n^2 \cdot 2^n)$. Such problems are of one of the following types. The following is adapted from [53].

- *Subset problems:* feasible solutions of these problems are subsets of a ground set. If the ground set is of cardinality n then the brute-force search is done in $O^*(2^n)$ time. MAXIMUM INDEPENDENT SET is an instance of subset problems.

- *Permutation problems:* every feasible solution of a permutation problem is an ordering of elements of a ground set. When the size of the ground set is n then the number of feasible solutions are around $n!$ and the trivial algorithm that examines all the possible orderings takes $O^*(n!)$ time. An example of these problems is the problem of TRAVELING SALESMAN where every tour is an ordering of the vertex set of the given graph.

- *Partition problems:* every feasible solution of these problems are described by a partition of a ground set. GRAPH COLORING problem is an instance of partition problems where we want to partition the vertex set into some color classes. If the ground set is a set of size n then a brute-force algorithm considers all the possible partitions of the set and therefore it takes $O^*(n^n)$ time.

Since many interesting and applied problems in graph theory are NP-hard, we need to cope with these problems in a way. Unless P $\neq$ NP, we cannot expect that an NP-hard problem can be solved optimally, for all instances, and in polynomial time. Relaxing each of these requirements gives different ways to cope. Approximation algorithms, algorithms that solve a problem on restricted instances, and exact exponential time algorithms are three techniques corresponding to these relaxations. In the following sections we explain these three techniques together with another technique which is known as fixed-parameter tractability.

3.1 Approximation algorithms

One way to deal with an NP-hard optimization problem is to relax the requirement that asks for an optimal solution. In practice, an approximation of the best solution might be acceptable for many applications. Therefore, designing an algorithm that finds a nearly optimal solution for every instance of an NP-hard optimization problem in polynomial time is an established research line to cope with these problems. We focus only on constant-factor approximation algorithms.

Definition 11. *[110] An* α-approximation algorithm *for an optimization problem is a polynomial-time algorithm that for all instances of the problem produces a solution whose value is within a constant factor of* α *of the value of an optimal solution.*

The constant α is said to be the *performance guarantee* of an α-approximation algorithm and all algorithms which are α-approximation are known as *constant-factor* algorithms.

An optimization problem that minimizes a value, like MINIMUM VERTEX COVER, is said to be a *minimization problem*. The performance guarantee of a minimization problem is strictly larger than 1. For example a 2-approximation algorithm is an approximation algorithm that for every instance of a minimization problem returns in polynomial time a solution whose value is at most two times the optimal value. For instance, the following is a 2-approximation algorithm to solve MINIMUM VERTEX COVER problem.

Algorithm VC.
1. $S \leftarrow \emptyset$
2. **while** G is nonempty **do**
3. select an arbitrary $uv \in E$
4. $S \leftarrow S \cup \{u, v\}$
5. delete all edges incident with u or v from G
6. **return** S

Let $vc(G)$ be the size of an optimal vertex cover of G. To show that the algorithm is actually a 2-approximation algorithm to solve MINIMUM VERTEX COVER, we need to show that it returns a vertex cover of G and the number of elements in the returned set is at most $2 \cdot vc(G)$. Since S contains both endpoints of the selected edges by the algorithm and at least one endpoint of the removed edges, it forms a vertex cover for G. Let us show that $|S| \leq 2 \cdot vc(G)$. Note that the selected edges, the edges whose both endpoints have been added to S, form a matching M of G. Therefore, every vertex cover of G needs to contain at least one endpoint of each edge in M. It implies that every vertex cover is of size at least $|S|/2$ and thus $vc(G) \geq |S|/2$. This implies that $|S| \leq 2 \cdot vc(G)$.

Similarly, an optimization problem that aims to maximize a value, like MAXIMUM INDEPENDENT SET, is called a *maximization problem*. Performance guarantees for maximization problems are strictly less than 1. Therefore, a $1/2$-approximation algorithm is an approximation algorithm that for every instance of a maximization problem returns in polynomial time a solution whose value is at least half of the optimal value.

However, there are NP-hard optimization problems for which there exist no polynomial-time constant-factor approximation algorithms, unless P = NP. TRAVELING SALESMAN is one of those problems.

Theorem 3.1.1. *For any constant α, the problem of approximating* TRAVELING SALESMAN *with factor α is NP-hard.*

Proof. We present a polynomial time reduction from the NP-complete problem HAMILTONIAN CYCLE to the problem of α-approximating TRAVELING SALESMAN. This would imply that if we are able to α-approximate TRAVELING SALESMAN in polynomial time then we would be able to solve HAMILTONIAN CYCLE in polynomial time and therefore P = NP.

For a given input $G = (V, E)$ to HAMILTONIAN CYCLE, we construct a weighted graph $G' = (V', E')$. Graph G' is a complete graph on vertex set V. We define $w(e) = 1$ for every edge $e \in E \cap E'$, and for every $e \in E' \setminus E$ we define its weight to be a very large number L ($> \alpha n$). Let $\mathcal{A}$ be a polynomial-time α-approximation algorithm for the TRAVELING SALESMAN problem. The following algorithm will solve the HAMILTONIAN CYCLE problem in polynomial time.

Algorithm HC.
1. Construct $G' = (V', E')$ as described above.
2. **if** $\mathcal{A}(G')$ returns a tour of cost $\leq \alpha n$ **then**
3.   **return** YES
4. **if** $\mathcal{A}(G')$ returns a tour of cost $\geq L$ **then**
5.   **return** NO

□

3.2 Restricted instances

Despite NP-hardness of a problem in general, there might exist algorithms that solve the problem for instances with some specified properties in polynomial time. Hence, one approach to deal with NP-hard problems is to restrict instances of these problems, i.e., instead of solving all the instances of an NP-hard problem we just solve the instances that satisfy some specified properties. For example MINIMUM VERTEX COVER, which is an NP-hard problem, is solved in linear time when it is restricted to the class of trees.

Theorem 3.2.1. MINIMUM VERTEX COVER *restricted to trees is solvable in linear time.*

Proof. Let T be a tree on n vertices. Root T at a vertex v. Let u be a vertex of T. We denote the subtree of T_v below u by T_u and we define $vc(u)$ to be the size of the minimum vertex cover of T_u. If u is a leaf, then the subtree rooted at u has no edges, and hence $vc(u) = 0$. The crucial observation is the following: if a vertex cover does not include a vertex, then it has to include all of its neighboring vertices. Therefore, for every internal vertex x we have the following

$$vc(x) = min\{\sum_{y:(x,y)\in E(T_x)} (1 + \sum_{z:(y,z)\in E(T_y)} vc(z)), 1 + \sum_{y:(x,y)\in E(T_x)} vc(y)\}.$$

The algorithm can then solve all the subproblems starting from leaves and processing every vertex as soon as its children are processed. The minimum vertex cover of T is the value $vc(v)$ which is the last in the ordering. Since to calculate all the values every edge of T is checked two times, the running time is linear in n. □

There are NP-hard graph problems that remain NP-hard on many restricted classes of graphs. Split graphs and bipartite graphs are structured classes of graphs on which many hard problems become polynomial or even trivial. INDEPENDENT SET and VERTEX COVER are among those problems [42, 54, 59]. The DOMINATING SET problem remains NP-hard on both these classes. The following reduction from SET COVER to DOMINATING SET restricted to split graphs shows the NP-hardness of the latter [8].

Theorem 3.2.2. *[8]* DOMINATING SET *restricted to split graphs is NP-hard.*

Proof. Given a set $U = \{x_1, x_2, \ldots, x_n\}$ and a collection of its subsets $S = \{S_1, S_2, \ldots, S_m\}$ as an instance of the SET COVER problem, we create the following instance of the DOMINATING SET problem restricted to split graphs in polynomial time. Note that without loss of generality we may assume that all the subsets S_i, $1 \le i \le m$, are non-empty and $\bigcup_{i=1}^{m} S_i = U$.

We construct a split graph G with split partition (C, I) where $C = S$ and $I = U$. The edge set of G is

$$E(G) = \{S_iS_j \,|\, S_i, S_j \in S \text{ and } i \neq j\} \cup \{x_iS_j \,|\, x_i \in U, S_j \in S \text{ and } x_i \in S_j\}.$$

$$\begin{aligned} U &= \{x_1, x_2, x_3, x_4, x_5, x_6\} \\ S &= \{S_1, S_2, S_3, S_4\} \\ S_1 &= \{x_1, x_2, x_3\} \\ S_2 &= \{x_1, x_4\} \\ S_3 &= \{x_2, x_5, x_6\} \\ S_4 &= \{x_4, x_5, x_6\} \end{aligned} \quad \Longrightarrow$$

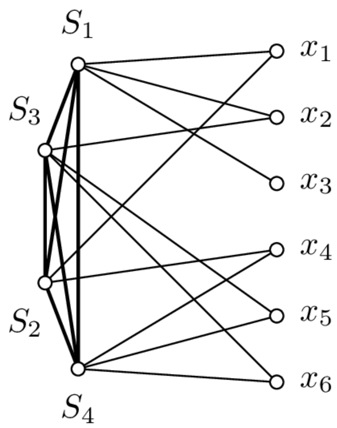


Figure 3.1: An example of the reduction from the SET COVER problem to the problem of DOMINATING SET restricted to split graphs.

The assumption $\bigcup_{i=1}^{m} S_i = U$ implies that G is a connected graph. Moreover, every vertex in C has at least one neighbor in I since we have assumed that $S_i \neq \emptyset$ for all i. It is readily verified that the instance of the SET COVER problem has a solution of size at most k if and only if G has a dominating set of size at most k. □

Having proved that DOMINATING SET is NP-hard on split graphs, the following theorem shows that the problem is NP-hard on bipartite graphs as well.

Theorem 3.2.3. *[8]* DOMINATING SET *restricted to bipartite graphs is NP-hard.*

Proof. Let G be an instance of DOMINATING SET problem restricted to split graphs and let (C,I) be its split partition. Without loss of generality we assume that G has no isolated vertices, I is not an empty set and every vertex in C has at least a neighbor in I. We create an instance of the same problem restricted to bipartite graphs by constructing a bipartite graph $G' = (A,B,E')$ in polynomial time. Set A is $I \cup \{a\}$, set B is $C \cup \{b\}$ and

$$E' = \{uv \in E(G) \,|\, u \in I \text{ and } v \in C\} \cup \{av \,|\, v \in C \cup \{b\}\}.$$

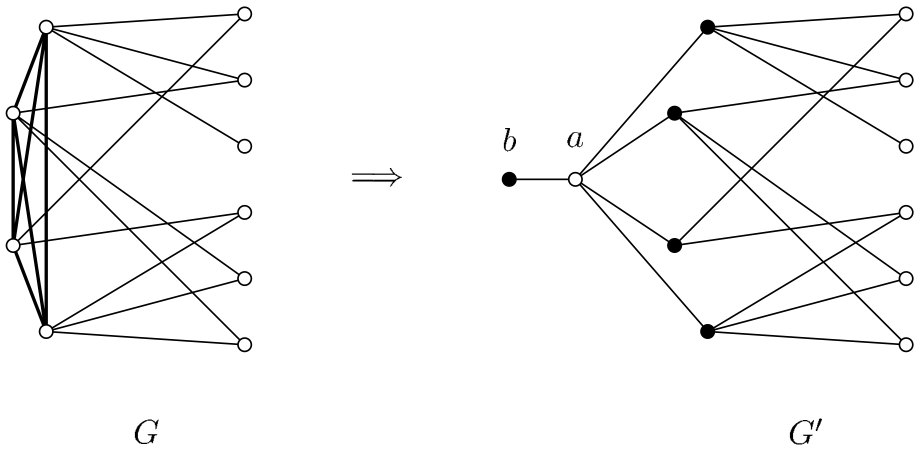


Figure 3.2: An example of the reduction from the problem of DOMINATING SET restricted to split graphs to the problem of DOMINATING SET restricted to bipartite graphs.

Let us now show that G has a dominating set of size at most k if and only if G' has a dominating set of size at most $k+1$. The crucial observation is the following: at least one of the vertices a and b must belong to any dominating set of G' as otherwise vertex b is not dominated. First assume that G has a dominating set D of size k. It is obvious that $D' = D \cup \{a\}$ is a dominating set in G'. For the other way around, let us assume that D' is a dominating set of size $k+1$ in G'. We show that $D' \setminus \{a,b\}$ is a dominating set in G. If no vertices from C belong to D' then it holds that $I \subseteq D'$ and thus $D' \setminus \{a,b\}$ is a dominating set for G (recall the assumption on G). In the case that some vertex from C belongs to D', that vertex dominates all the vertices of C in G and those vertices of $D' \setminus \{a,b\}$ that dominate vertices of I in G' do the same in G. $\square$

When a problem remains NP-hard on a graph class, then concentrating on subclasses of that class, which are more restricted and structured, might be a fruitful way for further research. Metric dimension is another NP-hard problem which remains NP-hard on both split graphs and bipartite graphs [41]. In Paper I of Part II [48], we focus on a subclass of bipartite graphs namely chain graphs and we manage to solve the problem on these graphs in linear time. This result even yields a formula to compute this number on twin-free chain graphs.

3.3 Exact exponential-time algorithms

When an optimal solution is needed for an NP-hard optimization problem, and we need to get a solution for every instance of the problem then we must ignore the requirement of polynomial running time, unless P = NP. So another line of research to deal with NP-hard optimization problems is designing exponential-time algorithms that find the optimal solutions. The main line of research is to design algorithms that avoid brute-force. One of the powerful algorithmic techniques to design exact exponential-time algorithms is branching.

Branching

A branching algorithm is simply a recursive algorithm that searches through all feasible solutions of a problem. A typical branching algorithm mainly comprises two collections of rules, *reduction rules* and *branching rules*. Every rule generates subproblems[1] of smaller size that are solved recursively. A reduction rule generates a single subproblem, whereas a branching rule generates at least two subproblems, which results in the exponential nature of these algorithms.

To give a very simple example with a single branching rule, let us study how to generate all vertex subsets of a graph:

Algorithm ALLSUBSETS(G,S).
if G is empty **then**
  return S
else
  pick a vertex v of G
  ALLSUBSETS$(G-v,S)$
  ALLSUBSETS$(G-v,S\cup\{v\})$

The initial call to the algorithm is then ALLSUBSETS$(G,\emptyset)$. For every branching step, the algorithm will generate in the one branch all vertex subsets of G containing S and v, and in the other branch vertex subsets of G containing S but not containing v. This is why we view the computation of such an algorithm as a rooted tree, called the *search tree*. The following has been adapted from [53].

The *search tree* is a tool to illustrate an execution of a branching algorithm which is also useful in analyzing the running time of the algorithm. That is a rooted tree which is obtained as follows: the root vertex is assigned to the input of the problem; recursively a child of a vertex is assigned to a subproblem which is created by applying a branching rule to the instance of that vertex. Each subproblem which is reached by a branching rule is simplified by the reduction rules if they are applicable, but no new vertex is created by applying the reduction rules. Normally, the leaves of a search tree are the feasible solutions of the problem which is solved by the corresponding branching algorithm. Therefore, the number of leaves of a search tree is an upper bound for the number of solutions or enumerated objects. Other than this, the number of leaves of a search tree is important in calculating the running time of the branching algorithm.

Let us see how the search tree of a branching algorithm and its number of leaves are used to analyze the running time of the algorithm. Let us assume that our branching algorithm spends polynomial time on each vertex of the search tree during its execution. It is clear that the total running time of the algorithm depends on the number of vertices

[1] A *subproblem* of an instance of a problem is an instance of the same problem of smaller size.

of the search tree. It is easy to see that every vertex of a search tree other than the leaves, and maybe the root, has degree at least 3. It is also easy to show that in a tree, the number of leaves is bigger than the number of vertices of degree at least 3. Hence, the number of all vertices of a search tree is a constant factor times the number of leaves of that tree. Therefore, to analyze the running time of a branching algorithm, it suffices to upper bound the number of leaves of the corresponding search tree. Let us show how to upper bound the number of leaves in a search tree.

Let $T(n)$ be the total number of leaves of a search tree of an instance of size n. Generally, analyzing the running time of every branching rule is done separately. For every branching rule, we assume that the algorithm does only this branching rule during its execution. Then we calculate the maximum number of leaves that this assumption would yield. Let's assume that for every branching rule we have calculated the maximum number of leaves. Since a mixture of all the branching rules are applied during an execution of the algorithm, the actual number of leaves is upper bounded by the maximum of these numbers.

Let us assume that b is a branching rule, I is an instance of a problem of size n and we only apply branching rule b. Let $n-s_1, n-s_2, \ldots, n-s_r$ be the sizes of r new subproblems created by applying b to I. The vector $(s_1, s_2, \ldots, s_r)$ is known as the *branching vector* of branching rule b. The number of leaves that recursively applying of branching rule b to the vertices of the search tree creates is upper bounded by the following linear recurrence formula

$$T(n) \leq T(n-s_1) + T(n-s_2) + \cdots + T(n-s_r).$$

Techniques to solve the linear recurrence relations are applied to calculate $T(n)$. Solution of the relation above is of the form α^n, where α (it is customary to round up α at the fourth digit) is the unique positive real root of the following equation:

$$x^n - x^{n-s_1} - x^{n-s_2} - \cdots - x^{n-s_r} = 0.$$

This unique positive real root is called the *branching factor* of branching rule b. Without loss of generality, assume that $s_1 \leq s_2 \leq \cdots \leq s_r$. Hence, instead of the formula above one can simplify it to the following formula

$$x^{s_r} - x^{s_r - s_1} - x^{s_r - s_2} - \cdots - 1 = 0,$$

by dividing both of the sides by x^{n-s_r}; note that $x \neq 0$.

Let us give an example. Assume that $\mathcal{Q}$ is a problem and assume that there is a branching algorithm that solves $\mathcal{Q}$. Suppose b_1, b_2 and b_3 are three branching rules of the algorithm. Let I be an arbitrary instance of $\mathcal{Q}$ of size n. The branching rule b_1 branches I into three new instances of sizes $n-3, n-2$ and $n-4$. Rule b_2 branches I into two new instances of sizes $n-3$ and $n-3$ and branching rule b_3 branches the instance into two instances $n-2$ and $n-4$. Let us calculate the running time of this algorithm. Branching vectors of b_1, b_2 and b_3 are respectively, $(3,2,4)$, $(3,3)$ and $(2,4)$. The three obtained linear recurrence formulas are the following

- $T(n) \leq T(n-3) + T(n-2) + T(n-4)$,
- $T(n) \leq T(n-3) + T(n-3)$,

- $T(n) \leq T(n-2) + T(n-4)$;

and the formulas to be solved are

- $x^4 - x - x^2 - 1 = 0$, the solution of which gives $x \approx 1.4655$,
- $x^3 - 2 = 0$, the solution of which gives $x \approx 1.2597$,
- $x^4 - x^2 - 1 = 0$, the solution of which gives $x \approx 1.2720$.

Since $1.4655, 1.2597$ and 1.2720 are respectively the branching factors of b_1, b_2, and b_3, the maximum number of leaves of the search tree is at most 1.4655^n and therefore running time of the algorithm is $O^*(1.4655^n)$ or $O(1.4656^n)$. It is also obvious and useful to know that the function $T(n)$ is an increasing function.

We continue with some useful properties of the branching vectors and the branching factors. In the following two lemmas $\tau(s_1, s_2, \ldots, s_r)$ denotes the branching factor of branching vector $(s_1, s_2, \ldots, s_r)$.

Lemma 3.3.1. *[53] Let $r \geq 2$. Let $s_i > 0$ for all $i \in \{1, 2, \ldots, r\}$. Then the following properties hold.*

- $\tau(s_1, s_2, \ldots, s_r) > 1$.
- $\tau(s_1, s_2, \ldots, s_r) = \tau(s_{\pi(1)}, s_{\pi(2)}, \ldots, s_{\pi(r)})$ *for any permutation* π.
- *If* $s_1 > s'_1$ *then* $\tau(s_1, s_2, \ldots, s_r) > \tau(s'_1, s_2, \ldots, s_r)$.

Lemma 3.3.2. *[53] Let i, j and k be positive integers.*

- $\tau(k, k) \leq \tau(i, j)$ *for all branching vectors* (i, j) *satisfying* $i + j = 2k$.
- $\tau(i, j) > \tau(i + \varepsilon, j - \varepsilon)$ *for all* $0 < i < j$ *and all* $0 < \varepsilon < \frac{j-i}{2}$.

Adding branching vectors can also be useful in analyzing of branching algorithms. Assume that, during the execution, whenever a branching rule branches an instance of size n into (i, j), i.e., it creates an instance of size $n - i$ on the left side and an instance of size $n - j$ on the right side, it branches the left side instance into (k, l) right after. We can combine these two branchings into one branching that branches the instance of size n into three subproblems. It creates on the left side an instance of size $n - (i + k)$, in the middle an instance of size $n - (i + l)$ and on the right side an instance of size $n - j$. Hence the combined branching vector is $(i + k, i + l, j)$. Similarly, whenever a branching algorithm branches an instance into (i, j), if it branches the left side into (k, l) and the right side into (k', l'), then the combined branching rule is equivalent to a rule that branches into $(i + k, i + l, j + k', j + l')$. [53]

3.4 FPT algorithms

Parameterized complexity studies the different behaviors of NP-complete problems. Even though two problems are both NP-hard, one can have a much faster (exponential-time) algorithm than the other. Indeed in parameterized complexity, the complexity of a problem is measured as a function in terms of a parameter. This provides a finer scale than the classical setting to classify the NP-hard problems, where in the classic setting the only measure to classify the problems is the length of inputs. Informally, a parameterized algorithm tries to limit the exponential part of the running time to a small part of the input.

Definition 12. *A* parameterized problem *is a decision problem whose input can be partitioned into two parts: a main part and a parameter. Every input of a parameterized problem is denoted by (x,k) where x is the main part of the input and the nonnegative integer k is the parameter.*

For instance, the following is a parameterized version of the classical CLIQUE problem where the parameter is the solution size.

k-CLIQUE
Instance: A graph G.
Parameter: k.
Question: Does G contain a clique of size at least k?

An algorithm $\mathcal{A}$ to solve a parameterized problem $\mathcal{Q}$ needs to determine whether (x,k) is a YES-instance of $\mathcal{Q}$ for each input (x,k). Algorithm $\mathcal{A}$ is called a *parameterized algorithm* if its computational complexity is measured in terms of the input size $|x|$ and the value of the parameter.

Definition 13. *A parameterized problem $\mathcal{Q}$ is said to be* fixed-parameter-tractable *(or* FPT*) with respect to the parameter if there exists an algorithm $\mathcal{A}$ that solves each pair (x,k) in time $f(k)|x|^c$ for some constant c.*

The class of all fixed-parameter tractable problems is called *FPT* and algorithm $\mathcal{A}$ in the definition above is known as a *fixed-parameter-tractable algorithm* (or *FPT algorithm*).

Definition 14. *A parameterized problem $\mathcal{Q}$ has a* kernel *if there is an algorithm that transforms each instance (x,k) into an instance (x',k') in time $(|x|+k)^c$, for some constant c, such that $(x,k) \in \mathcal{Q}$ if and only if $(x',k') \in \mathcal{Q}$; and $|x'|+k' \leq g(k)$ for some function g. Function g is normally an exponential function of k.*

The algorithm in the definition above is known as a *kernelization algorithm* or in short a *kernel* for the parameterized problem.

Theorem 3.4.1. *[39] Let $\mathcal{Q}$ be a decidable parameterized problem. Then $\mathcal{Q} \in FPT$ if and only if $\mathcal{Q}$ has a kernel.*

Proof. First assume that $\mathcal{Q} \in$ FPT. Hence there is an algorithm $\mathcal{A}$ that decides $\mathcal{Q}$ in time $f(k)n^c$, where f is a computable function, n is the input size, and c is a constant. Let

(x,k) be an arbitrary instance of $\mathcal{Q}$ of size n. Run algorithm $\mathcal{A}$ only n^{c+1} steps. If $\mathcal{A}$ terminates, then output the trivial answer YES or NO as instances of size $O(1)$. In the case that $\mathcal{A}$ does not terminate, it holds that $n \leq f(k)$ and therefore we can output (x,k) itself. Hence $\mathcal{Q}$ has a kernel.

Let us now assume that $\mathcal{Q}$ has a kernel. Therefore there is a kernelization algorithm $\mathcal{A}$. We first run $\mathcal{A}$ on the input of $\mathcal{Q}$ and create a reduced instance of $\mathcal{Q}$. Since $\mathcal{Q}$ is decidable, we can run any decision algorithm on the created instance. From the fact that the size of the reduced instance is upper bounded by a function of the parameter k and $\mathcal{A}$ is a polynomial-time algorithm, the combination of these two algorithms is an FPT algorithm. □

Theorem 3.4.1 shows that a kernelization algorithm can be turned into an FPT algorithm directly; first building a smaller instance of the problem by applying the kernelization algorithm and then solving the smaller instance by brute force. This means that smaller kernels would yield faster FPT algorithms. Therefore the aim in kernelization is to get a kernel which is as small as possible in size. Indeed, kernelization can be considered as a polynomial preprocessing before attacking the problem.

Let us give as an example a kernelization of the k-VERTEX COVER problem. We present two reduction rules. When the rules are not applicable anymore, we prove that the size of the instance is bounded by a function of k. For the instance (G,k) of k-VERTEX COVER

Rule 1: If v is an isolated vertex then $(G-v,k)$.

Rule 2: If $d(u) > k$ then $(G-v,k-1)$.

If neither Rule 1 nor rule 2 is applicable then one of the following cases happens:

- If $|v(G)| > k(k+1)$ then the answer is NO.
- Otherwise, $|V(G)| \leq k(k+1)$ which means that there is a $k(k+1)$ vertex kernel.

There are smaller kernels for k-VERTEX COVER problem which require more and smarter reduction rules.

If the function g in the definition of a kernel, which upper bounds the size of the new instance, is polynomial then the kernel is said to be a *polynomial kernel*. In particular, if g is a linear function then the kernel is called *linear*. As we are interested in small kernels, therefore linear kernels and kernels of polynomial size are of particular importance. However, there are some parameterized problems that do not admit polynomial kernel, under widely believed complexity theoretical assumptions. To show that a problem does not admit a polynomial kernel, some techniques such as *AND*-composition, *OR*-composition, *cross*-composition or transformation exist [11, 86].

Parameterized complexity

We are now about building the complexity theory of parameterized problems. Note that if non-parameterized form of a problem is solvable in polynomial time then parameterized form of the problem with respect to any parameter is FPT and thus we can deduce that FPT contains P. No FPT algorithms are known for k-INDEPENDENT SET and k-CLIQUE when k is the solution size. We do not claim that these problems are not in FPT, as otherwise having $P \subseteq FPT$ would imply that these problems are not in P and therefore we have proved that P $\neq$ NP. Informally speaking we are about to define some bigger class of parameterized problems that contains FPT resembling the class NP and its relation to P.

To define this class we need first to define a special kind of polynomial reduction between parameterized problems. Such a reduction between two parameterized problems $\mathcal{Q}$ and $\mathcal{Q}'$ is supposed to guarantee that if one of the problems is solved in FPT time then the other is solved in FPT time.

Definition 15 (Parameterized reduction)**.** *A parameterized reduction from problem $\mathcal{Q}$ to problem $\mathcal{Q}'$ is a function $\varphi : \mathcal{Q} \longrightarrow \mathcal{Q}'$ such that*

- *$\varphi((x,k))$ is computed in FPT time $f(k)|x|^c$, where c is a constant,*
- *$\varphi((x,k))$ is a YES-instance of $\mathcal{Q}'$ if and only if (x,k) is a YES-instance of $\mathcal{Q}$,*
- *parameter of $\varphi((x,k))$ is bounded by a function of k.*

To see the difference between standard polynomial reductions, and parameterized reductions, observe that the following reduction is a polynomial reduction but it is not a parameterized reduction.

Example 3.4.1. *Consider the following polynomial reduction between k-*VERTEX COVER *and k'-*INDEPENDENT SET*: for a given instance of k-*VERTEX COVER*, (G,k), our reduction creates $(G,n-k)$ as an instance of k'-*INDEPENDENT SET.

Since $n-k$ is not bounded by a function of k then the reduction is not a parameterized reduction. In addition, we know that k-VERTEX COVER is in FPT, whereas there is no known FPT algorithm for k-INDEPENDENT SET. The following example presents a parameterized reduction.

Example 3.4.2. *Let φ be a reduction between k-*INDEPENDENT SET *and k-*CLIQUE *such that it transforms (G,k) into $(\overline{G},k)$. It is clear that it is a parameterized reduction between these two problems.*

Definition 16 (Boolean circuit)**.** *A Boolean circuit consists of input line, binary gates AND, binary gates OR, unary negation gates and output line. For every truth assignment of the inputs of a Boolean circuit, it outputs TRUE or FALSE.*

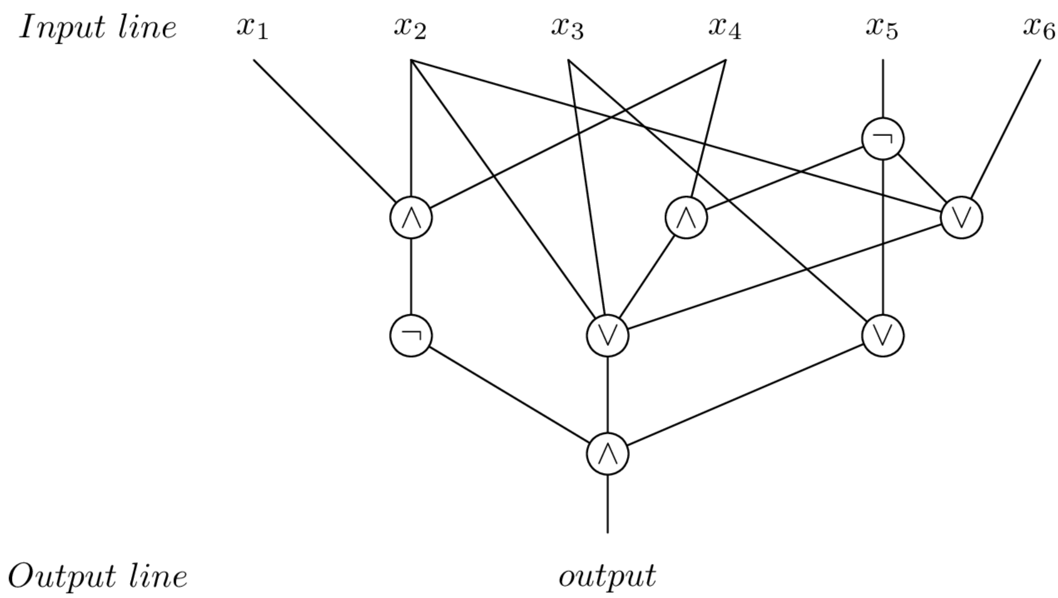


Figure 3.3: A Boolean circuit.

CIRCUIT SATISFIABILITY
Instance: A Boolean circuit C.
Question: Is there a satisfying truth assignment of the inputs of C?

The circuit in Figure 3.3 is a YES-instance of the CIRCUIT SATISFIABILITY problem as $x_1 = x_2 =$ TRUE and $x_3 = x_4 = x_5 = x_6 =$ FALSE is a satisfying truth assignment for it.

k-WEIGHTED CIRCUIT SATISFIABILITY
Instance: A Boolean circuit C.
parameter: k.
Question: Is there a satisfying truth assignment of the inputs of C with k TRUE values?

The *fan-in* of a gate is the number of inputs that come into the gate and *fan-out* of a gate is the number of gates that the gate is their input. A gate of a Boolean circuit is called *large* if its fan-in is more than some pre-agreed bound [39]. The maximum number of gates on any path from the input line to the output line of a Boolean circuit is called the *depth* of that circuit. Similarly, the *weft* of a Boolean circuit is defined to be the maximum number of large gates on any path from the input line to the output line. Let k-L(t,h) be the following parameterized problem:

k-L(t,h)
Instance: A weft t depth h Boolean circuit C.
parameter: k.
Question: Is there a satisfying truth assignment of the inputs of C with k TRUE values?

Definition 17. *We define a problem Ω to be in the class W[t] if and only if Ω is parameterized reducible to k-L(t,h) for some h.*

It is easy to see the following hierarchy

$$P \subseteq FPT \subseteq W[1] \subseteq W[2] \subseteq W[3] \subseteq \cdots \subseteq XP,$$

where XP is the class of all parameterized problems that can be solved in $O(|x|^{(f(k))})$ time. W[1] is analogue of NP for parameterized problems.

Definition 18. *A parameterized problem $\mathcal{Q}$ is* W[1]-hard *if every problem in W[1] is parameterized reducible to $\mathcal{Q}$.*

Definition 19. *A parameterized problem is* W[1]-complete *if it belongs to W[1] and it is W[1]-hard.*

W[t]-hardness and W[t]-completeness for $t \geq 2$ are defined similarly. Downey and Fellows [39] give an analogue of Cook-Levin theorem to show the W[1]-completeness of some parameterized problems. To prove that a parameterized problem $\mathcal{Q}$ is W[1]-hard, it suffices to show that there is a parameterized reduction from a W[1]-complete problem to $\mathcal{Q}$. Both k-INDEPENDENT SET and k-CLIQUE are W[1]-complete parameterized problems and k-DOMINATING SET is W[2]-complete [39]. It means that there are no FPT algorithms to solve these problems unless FPT = W[1] and FPT = W[2], respectively. Notice also that if any problem in W[1]-complete is solved in FPT time then FPT = W[1]. Similarly, if a W[2]-complete problem is in W[1] then W[1] = W[2].

Chapter 4

Enumeration and maximum number of objects in graphs

Enumeration and determining the maximum number of graph objects are two important questions in theoretical computer science. Due to their application in computer science, they have received much interest over the last decades. Enumeration has also found interest in other fields such as biology, chemistry, data mining and social networks [1, 37, 68, 70, 84]. Enumeration problems are more general than the decision and optimization problems, as every algorithm for enumeration of feasible solutions of a problem results in a solution for the appropriate decision and optimization problems. We will later see that enumeration of some objects are used to obtain faster algorithms for problems which are not related to the enumerated objects. Whether we can solve every NP-hard problem without the help of enumeration is a challenging question, asked by Gödel in 1956 [97], and we still do not have any answer to it [109]. In fact, the best known exponential-time algorithms to solve many NP-hard optimization problems list all the feasible solutions first and then pick a best solution among the listed items. Very often these algorithms provide an upper bound on the number of the objects they list. We will briefly study enumeration problems in Section 4.1.

Determining the maximum number of graph objects is an interesting combinatorial problem which has found application in computer science, especially in exact exponential-time algorithms whose running time depend on the number of specified objects [53]. The most famous classical combinatorial result of this type is due to Moon and Moser [87]. They proved that every graph on n vertices has at most $3^{n/3}$ maximal cliques and $3^{n/3}$ maximal independent sets. There are graph objects for which the best known upper bounds on their number have been obtained by enumeration algorithms. For most of the objects in graphs the best known upper and lower bounds on their number do not match. For example, the best known upper bound and lower bound on the maximum number of minimal dominating sets in a graph on n vertices are respectively, 1.7697^n and 1.5704^n [51]. When the best known upper and lower bounds do not match, we say that there is a *gap*. When there is a gap between the best known upper and lower bounds on the number of some objects, improving any of the bounds yields a narrower gap. We will later define these bounds formally. The difficulties in obtaining better upper bounds for general graphs motivates restricting the input graphs to exploit their structural properties. Even if there is no gap between the bounds, working on enumeration and maximum number of graph objects in restricted classes of graphs is an interesting line of research as it shows the different behavior of objects in graph classes. In Section 4.2, we focus on the maximum number of objects in graphs, we present some of the achieved combinatorial and algorithmic upper bounds, and we discuss the gaps in general graphs and graph classes.

4.1 Enumeration algorithms

In most of the interesting enumeration problems, the number of objects to be listed is exponential in the size of the input. There are two main approaches in designing enumeration algorithms: output-sensitive and input-sensitive.

4.1.1 Output-sensitive algorithms

The *output-sensitive* algorithms are the enumeration algorithms whose running time is measured in terms of the input size and the output size.

Definition 20 (Output-polynomial)**.** *Let I be the set of all graphs and let F_G be the family of some specified objects of G, for every graph $G \in I$. Suppose $\mathcal{A}$ is an enumeration algorithm that lists all the elements of F_G, for every $G \in I$. The running time of $\mathcal{A}$ is called* output-polynomial *if there exists a polynomial function $P(x,y)$ such that all elements of F_G are generated in time bounded by $P(||G||,|F_G|)$, where $||G||$ is the size of input graph G and $|F_G|$ is the cardinality of F_G.*

There exist several enumeration problems that can be solved by output-polynomial algorithms. However, there are some problems for which even this weak notion of polynomial time is not possible, assuming that P $\neq$ NP. Let us discuss the existence of output-polynomial algorithms a bit more in details.

Definition 21. *[79] Let $\mathcal{E}$ be a finite set of elements and let $\mathcal{I}$ be a collection of subsets of $\mathcal{E}$. If for any $I \in \mathcal{I}$ and $I' \subseteq I$ it holds that $I' \in \mathcal{I}$, then we call $(\mathcal{E},\mathcal{I})$ an* independence system *and $\mathcal{I}$ the collection of its* independent sets. *An independent set I is* maximal *if there is no independent set I' such that $I \subset I'$.*

For example the set of all vertices of a graph G together with the collection of all cliques of G form an independence system. Lawler, Lenstra and Kan [79] show that the problem of generating all maximal independent sets of an arbitrary independence system is NP-hard, meaning that if there is an output-polynomial algorithm to solve the problem then there is a polynomial-time algorithm that solves the SAT problem.

Theorem 4.1.1. *[79] If there exists an algorithm for generating all the maximal independent sets of an arbitrary independence system in output-polynomial time, then P = NP.*

Therefore, we cannot expect output-polynomial algorithm for every enumeration problem, unless P = NP. However, there are some independence systems for which there exist output-polynomial algorithms. The following is adapted from Lawler et al. [79]. Let $\mathcal{E} = \{1,\ldots,n\}$, let $\mathcal{I}$ be such that $(\mathcal{E},\mathcal{I})$ form an independence system. Let us also assume that the testing that a given subset is independent requires time c.

Suppose $\mathcal{I}_i$ is the collection of all maximal independent sets within $\{1,\ldots,i\}$. We want to construct $\mathcal{I}_i$ from $\mathcal{I}_{i-1}$ for all $1 \le i \le n$, in order to obtain $\mathcal{I}_n$. Notice that $\mathcal{I}_0 = \emptyset$ and $\mathcal{I}_n$ is the collection of all $\mathcal{L}$ independent sets that are maximal within $\mathcal{E}$.

Assume that $I \in \mathcal{I}_{i-1}$. If $I \cup \{i\} \in \mathcal{I}$, then it is clear that $I \cup \{i\} \in \mathcal{I}_i$. In the case that $I \cup \{i\} \notin \mathcal{I}$, then $I \notin \mathcal{I}_i$. Therefore,

$$|\mathcal{I}_0| \le |\mathcal{I}_1| \le \cdots \le |\mathcal{I}_n| = \mathcal{L}.$$

Notice that the elements of $\mathcal{E}$ can be numbered arbitrarily and therefore the following theorem is obtained.

Theorem 4.1.2. *[79] For any $J \subseteq \mathcal{E}$, the number of independent sets which are maximal within J does not exceed $\mathcal{L}$.*

Theorem 4.1.3. *[79] All the maximal independent sets of an independence system can be generated in time polynomial in n, c and $\mathcal{L}$, if it is possible to list in polynomial time all independent sets that are maximal within $I \cup \{i\}$, for any $I \in \mathcal{I}_{i-1}$, $i = 1,\ldots,n$.*

It is easy to see that the independence system $(\mathcal{E},\mathcal{I})$ where $\mathcal{E}$ is the vertex set of a graph and $\mathcal{I}$ is a collection of independent sets, satisfies the condition of Theorem 4.1.3 and therefore we can list all of them in output-polynomial. Paull and Unger presented an output-polynomial algorithm for listing all maximal independent sets of graphs [88]. A stronger notion in output-sensitive algorithms is the notion of *incremental polynomial delay*.

Definition 22 (Incremental polynomial)**.** *[73] Given an input and a part of a collection of objects, the running time of an algorithm that finds another object or determines that no other objects exist in time polynomial in terms of the size of the input and the total size of the given objects is called* incremental polynomial.

Definition 23 (Incremental polynomial-delay)**.** *The running time of an algorithm that applies an incremental polynomial algorithm as subroutine to generate all the elements of a finite family of objects is called* incremental polynomial-delay *and the running time of the subroutine, the incremental polynomial algorithm, is known as the* delay.

Assuming that the length of each object is polynomial in the size of input, it is not hard to see that incremental polynomial-delay implies output-polynomial. As an example, Golovach, Heggernes, Kratsch and Villanger [63] gave an incremental polynomial-delay algorithm to generate all the minimal edge dominating sets of graphs.

Theorem 4.1.4. *[63] All minimal edge dominating sets of a graph can be generated in incremental polynomial-delay time. On input graphs on m edges, the delay is $O(m^6|\mathcal{L}|)$, and the total running time is $O(m^4|\mathcal{L}^*|^2)$, where $\mathcal{L}$ is the set of already generated minimal edge dominating sets and $\mathcal{L}^*$ is the set of all minimal edge dominating sets.*

Since every minimal edge dominating set of a graph is a minimal dominating set of its line graph, they generate all the minimal dominating sets of the line graphs of the input graphs. To generate all the minimal dominating sets of the line graphs, they use the list of all maximal independent sets. We mentioned earlier that a list of objects can be used to obtain faster algorithms for problems which are not related to the enumerated objects. A witness to this claim is the graph coloring algorithm by Lawler.

In a coloring of a graph, Lawler noted that one of the colors can be given to a maximal independent set and therefore, using the list of all maximal independent sets, he came up with $O(2.4423^n)$ algorithm to calculate the chromatic number of graphs [78] in 1976. This algorithm was the fastest known algorithm for this purpose for more than two decades. Eppstein [40] gave an $O(2.4150^n)$ time algorithm to find the chromatic number in 2001 and one year later, Byskov improved it by presenting an $O(2.4023^n)$ time algorithm [18].

Even more stronger notion than incremental polynomial-delay in output-sensitive algorithms is the notion of *polynomial delay*.

Definition 24 (Polynomial-delay). *[60] The running time of an enumeration algorithm is said to be* polynomial-delay *if and only if the following conditions are satisfied:*

1. *The algorithm takes polynomial time in the input size before either generating the first output or halting, and*

2. *after any output, it takes polynomial time in the input size to generate the next output or to halt.*

Polynomial-delay implies output-polynomial. Applying a polynomial-delay algorithm ensures that within a reasonable time a new object is generated or it is concluded that the list is complete. In general, an output-polynomial algorithm might list all the objects in the end. For example, the algorithm by Paull and Unger in [88] to list all maximal independent sets runs in $O(n^2|\mathcal{L}|)$ time, where $\mathcal{L}$ is the set of all objects, without generating any sets and then outputs all the sets in a rapid succession [73]. Moreover, a certain number of solutions is enough for several applications and therefore there is no need to wait for all of the objects to be listed. Hence, polynomial-delay algorithms are more efficient in this case. Another output-polynomial algorithm to enumerate maximal independent sets of a graph G on n vertices and m edges is due to Tsukiyama, Ide, Ariyoshi and Shirakawa [103] in 1977. Their algorithm runs in time $O(nm|\mathcal{L}|)$, where $\mathcal{L}$ is the set of maximal independent sets. In 1988, Johnson, Yanakakis and Papadimitriou [73] presented the first polynomial-delay algorithm to list all maximal independent sets of a graph. Their algorithm generates the maximal independent sets in lexicographic[1] order. They also showed that generating these objects in reverse lexicographic order cannot be done in polynomial-delay time, unless P = NP.

Theorem 4.1.5. *[73] All maximal independent sets of a graph on n vertices and m edges can be generated in lexicographic order with $O(n(m+n\log|\mathcal{L}|)) = O(n^3)$ delay, where $\mathcal{L}$ is the set of all maximal independent sets.*

4.1.2 Input-sensitive algorithms

The *input-sensitive* algorithms are the enumeration algorithms whose running time depend on the size of the input. Techniques for exact exponential-time algorithms like branching are often applied in designing the input-sensitive algorithms.

[1]In lexicographic order of subsets of an ordered set U, a subset R comes before a subset S if the first element of U at which R and S disagree is in R.

A branching algorithm for listing all maximal independent sets

The algorithm[2] has one reduction rule and one branching rule. Let G be an input graph on n vertices. It is obvious that if I is a maximal independent set of G, then I contains at least one of the vertices in $N[v]$ for every vertex v. It is also clear that if I contains a vertex, it does not contain any of its neighbors. Hence, the algorithm picks a vertex v and branches the problem into $|N[v]|$ smaller subproblems and recursively finds the maximal independent sets of the subproblems. For analytic reasons we always pick a vertex v of minimum degree. Let (G,I) be any subproblem. In the beginning G is the input graph and $I = \emptyset$. If G has no vertex then we return I.

Reduction Rule: If G contains an isolated vertex, then reduce (G,I) into $(G-v, I \cup \{v\})$.

Branching Rule: If $V(G) \notin \emptyset$, then pick a vertex v of minimum degree in G and branch it into $|d(v)+1|$ new subproblems

$$(G-N[v], I \cup \{v\})$$
$$(G-N[u_1], I \cup \{u_1\})$$
$$(G-N[u_2], I \cup \{u_2\})$$
$$\vdots$$
$$(G-N[u_{d(v)}], I \cup \{u_{d(v)}\}),$$

where $u_1, \ldots, u_{d(v)}$ are the neighbors of v.

Let I be an independent set which appears in one of the leaves of this search tree. Set I is a maximal independent set, since every vertex that does not belong to I, one of its neighbors has been added to I during the execution of the algorithm. It is also not hard to see that each of the maximal independent sets appears in at least one leaf.

Since there is only one branching rule in our algorithm, to analyze its running time we need to calculate the corresponding branching vector. The following recurrence formula is obtained immediately for the number of leaves of the branching tree

$$T(n) \leq T(n-d(v)-1) + \sum_{i=1}^{d(v)} T(n-d(u_i)-1).$$

Due to the fact that we branch on a vertex of minimum degree, $d(v) \leq d(u_i)$ holds for all $1 \leq i \leq d(v)$. Therefore we find that $n-d(u_i)-1 \leq n-d(v)-1$, for all $1 \leq i \leq d(v)$. Since T is an increasing function, for every $1 \leq i \leq d(v)$, we have $T(n-d(u_i)-1) \leq T(n-d(v)-1)$. Hence, we get

$$T(n) \leq T(n-d(v)-1) + \sum_{i=1}^{d(v)} T(n-d(u_i)-1) \leq (d(v)+1)T(n-d(v)-1).$$

[2] This is a modification of an algorithm from [53] which outputs the maximum cardinality of an independent set of the input graph.

Let us simplify the formula by putting s instead of $d(v)+1$. The branching vector of $T(n) \leq sT(n-s)$ is s-tuple $(s,s,\ldots,s)$. We need to solve $x^s - s = 0$. The root of this formula is $\alpha = s^{1/s}$ and thus $T(n) \leq \alpha^n = s^{n/s}$. Since $s^{1/s}$ gets its maximum value for $s = 3$, we find that $T(n) \leq 3^{n/3}$. Then the maximum number of leaves is $3^{n/3}$ and the number of vertices of the search tree is $O(3^{n/3})$. Taking into account the polynomial time spent for each vertex of the search tree, the total running time is $O^*(3^{n/3})$.

Upper bounding the number of specified combinatorial objects is an important area in combinatorics. We address this kind of problems in the next section. The running time of an input sensitive enumeration algorithm provides a trivial upper bound on the number of the objects that it generates. Therefore, the algorithm above upper bounds the number of maximal independent sets by $O^*(3^{n/3})$. Normally, in the branching enumeration algorithms, all the objects appear in the leaves of the corresponding search trees and therefore, the number of leaves upper bounds the number of objects. Hence, this algorithm guarantees at most $3^{n/3}$ maximal independent sets. This is an algorithmic proof of the result by Moon and Moser that proves the same upper bound by combinatorial arguments.

There are many branching enumeration algorithms that provide the best known upper bounds on the number of the objects they generate. We will present some of them in the next section together with the corresponding best known lower bounds.

4.2 Maximum number of objects in graphs

Let $\mathcal{O}$ be a graph object, and for every graph G, let F_G denote the family of all objects $\mathcal{O}$ of G. Suppose $f : \mathbb{N} \longrightarrow \mathbb{N}$ is such a function that for every $n \in \mathbb{N}$ and for every graph G on n vertices, $|F_G| \leq f(n)$. The function f is called an *upper bound function* (or simply an *upper bound*) on the number of $\mathcal{O}$. An upper bound is *tight for n* if there exists a graph G on n vertices for which $|F_G| = f(n)$. Graph G is called an *extremal graph.* An upper bound is *tight* if it is tight for some $n \in \mathbb{N}$. Ideally, we look for an upper bound which is tight for every $n \in \mathbb{N}$; however, normally, the obtained upper bounds are tight for infinitely many n. We concentrate on the objects which are some specified subsets of vertices like maximal independent set, maximal cliques, or minimal vertex covers. For the subsets of vertices as objects the function $f(n) = 2^n$ is a trivial upper bound function. Therefore, the upper bound functions for the subset of vertices are normally of the form $f(n) = \alpha^n$ for some real positive $\alpha < 2$.

For example assume that we want to find an upper bound on the number of maximal independent sets of graphs. Hence, we need to determine a function $f(n) = \alpha^n$ such that no graph on n vertices has more than α^n maximal independent sets. The result by Moon and Moser implies that $f(n) = 3^{n/3} \approx 1.4422^n$.

Let us assume again that $\mathcal{O}$ is a graph object and the best known upper bound function f on the number of $\mathcal{O}$ is not tight. It means that, for every graph G on n vertices that we have checked $|F_G|$ so far, $|F_G| < f(n)$. Let L be the set of all graphs for which the size of the corresponding families of objects have been checked. For every graph $G \in L$, we calculate $|F_G|^{1/n}$ where n is the number of vertices of G. Let β be the largest number among these calculated numbers which is obtained for a graph $H \in L$. The function

$g(n) = \beta^n$ is a function of the desired form that passes through the point $(n, |F_H|)$ on the plane. When β gets bigger by finding better families of objects, g gets closer to the upper bound f. Function g is called a *lower bound function* (or simply *lower bound*) on the maximum number of $\mathcal{O}$.

The upper bound function f can be determined by combinatorial arguments or algorithmic techniques. As we mentioned earlier for most of the studied graph objects the best known lower bounds do not match the obtained upper bounds. In the rest of this section and this chapter we present the obtained upper and lower bounds for some interesting objects.

4.2.1 Maximum number of maximal independent sets

Let us begin with an object for which there is no gap between the known upper and lower bounds. The upper bound $3^{n/3}$ on the maximum number of maximal independent sets, which we presented before, is a tight upper bound. The disjoint union of k copies of triangles has $3^k = 3^{n/3}$ maximal independent sets. This upper bound function is of the form α^n and it is tight for all $n = 3k$, $k \in \mathbb{N}$. However, there are upper bounds which are not of the form α^n, but they are tight for every $n \in \mathbb{N}$. The following theorem is the famous result by Moon and Moser.

Theorem 4.2.1. *[87] If $n \geq 2$, then*

$$f(n) = \begin{cases} 3^{n/3}, & \text{if } n = 3k; \\ 4 \cdot 3^{(n-4)/3}, & \text{if } n = 3k+1; \\ 2 \cdot 3^{(n-2)/3}, & \text{if } n = 3k+2. \end{cases}$$

We do not give the proof of Theorem 4.2.1. To show that the bound is tight, we need to introduce some graphs on n vertices which have exactly $f(n)$ maximal independent sets. When $n = 3k$, the graph is the disjoint union of $n/3$ copies of triangles. In the case of $n = 3k+1$, the graph is the disjoint union of $(n-4)/3$ copies of triangles and 2 copies of K_2. When $n = 3k+2$, the disjoint union of $(n-2)/3$ copies of triangles and a K_2 is the desired graph.

The extremal graphs with the maximum number of maximal independent sets contain many triangles. Hujter and Tuza in 1993 [72] found a tight upper bound for the maximum number of maximal independent sets in triangle-free graphs.

Theorem 4.2.2. *[72] If G is a triangle-free graph on n vertices, $n \geq 4$, then*

$$f(n) = \begin{cases} 2^{n/2}, & \text{if } n \text{ is even}; \\ 5 \cdot 2^{(n-5)/2}, & \text{if } n \text{ is odd}. \end{cases}$$

If n is an even number then the disjoint union of $n/2$ copies of K_2 is a graph that has $f(n)$ maximal independent sets. When n is an odd number then the disjoint union of $(n-5)/2$ copies of K_2 and a C_5 is an extremal graph.

4.2.2 Maximum number of maximal r-regular induced subgraphs

The result by Moon and Moser has been recently generalized by Gupta, Raman and Saurabh [69]. They studied the problem of MAXIMUM r-REGULAR INDUCED SUBGRAPH which is an NP-hard problem for any value of r [21]. They designed an exact exponential-time algorithm which gives an upper bound on the maximum number of maximal r-regular induced subgraphs.

Theorem 4.2.3. *[69] Let $G = (V, E)$ be a graph on n vertices and $r \geq 1$ be a fixed constant. Then there exists a constant $c < 2$ such that* MAXIMUM r-REGULAR INDUCED SUBGRAPH *is solved in time $O(c^n)$.*

The constant c in their theorem is

$$c = \begin{cases} 2^{(1-\frac{1}{2r})} & \text{when } r \geq 5 \text{ and} \\ 1.7635 & \text{when } 1 \leq r \leq 4. \end{cases}$$

Hence, it is concluded that the number of maximal r-regular induced subgraphs is upper bounded by $O(c^n)$ where c comes from the above formula. The upper bound is not tight for the small values of r, and therefore they improve the general upper bound when r is small.

Theorem 4.2.4. *[69] Every graph on n vertices has at most*

- $10^{n/5}(\approx 1.5848^n)$ *maximal 1-regular induced subgraphs, and*
- $35^{n/7}(\approx 1.6618^n)$ *maximal 2-regular induced subgraphs.*

Both upper bounds are tight. The disjoint union of $n/5$ copies of K_5, when n is a multiple of 5, is an extremal graph for the case of maximal 1-regular induced subgraphs. Note that every maximal 1-regular induced subgraph contains exactly one edge from each of the components. For the other bound observe that the disjoint union of $n/7$ copies of K_7, when n is a multiple of 7, is an extremal graph. Each maximal 2-regular induced subgraph contains exactly one triangle from each component and there are 35 different triangles in each component.

In paper V [4] of Part II, we concentrate on the class of triangle-free graphs and we find a tight upper bound on the number of maximal 1-regular induced subgraphs which are also known as maximal induced matchings.

4.2.3 Maximum number of other objects in graphs

There are other objects in graphs that the maximum number of them have been studied. For most of them the best known upper bound have been obtained by enumeration algorithms. Here we have collected some of the important results.

Maximum number of minimal dominating sets

Fomin, Grandoni, Pyatkin, and Stepanov in [52] upper bounded the number of minimal dominating sets.

Theorem 4.2.5. *[52] For every graph on n vertices, the number of all minimal dominating sets is at most 1.7159^n.*

The best known lower bound for the maximum number of minimal dominating sets is 1.5704^n [52]. The disjoint union of $n/6$ copies of the Octahedron graph is the graph which has $15^{n/6}$ ($\approx 1.5704^n$) minimal dominating sets.

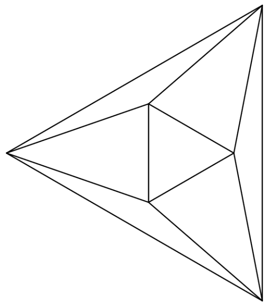


Figure 4.1: The Octahedron graph.

In an Octahedron graph, we need at least two vertices to dominate all of the vertices and every pair of vertices is indeed a minimal dominating set. Therefore, each Octahedron graph has 15 different minimal dominating sets. There is a large gap between the best known upper and lower bounds for the maximum number of minimal dominating sets. Couturier, Heggernes, van 't Hof, and Kratsch [32] focused on graph classes to close this gap. For chordal graphs, split graphs and proper interval graphs, they devise branching algorithms and get significantly better upper bounds than 1.7159^n for general graphs. Their upper bound for chordal graphs is 1.6181^n and for both split graphs and proper interval graphs is 1.4656^n. The lower bound is $3^{n/3}$ for all of these three classes. They also obtained tight upper bounds for the classes of cographs, trivially perfect graph, threshold graphs and chain graphs using combinatorial arguments. In 2013, Couturier and Liedloff [33] showed that in split graphs the upper bound is $3^{n/3}$, and it matches the lower bound given in [32]. Golovach, Heggernes, Kanté, Kratsch and Villanger in [61], which has been submitted recently, prove that the upper bound for interval graphs is also $3^{n/3}$, again matching the lower bound presented in [32]. The following table summarizes the best known results on the mentioned classes of graphs.

Graph class	Lower bound	Upper bound
General graphs	$15^{n/6}$	1.7159^n
Chordal graphs	$3^{n/3}$	1.6181^n
Split graphs	$3^{n/3}$	$3^{n/3}$
interval graphs	$3^{n/3}$	$3^{n/3}$
Cographs	$15^{n/6}$	$15^{n/6}$
Trivially perfect graphs	$3^{n/3}$	$3^{n/3}$

Table 4.1: Best known lower and upper bounds on the maximum number of minimal dominating sets [32, 33, 52, 61].

Maximum number of maximal induced bipartite subgraphs

Byskov, Madson and Skjernaa gave upper and lower bounds on the maximum number of maximal induced bipartite subgraphs.

Theorem 4.2.6. *[19] There exist graphs which have* $105^{n/10}(\approx 1.5926^n)$ *maximal induced bipartite subgraphs.*

Proof. Disjoint union of $n/10$ copies of the following graph has $105^{n/10}$ maximal induced bipartite subgraphs.

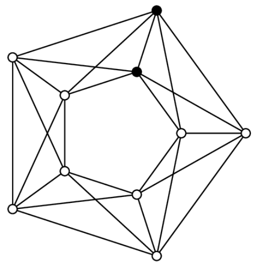
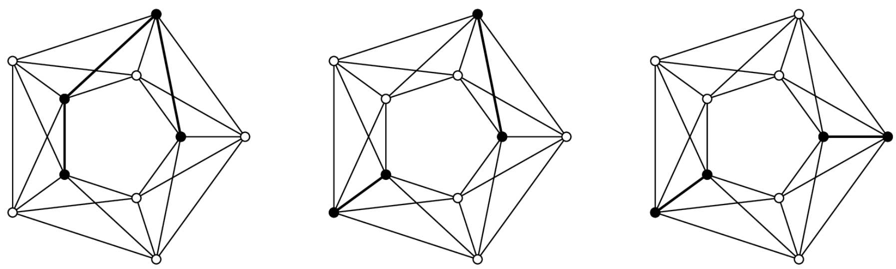


Figure 4.2: Generating graph with a pair marked.

A vertex on the outer C_5 together with the closest vertex on the inner C_5 is called a *pair*. Each component, the generating graph of the disjoint union, has 105 maximal induced bipartite graphs.

- $5 \cdot 2^4 = 80$ of them contain one vertex from four of the pairs (see Fig.4.3a),
- $5 \cdot 2^2 = 20$ of them contain one pair and one vertex from each of the opposite pairs (see Fig. 4.3b), and
- 5 of them contain two pairs (see Fig. 4.3c).



(a) (b) (c)

Figure 4.3: Three types of maximal induced bipartite subgraphs in the generating graph

□

They also presented a combinatorial proof to upper bound the number of maximal induced bipartite subgraphs.

Theorem 4.2.7. *[19] Every graph on n vertices contains at most* $O(12^{n/4}) = O(1.8613^n)$ *maximal bipartite subgraphs.*

Maximum number of minimal feedback vertex sets

A subset of vertices in a graph whose removal results in an acyclic graph is called a *feedback vertex set*. In 2008, Fomin, Gaspers, Pyatkin, and Razgon [49] found an upper bound on the number of minimal feedback vertex sets of graphs.

Theorem 4.2.8. *[49]. A graph on n vertices contains at most* 1.8638^n *minimal feedback vertex sets.*

They have a branching algorithm in order to obtain the upper bound above. They also show that the maximum number of these objects is lower bounded by 1.5926^n. The same graph that lower bounds the number of maximal induced bipartite subgraphs, lower bounds the number of minimal feedback vertex sets. Each of the 105 minimal dominating sets in the generating graph (see Fig. 4.2 in the proof of theorem 4.2.6), forms also a maximal forest in the graph and thus its complement forms a minimal feedback vertex set. In this problem the gap is considerable between the upper and lower bounds. Motivated by this, in 2012, Couturier, Heggernes, van 't Hof and Villanger [34] looked at some restricted classes of graphs. They closed this gap, i.e., found matching upper bound and lower bound, for the class of chordal graphs and cographs.

Theorem 4.2.9. *[34] Every chordal graph on n vertices has at most* $10^{n/5}$ *minimal feedback vertex sets.*

Theorem 4.2.10. *[34] Every cograph graph on n vertices has at most* $10^{n/5}$ *minimal feedback vertex sets.*

The lower bound that they presented for both chordal graphs and cographs is the disjoint union of $n/5$ copies of K_5. Each of the copies contains 10 maximal induced forests as every edge is such a forest. Therefore, the same number of minimal feedback vertex sets exist in each component.

Maximum number of minimal subset feedback vertex sets

Given a graph G and a subset S of its vertices, a *subset feedback vertex set* of (G,S) is a set X of vertices of G such that its removal destroys all cycles that contain a vertex of S. It generalizes the definition of feedback vertex set, as a subset feedback vertex set of (G,V) is a feedback vertex set of G. An *S-forest* of $G=(V,E)$, where S is a subset of V, is a vertex set $Y \in V$ such that $G[Y]$ contains no cycle with a vertex in S. An S-forest Y of $G=(V,E)$ is maximal if and only if $V \setminus Y$ is a minimal subset feedback vertex set of (G,S). The best known lower bound on the maximum number of minimal subset feedback vertex sets is the same as on the maximum number of minimal feedback vertex sets, 1.5926^n. In 2011, Fomin, Heggernes, Kratsch, Papadopoulos and Villanger [50] enumerated all subset feedback vertex sets of graphs.

Theorem 4.2.11. *The minimal feedback vertex sets of an input* (G,S)*, where G is a graph on n vertices, can be enumerated in time* $O(1.8638^n)$*. Moreover, the maximum number of maximal S-forests of a graph on n vertices is at most* 1.8638^n*.*

In Paper IV [62] of Part II, we enumerate minimal subset feedback vertex sets of chordal graphs. We prove that a chordal graph on n vertices can have at most 1.6708^n

minimal subset feedback vertex sets, regardless of S. This narrows the gap with respect to the best known lower bound $10^{n/5} (\approx 1.5848^n)$, see Theorem 4.2.9, for chordal graphs.

Chapter 5

Extremal graph theory and Ramsey numbers

Extremal Graph Theory concerns problems that attempt to find a relation between a graph property and graph invariants. In this chapter we provide a brief introduction to extremal graph theory, focusing on a special kind of extremal problems known as the Ramsey-type problems on graphs. This chapter comprises four sections. In the first section we introduce extremal graph theory problems, we discuss the history and the first problem from which all extremal graph theory problems are believed to spring. In this section we also introduce the Ramsey numbers of graphs as an extremal graph theory problem. Ramsey-type problems are not limited only to graph theory and there are many other mathematical structures on which similar problems are defined. Therefore, we will study the Ramsey theory separately in the second section. There, we present the history of Ramsey theory and give some examples of Ramsey-type problems. The third section concerns the Ramsey theory on graphs. Definition and results on Ramsey numbers of graph classes are presented in the last section.

5.1 Extremal graph theory

Generally, in all extremal graph problems we are given a graph property $\mathcal{P}$ and we are asked to determine the least value t of a graph invariant (such as number of vertices, number of edges, minimum degree or chromatic number) to ensure that the graph has property $\mathcal{P}$ [64]. This implies that there exists a graph G with the invariant of value $t-1$ such that property $\mathcal{P}$ does not hold for it. Graph G is then called an *extremal graph*, and $t-1$ is known as the *extremal value*. The following is a simple example of these problems: what is the least number of edges of a graph on n vertices so that it has a cycle? The extremal graphs here are trees and the extremal value is $n-1$. Although this is an old result in graph theory, it is not known as the first problem in extremal graph theory. It is believed that the extremal graph theory started with Turán's theorem in 1941 [12].

Definition 25 (Turán Number)**.** *Let H be an arbitrary graph and let m be the least integer such that every graph on n vertices and m edges contains H as its subgraph. The extremal value $m-1$ is called the* Turán number *of H and is denoted by $T(n,H)$.*

The problem of determining the Turán numbers is known as *Turán's problem*. Turán solved in 1941 a special case of this problem which is one of the famous results in graph theory. In the following theorem which is known as Turán's theorem, Turán determined the Turán numbers when the graph H is a complete graph of arbitrary size. There are different proofs of Turán's theorem; we present one of those proofs here for the sake of completeness.

Theorem 5.1.1 (Turán's theorem, 1941)**.** [104]

$$T(n,K_r) = (1 - \frac{1}{r-1})\frac{n^2}{2}.$$

Proof. We prove the theorem by induction on the number of vertices. The result is trivially true for $n = 1,2,3,\ldots,r-1$. Therefore, we assume that $n \geq r$ and the statement holds for all graphs on less than n vertices. Among all graphs on n vertices which have no clique of size r, let G be the one that has maximum number of edges. Graph G must contain a clique of size $r-1$; since otherwise by adding some edges to G we can create another graph which is on n vertices, contains no clique of size r, and has more edges than G.

Let A be a clique of size $r-1$ and let B denote $G-A$. Graph B is on $n-r+1$ vertices and contains no clique of size r. Therefore, it contains at most $(1-\frac{1}{r-1})\frac{(n-r+1)^2}{2}$ edges. The number of edges between A and B is at most $(n-r+1)(r-2)$ as no vertex in B is adjacent to all vertices of A. This implies that

$$|E(G)| \leq \frac{(r-1)r}{2} + (n-r+1)(r-2) + (1-\frac{1}{r-1})\frac{(n-r+1)^2}{2} = (1-\frac{1}{r-1})\frac{n^2}{2},$$

and this completes the proof. □

A special case of Theorem 5.1.1 when H is K_3, was proved long before by Mantel in 1907.

Theorem 5.1.2 (Mantel's theorem, 1907)**.** [83]

$$T(n,K_3) = \lfloor\frac{n}{2}\rfloor \cdot \lceil\frac{n}{2}\rceil.$$

Definition 26 (Turán graph)**.** *Let n and r be two positive integers. The unique complete r-partite graph $T(n,r)$ on n vertices is called the* Turán graph *if its vertex set is partitioned into r subsets such that they differ in size by at most* 1.

To be more precise, if $n = ar+b$ for positive integers a and b then the graph has b sets of size $a+1$ and each of the remaining $r-b$ sets are of size a. Since every set of vertices of size $r+1$ needs to have at least two vertices in one of the r sets, Turán's graph has no complete graph of size $r+1$. The number of edges of this graph is $(1-\frac{1}{r})\frac{n^2}{2}$. Hence, Turán graph $T(n,r-1)$ is an extremal graph for Turán's problem when $H = K_r$. Figure 5.1 illustrates $T(13,4)$.

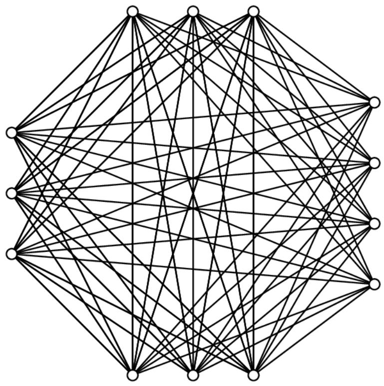


Figure 5.1: Turán graph $T(13,4)$.

In both Mantel's theorem and Turán's theorem the given graph H is a complete graph. Determining the Turán number of other graphs such as complete bipartite graphs or even cycles have been studied extensively and there are many open problems and conjectures in the field. We refer the interested reader to the book by Chung and Graham [24]. There are other problems in extremal graph theory that have branched from the definition of Turán number. The following is an instance of such problems. The fact that Turán graph has a large independent set of size $\lceil n/r \rceil$ motivated investigation on graphs with smaller independent sets.

Definition 27 (Ramsey-Turán number)**.** *Let H be a graph and let m be a least integer m such that every graph on n vertices and m edges that has no independent set of size j, has H as its subgraph.* Ramsey-Turán number *is defined to be the extremal value $m-1$ and is denoted by $RT(n,H,j)$.*

However, there are many problems in extremal graph theory that are intending different graph invariants. Determining Ramsey numbers of graphs, which is one of the most extensively studied problems in extremal graph theory, deals with the number of vertices instead of the number of edges.

Definition 28 (Ramsey number)**.** *Let $G_1, G_2, \ldots, G_t$ be graphs. The least integer n is called the* Ramsey number *of $G_1, G_2, \ldots, G_t$ if for every coloring of edges of K_n by t colors there is a monochromatic subgraph isomorphic to G_i for some $i \in \{1,2,\ldots,t\}$. We denote the Ramsey number of $G_1, G_2, \ldots, G_t$ by $R(G_1, G_2, \ldots, G_t)$.*

Note that unlike the definitions of Turán numbers and Ramsey-Turán numbers, where the extremal values are defined to be the numbers, here the least values are the Ramsey numbers.

The problem of determining the Ramsey numbers of graphs belongs to a large field of mathematics, namely Ramsey theory, which is approximately 100 years old [98]. Ramsey theory consists of problems of the same type from various fields such as number theory, algebra, and graph theory. The phrase *Ramsey-type problem* is to describe all these problems. Before we study the Ramsey numbers of graphs, we first introduce Ramsey theory, we present its history and early Ramsey-type problems, and then we will describe the Ramsey-type problems on graphs.

5.2 Ramsey theory

In 1928 Frank P. Ramsey who was a British economist, philosopher and mathematician [108], worked on the problem of finding a regular procedure to determine whether a given logical formula is valid. In the paper which was published in 1930, he proved certain combinatorial theorems that are interesting on their own. One of these theorems that became known as the Ramsey's theorem, led to a large area of mathematics known as Ramsey theory. The theory deals with a wide range of mathematical structures such as integer numbers, vector spaces, and graphs. There is no unique definition of Ramsey theory which is accepted universally. However, the main mathematical idea of the theory is the following: "no matter how large and elaborate a system S is, and how large a positive integer n is, we can choose a large enough super system Q containing S, so that no matter how Q is colored in n colors, Q contains a monochromatic copy of S" [98]. Ramsey theory is the mathematics of coloring.

This kind of problems were worked on much earlier than Ramsey's theorem. Here we mention three of the important earlier works on problems of Ramsey-type prior to Ramsey's theorem, namely the Hilbert cube lemma, Schur's theorem and Vander Waerden's theorem, which are all from number theory. The following history of Ramsey-type problems has been adapted from the book [98], so an interested reader is referred to this book for an extended survey of the history. The explanations of the problems within the history have been enriched from the book [77] by Landman and Robertson.

Hilbert's Cube Lemma

The first work is due to David Hilbert which is the first Ramsey-type problem to the best of our knowledge. A set $Q_n(a,x_1,x_2,\ldots,x_n)$ of integers is called an *n-dimensional affine cube* if there exist $n+1$ positive integers $a,x_1,x_2,\ldots,x_n$ such that

$$Q_n(a,x_1,x_2,\ldots,x_n) = \{a+\sum_{i\in F} x_i : \emptyset \neq F \subseteq \{1,2,\ldots,n\}\}.$$

Lemma 5.2.1 (The Hilbert cube lemma, 1892)**.** *For every pair of positive integers r and n, there exists a least positive integer $m = H(r,n)$ such that in every r-coloring of $\{1,2,3,\ldots,m\}$ there exists a monochromatic n-dimensional affine cube.*

Nobody at that time appreciated this lemma and even Hilbert himself did not continue to work in that direction.

Schur's Theorem

The second Ramsey-type problem appeared in 1916. Issai Schur offered another proof to the following theorem that was proved earlier by Leonardo E. Dickson [36] in 1908.

Theorem 5.2.1. *Let $n \geq 1$. There exists a prime q such that for all primes $p \geq q$ the congruence $x^n + y^n \equiv z^n \pmod{p}$ has a solution in the integers with $xyz \equiv 0 \pmod{p}$.*

Dickson was trying to prove the Fermat's last theorem. Fermat's last theorem was conjectured by Pierre de Fermat in 1637 and was open for 358 years. The following is the translation of the original statement by Fermat:

"it is impossible to separate a cube into two cubes, or a fourth power into two fourth powers, or in general, any power higher than the second, into two like powers. I have discovered a truly marvelous proof of this, which this margin is too narrow to contain". [82]

Theorem 5.2.2 (Fermat's last theorem). *No three positive integers x, y, and z can satisfy the equation $x^n + y^n = z^n$ for any $n > 2$.*

Schur proved the following theorem in order to use it as a lemma in his proof to Theorem 5.2.1. His lemma is known as Schur's theorem.

Theorem 5.2.3 (Schur's Theorem, 1916). [95] *For any $r \geq 1$, there exists a least positive integer $s = s(r)$ such that for any r-coloring of $\{1,2,3,\ldots,s\}$ there is a monochromatic solution to $x + y = z$.*

Numbers $s(r)$ are called the *Schur's numbers*. We know that $z = x + y$ is the equation of a plane in three-dimensional space that every point (x,y,z) on it satisfies the condition $x + y = z$. Let P be the set of all points on this plane whose coordinates are positive integers. Consider an arbitrary coloring of positive integers with a finite number of colors. For each element (a,b,c) of P, if all a, b and c get the same color then in the plane color the corresponding point with that color and otherwise put the label X on the point. The question is that whether every coloring leads to a colored point, i.e., whether there is a solution for $x + y = z$ in one of the partition sets. The answer is "yes, there is always a colored point," by Schur's theorem. The theorem, guarantees the existence of the least number and Schur actually proved by induction that $s(r) = r!e$ works. As we said before, Schur was not motivated by coloring the points of plane and it seems that he was perhaps motivated by last theorem of Fermat. Therefore he did not continued working on this direction and nobody else did so.

Van der Waerden's Theorem

Van der Waerden theorem is another Ramsey-type theorem which appeared three years before the publication of Ramsey's theorem. Let us first define Van der Waerden numbers by an example.

Example 5.2.1. [77] *Consider the arithmetic progressions of length three. The least positive integer $w = w(3;2)$ such that any two-coloring of integers $1,2,\ldots,w$ yields a monochromatic arithmetic progressions of length 3 is called a* Van der Waerden number of 3 and 2.

Let us show that $w(3;2) \geq 9$. To do so, it suffices to give a two-coloring of $\{1,2,\ldots,8\}$ such that there exists no monochromatic arithmetic progression of length 3. Color numbers $1,4,5$ and 8 by red and color the rest by blue. It is easy to see that there is no arithmetic progression in any of the sets $\{1,4,5,8\}$ and $\{2,3,6,7\}$. So the smallest such number is not 8 and thus $w(3;2) \geq 9$. But, does $w(3;2)$ actually exist? It has been shown that the number exists and $w(3;2) = 9$. What if we use more than two colors or desire a longer monochromatic arithmetic progression?

Definition 29 (Van der Waerden number)**.** *Let k and r be two positive integers. Consider the arithmetic progressions of length k. Smallest positive integer $w = w(k;r)$ such that any r-coloring of integers $1,2,\ldots,w$ yields a monochromatic arithmetic progressions of length k is called a* Van der Waerden number of k and r.

The following theorem states that actually for any number of colors and any length of arithmetic progression such a number exists.

Theorem 5.2.4 (Van der Waerden, 1927)**.** *Let $k,r \geq 2$ be two integers. There exists a least positive integer $w = w(k;r)$ such that for all $n \geq w$, for every r-coloring of $\{1,2,3,\ldots,n\}$ there is a monochromatic arithmetic progression of length k.*

The desired elements to have the same color in Van der Waerden numbers, do not necessarily need to form an arithmetic progression. Let us show the set of all arithmetic progressions by AP and change the notation of the Van der Waerden numbers to $R(AP,k;r)$. To generalize the definition we assume that F is some specific collection of sequences. Therefore, in the case of Van der Waerden's numbers we have $F = AP$ and Theorem 5.2.4 guarantees the existence of the least number. Note that there is no guarantee that for every F the value $R(F,k;r)$ exists and we need to choose F carefully. For example, define F as a collection of sequences that all start with $i, i+1$ for some positive integer i. It is easy to see that $R(F,2;2)$ does not exist. For any n, consider the 2-coloring of $\{1,2,\ldots,n\}$ that gives red to odd numbers and blue to even numbers. It implies that there is no monochromatic sequence that belongs to F. This means that $R(F,2;2)$ does not exist.

Consider an arithmetic progression of length three like x,y,z. There exists a positive integer d such that $x = a$, $y = a+d$ and $z = a+2d$. Therefore, we have got $x+y = 2z$. Van der Waerden's theorem proved that the smallest number exists for the case that F is such a sequence. What if we change F to a sequence of three positive integers x,y,z that $x+y = z$. Has this F been chosen wisely? Schur's theorem has proved the existence of the least number for such an F. Therefore, we can consider both the Schur numbers and the Van der Waerden numbers as two special cases of wisely chosen collections of sequences.

Ramsey-type problems other than graph theory and number theory have been studied on algebraic structures like vector spaces, see [65, 99]. We refer the reader to the book by Graham, Rothschild and Spencer [66] for further study on the theory. After this brief introduction to the Ramsey theory, it is time to concentrate on the Ramsey-type problems on graphs that is our aim of having this chapter.

5.3 Ramsey theory on graphs

Let $G_1, G_2, \ldots, G_t$ be t graphs. The smallest integer $n = R(G_1, G_2, \ldots, G_t)$ is called the *Ramsey number* of $G_1, G_2, \ldots, G_t$ if for every edge coloring of K_n by t colors there is a monochromatic subgraph isomorphic to G_i for some $i \in \{1,2,\ldots,t\}$. A Ramsey number is called a 2*-color Ramsey number* when there are only two graphs G_1 and G_2. The *classical Ramsey number* is a 2-color Ramsey number where both graphs G_1 and G_2 are cliques. We denote the classical Ramsey number of two clique K_i and K_j by $R(i,j)$. In this thesis we concentrate only on 2-color Ramsey numbers.

Definition 30 (Classical Ramsey number)**.** *The Ramsey number $R(i, j)$ is the smallest $n \in \mathbb{N}$ such that in any edge coloring of K_n by two colors, blue and red, there exists a blue copy of K_i or a red copy of K_j.*

It is simply written $R(s)$, in the special case that $i = j = s$. Note that every graph G on n vertices is obtained from K_n by removing some edges. For an arbitrary graph, let us color the removed edges of K_n by red and the rest of the edges by blue. Hence, it is clear that every graph corresponds to a 2-coloring of the edges of K_n. Let us always color the removed edges by red. In the definition of the classical Ramsey numbers, when we say that any blue and red coloring of the edges of K_n contains a blue copy of K_i or a red copy of K_j, we could instead say that any graph on n vertices contains a clique of size i or an independent set of size j. So the following definition is equivalent to the definition of the classical Ramsey numbers.

Definition 31. *The classical Ramsey number $R(i, j)$ is the smallest $n \in \mathbb{N}$ such that any graph on n vertices contains a clique of size i or an independent set of size j.*

To show that $R(i, j) = n$ for some $n \in \mathbb{N}$, we need to prove that n serves as an upper bound and as a lower bound for $R(i, j)$.

- To establish that $R(i, j) \leq n$, it suffices to show that every graph on n vertices has a clique of size i or an independent set of size j; or equivalently, we need to show that every 2-coloring of the edges of K_n has a blue copy of K_i or a red copy of K_j.

- To establish that $R(i, j) \geq n$, we need to provide an example of a graph on $n-1$ vertices such that it has neither a clique of size i nor an independent set of size j; or equivalently, it suffices to find a 2-coloring of the edges of K_n such that it has neither a blue copy of K_i nor a red copy of K_j.

Some results on the classical Ramsey numbers

The proof of the following theorem is easy and straightforward.

Theorem 5.3.1. *For any $i, j \geq 1$ we have*

- $R(i, j) = R(j, i)$*,*
- $R(1, j) = 1$*, and*
- $R(2, j) = j$*.*

Hence we assume that $i \geq 3$ and $j \geq 3$.

As we mentioned in generalizing the definition of Van der Waerden numbers, there is always the risk of defining a Ramsey-type number that might not exist. Is the definition of the classical Ramsey numbers wise? Do all classical Ramsey numbers exist? Ramsey's theorem guarantees the existence of these numbers.

Lemma 5.3.1. [31]

$$R(i+1, j+1) \leq R(i, j+1) + R(i+1, j).$$

Proof. Consider a 2-coloring of the edges of K_n such that there is neither a blue clique of size $i+1$ nor a red clique of size $j+1$. Let v be an arbitrary vertex of K_n. There are at most $R(i,j+1)-1$ blue edges incident with v, since otherwise the graph induced by the neighbors of v via blue edges would contain either a blue clique of size i or a red clique of size $j+1$. We are done if it contains a red K_{j+1}, but if it has a blue K_i then v together with this clique form a blue clique of size $i+1$. Similarly, the red neighborhood of v is of size at most $R(i+1,j)-1$. Hence,

$$n \leq 1+(R(i,j+1)-1)+(R(i+1,j)-1) = R(i,j+1)+R(i+1,j)-1.$$

Hence, when $n = R(i,j+1)+R(i+1,j)$ every 2-coloring of the edges of K_n has a blue clique of size $i+1$ or a red clique of size $j+1$. This would imply that $R(i+1,j+1) \leq R(i,j+1)+R(i+1,j)$. $\square$

The following theorem provides an upper bound for all the classical Ramsey numbers and therefore it implies that the smallest numbers always exist.

Theorem 5.3.2 (Ramsey's theorem)**.** [31] *For any* $i,j \geq 1$*,* $R(i,j) \leq \binom{i+j-2}{j-1}$.

Proof. We prove the statement by induction. From Theorem 5.3.1, we have that $R(i,1) = 1 = \binom{i-1}{0}$ and $R(1,j) = 1 = \binom{j-1}{j-1}$. Assume $R(s,t) \leq \binom{s+t-2}{t-1}$ for all (s,t) where $s < i$ or $t < j$. Using Lemma 5.3.1 and the induction hypothesis, we can conclude the following:

$$\begin{aligned} R(i,j) &\leq R(i-1,j)+R(i,j-1) \\ &\leq \binom{i+j-3}{j-1}+\binom{i+j-3}{j-2} \\ &= \binom{i+j-2}{j-1}. \end{aligned}$$

$\square$

Let us now try to determine the Ramsey number $R(3,3)$. Theorem 5.3.2 implies that $R(3,3) \leq \binom{4}{2}$ and hence we have that $R(3,3) \leq 6$. So we are done if we can find a graph on 5 vertices that has no clique of size 3 and no independent set of size 3. A cycle on 5 vertices is such a graph and therefore we conclude that $R(3,3) = 6$.

Theorem 5.3.3. $R(3,3) = 6$.

In the following theorem we determine the Ramsey number $R(3,4)$. Unlike the Ramsey number $R(3,3)$ the provided upper bound by the Ramsey's theorem, $R(3,4) \leq 10$, is not the smallest.

Theorem 5.3.4. *[67]* $R(3,4) = 9$.

Proof. We want to show that every 2-coloring of the edges of K_9 contains either a blue K_3 or a red K_4. Let us assume for contradiction that there is a 2-coloring of K_9, which has neither a blue K_3 nor a red K_4. We first show that for every vertex in K_9, exactly five of the incident edges are colored red. Let v be an arbitrary vertex in K_9. Let us partition the neighborhood of v into two sets $N_b(v)$ and $N_r(v)$ where $N_b(v)$ is the set of neighbors of v through blue edges and $N_r(v)$ denotes the set of neighbors of v through the red edges. If $|N_r(v)| \geq 6$, then the graph induced by $N_r(v)$ forms a complete graph on at least 6 vertices. Since $R(3,3) = 6$ then there must exist a monochromatic clique of size 3. This contradicts the assumption that there is neither a blue K_3 nor a red K_4. So the number of red edges incident with v is at most 5. Assume that $|N_r(v)| \leq 4$, i.e., $|N_b(v)| \geq 4$. If there exists a blue edge in the clique induced by $N_b(v)$, then there is a blue clique of size 3 in the clique induced by $\{v\} \cup N_r(v)$. If there is no blue edge in the clique induced by $N_b(v)$, then there is a red clique of size 4 in that clique. Hence in both cases we get contradiction. Therefore, we have proved that exactly five edges incident with v have been colored red.

We show that this cannot hold for all vertices of K_9. Assume for contradiction that it holds for all vertices of K_9. So, there are nine vertices of degree 5 in the graph induced by the red edges. A graph that has an odd number of vertices of odd degrees does not exist and thus we obtain a contradiction. This implies that such coloring of K_9 cannot exist.

It remains to show that there exists a 2-coloring, blue and red, of the edges of K_8 such that there is neither a blue copy of K_3 nor a red copy of K_4. Figure 5.2 shows a desired 2-coloring of K_8. This completes the proof. □

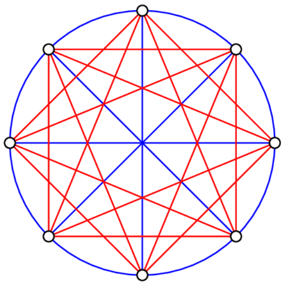


Figure 5.2: Two coloring of K_8 with no blue K_3 and no red K_4.

Having the value of $R(3,4)$ and Lemma 5.3.1 we conclude that $R(4,4) \leq 18$. There exists a graph on 17 vertices that has a 2-coloring of its edges with no monochromatic copy of K_4. To create such an edge coloring, color the following edges by blue and the rest by red: a cycle that contains all the vertices and all the chords of length 1, 2, 4 and 8 [67].

The other Ramsey numbers that have been determined are the followings: $R(3,5)$, $R(3,6)$, $R(3,7)$, $R(3,8)$, $R(3,9)$, and $R(4,5)$. In 1994 Radziszowski collected all results on Ramsey numbers and so far he has updated the survey 14 times [91]. Efforts to determine more Ramsey numbers continue. In the case that the exact values are unknown, the gap between the known upper and lower bounds are very large normally. Improving the existent bounds form a large part of the research in this field. To improve a lower bound, one should examine all 2-colorings of the edge set of a complete graph K_n, normally on a large number of vertices, to find a coloring that has no monochromatic cliques of some given sizes. Therefore, they need to examine an exponentially many, in terms of n, different 2-colorings of K_n which is not easy without making use of super computers. In the last three decades computers and supercomputers, better algorithms and nicer combinatorial arguments have assisted the researchers to obtain

$i \backslash j$	3	4	5	6	7	8	9	10
3	6	9	14	18	23	28	36	40–42
4		18	25	36–41	49–61	58–84	73–115	92–149
5			43–49	58–87	80–143	101–216	126–316	144–442
6				102–165	113–298	132–495	169–780	179–1171
7					205–540	217–1031	241–1713	289–2826
8						282–1870	317–3583	330–6090
9							565–6588	581–12677
10								798–23556

Table 5.1: Trivially, it holds that $R(1,j)=1$ and $R(2,j)=j$ for all $j \geq 1$, and $R(i,j)=R(j,i)$ for all $i,j \geq 1$. The above table contains the currently best known upper and lower bounds on $R(i,j)$ for all $i,j \in \{3,\dots,10\}$ [43, 91].

improvements in the lower bounds. The following table shows the most updated known values and bounds on the classical Ramsey numbers so far. For bigger table see [91]. The survey also contains more tables on different variations of Ramsey numbers.

In Theorem 5.3.2 we saw an upper bound on all Ramsey numbers $R(i,j)$. Taking a look at the table and the known exact values, we notice that most of them are the Ramsey numbers with $i=3$. So, in the case that $i=3$, we are more optimistic to be able to determine the exact values or to obtain better bounds. Therefore, some of the researchers have focused on the Ramsey numbers $R(3,j)$. To determine such numbers, we need to restrict ourselves to triangle-free graphs and search for the independent sets of size j. However, discovering the exact values of $R(3,j)$ when $j \geq 10$ is still far beyond the current theoretical knowledge and computational methods. The following gives an upper bound for all Ramsey numbers of the form $R(3,j)$.

Theorem 5.3.5. [98] $R(3,j) \leq j^2$.

Proof. To prove that $R(3,j) \leq n$, we need to show that every triangle-free graph G on n vertices has an independent set of size j. Let G be a triangle-free graph on j^2 vertices. Assume that there exists a vertex v in G of degree at least j. Since the graph is triangle-free then $N(v)$ is an independent set of size j and we are done. So we assume that every vertex in G has degree at most $j-1$.

The following algorithm outputs an independent set of size j.

Algorithm IS
1. $I \leftarrow \emptyset$
2. **while** G is nonempty
3. Select any v
4. Add v to I
5. Delete v and its neighbors from G
6. **return** I

For each vertex that is added to I, at most j vertices are deleted from the graph. Since the algorithm starts with j^2 vertices, the returned set I contains at least j vertices. Clearly, I is an independent set and this completes the proof. □

Other 2-color Ramsey numbers

The history of the Ramsey number indicates that the denser the graphs G_1 and G_2 are the harder the Ramsey numbers are to determine or estimate. Therefore, when the graphs to be avoided are sparse graphs we know more than the cases that they are very dense graphs like cliques. The following theorem is an evidence to support this claim.

Theorem 5.3.6. [25] $R(T_n, K_m) = (n-1)(m-1)+1$ *for any tree* T_n *with* n *vertices.*

Before we give the proof of theorem 5.3.6 we need the following lemma.

Lemma 5.3.2. [25] *If* H *is a graph of minimum degree at least* $n-1$ *then* H *contains every tree on* n *vertices.*

Proof. The proof is by induction on n. The statement holds clearly when $n = 2$. Let T be a tree on n vertices and v be a leaf of T. By the induction hypothesis, H contains a copy of $T - v$. Let S denote the set of all vertices of $T - v$. Since the minimum degree of H is at least $n-1$, every vertex $u \in S$ is adjacent to at least a vertex which does not belong to S. Therefore, H contains a copy of T. □

Now we are ready to give the proof of Theorem 5.3.6.

Proof. The disjoint union of $(m-1)$ copies of K_{n-1} is a graph that has neither a copy of T_n nor an independent set of size m. Therefore, $(m-1)(n-1)+1$ establishes a lower bound for $R(T_n, K_m)$. Now we show that the same number is an upper bound. To do so, assume that G is a graph on $(m-1)(n-1)+1$ vertices and it has no independent set of size m. We show that it contains a copy of T_n. Since there is no independent set of size m in G, at most $m-1$ vertices can get the same color in any vertex coloring of G. Hence, the chromatic number of G is at least n. From the fact that every vertex of a critical k-chromatic graph is of degree at least $k-1$ [38], G has a subgraph of minimum degree at least $n-1$. Due to Lemma 5.3.2 there is a copy of T in G and this completes the proof. □

Now assume that both graphs G_1 and G_2 in the definition of 2-color Ramsey numbers are cycles. The following theorem presents the exact values of all such Ramsey numbers.

Theorem 5.3.7. [45, 90, 93]

$$R(C_n, C_m) = \begin{cases} 2n-1 & \textit{for } 3 \le m \le n, \textit{ m odd}, \\ & \textit{and } (n,m) \ne (3,3); \\ n-1+m/2 & \textit{for } 4 \le m \le n, \textit{ m and n even}, \\ & \textit{and } (n,m) \ne (4,4); \\ \max\{n-1+m/2, 2m-1\} & \textit{for } 4 \le m \le n, \textit{ m even and n odd.} \end{cases}$$

Consider the 2-color Ramsey numbers for which both graphs G_1 and G_2 are disjoint union of short cycles. In 1975, Burr, Erdős and Spencer [16] have studied this type of Ramsey numbers. They proved that $R(nC_3, mC_3) = 3n + 2m$ for $n \ge m \ge 1$ and $n \ge 2$. Li and Wang determined the numbers for cycles on four vertices; $R(nC_4, mC_4) = 2n + 4m - 1$ when $m \ge n \ge 1$ and $(n,m) \ne (1,1)$ [80].

Assume now that G_1 is a path and G_2 is a cycle. The following theorem presents the exact values of all such Ramsey numbers.

Theorem 5.3.8. [44]

$$R(P_n, C_m) = \begin{cases} 2n-1 & \textit{for } 3 \le m \le n,\ m \textit{ odd}, \\ n-1+m/2 & \textit{for } 4 \le m \le n,\ m \textit{ even}, \\ \max\{n-1+\lfloor n/2 \rfloor, 2n-1\} & \textit{for } 2 \le n \le m,\ m \textit{ odd}, \\ m-1+\lfloor n/2 \rfloor & \textit{for } 2 \le n \le m,\ m \textit{ even}. \end{cases}$$

Due to Theorem 5.3.6 all Ramsey numbers of paths and cliques have been determined. What about the Ramsey numbers of cycles and cliques? Since a cycle is made by adding just one edge to a path on the same number of vertices, it seems not to be hard to determine the Ramsey numbers of cycles and cliques. However, surprisingly, there are not much known about these numbers. When the cycle is C_4 there are few numbers discovered. The theorem below presents some of the known numbers when the cycle is on four vertices. For the results on bigger cycles consult [90].

Theorem 5.3.9. [27, 28, 94]

- $R(C_4, K_3) = 7$,
- $R(C_4, K_4) = 10$,
- $R(C_4, K_5) = 14$,
- $R(C_4, K_6) = 18$.

Bielak [9] proved that Ramsey number $R(C_4, K_7)$ is either 21 or 22 and later Radziszowski and Tse in [92] showed that the exact value is equal to 22. Bielak also managed to lower bound the Ramsey numbers $R(C_4, K_t)$ when the clique is of size at least 6.

Theorem 5.3.10. *[9] For each integer $t \ge 6$, $R(C_4, K_{t+1}) \ge 3t + 2[\frac{t}{5}] + 1$.*

There are many more studied 2-color Ramsey numbers that we would not put them here. One can find the results on the Ramsey numbers of cycles and wheels for instance in [102], cycles and stars in [29], cycles and trees in [15, 46], cycles and complete bipartite graphs in [13], stars and stripes in [30]. For more types of 2-color Ramsey numbers take a look at [90].

5.4 Ramsey theory on graph classes

Despite the vast amount of investigations on the classical Ramsey numbers only the exact values of very few of them have been determined. Confronting such difficulty, it is natural to try to simplify the problem in some ways. To determine the classical Ramsey numbers, in the set of all simple graphs, we are looking for a graph with the maximum number of vertices such that it contains neither a clique of size i nor an independent set of size j. The class of all simple graphs seems to be too large such that vertices and edges of the graphs can be distributed very randomly and the interconnections between the elements do not adhere to any regulation. This suggests an approach to simplify the problem. What if the graphs that we consider have some special properties so that the distribution of vertices and edges are restricted according to some rules? Does it make the problem easier so that we are able to determine the exact values of all classical Ramsey numbers? Definitely applying the restrictions can simply be extended to any 2-color Ramsey numbers.

Definition 32 (2-color Ramsey number of graph classes)**.** *Let $\mathcal{G}$ be a class of graphs, and let G_1 and G_2 be two graphs not necessarily in $\mathcal{G}$. The* Ramsey number of G_1 and G_2 of class $\mathcal{G}$ *which is denoted by $R_{\mathcal{G}}(G_1, G_2)$ is the smallest $n \in \mathbb{N}$ such that any graph in $\mathcal{G}$ on n vertices contains a copy of G_1 or its complement contains a copy of G_2.*

We shall define it via 2-coloring of edges. It is the same as the 2-color Ramsey numbers of general graph except that in the 2-colorings of the edges of K_n, the blue graphs that are expected to contain G_1 should belong to $\mathcal{G}$.

Provided that graphs G_1 and G_2 are respectively cliques of sizes i and j, we call it classical Ramsey number of $\mathcal{G}$.

Definition 33 (Classical Ramsey number of graph classes)**.** *Let $\mathcal{G}$ be a class of graphs. The* classical Ramsey number of $\mathcal{G}$ *is denoted by $R_{\mathcal{G}}(i, j)$ and it is the smallest $n \in \mathbb{N}$ such that any graph in $\mathcal{G}$ on n vertices contains a copy of K_i or its complement contains a copy of K_j.*

It should be noted that the Ramsey numbers of graph classes are not always symmetric like the Ramsey numbers of general graphs (recall Theorem 5.3.1); unless the class is closed under complement. So, for graph class $\mathcal{G}$ which is not closed under complement, the equality $R_{\mathcal{G}}(i, j) = R_{\mathcal{G}}(j, i)$ does not hold necessarily.

Another point which is especially worthy of notice is that $R_{\mathcal{G}}(G_1, G_2) \leq R(G_1, G_2)$. To see this, note that all graphs on $R(G_1, G_2)$ vertices, including the graphs in $\mathcal{G}$, have a copy of G_1 or their complement have a copy of G_2. Therefore $R(G_1, G_2)$ forms an upper bound for $R_{\mathcal{G}}(G_1, G_2)$.

5.4.1 Classical Ramsey numbers of graph classes

The first paper that studied the Ramsey numbers of some graph class published in 1969 [105]. In that paper Walker started the study of classical Ramsey numbers of the class of planar graphs $\mathcal{P}$. He established bounds for $R_{\mathcal{P}}(4, j)$ and $R_{\mathcal{P}}(5, j)$ using Heawood's five-color theorem and he showed that $R_{\mathcal{P}}(3, j) = 3(j-1)$. He also proved that $R_{\mathcal{P}}(i, j) = 4j-3$ for $i \geq 4$ and $j \geq 3$, assuming that the four-color conjecture holds; note that at that time the four-color theorem was not proved. From the fact that $R_{\mathcal{P}}(1, j) = R_{\mathcal{P}}(i, 1) = 1$ for all $i, j \geq 1$, Walker had shown that, assuming the correctness of the conjecture, all the classical Ramsey numbers of planar graphs are determined. The four-color theorem which was conjectured in the mid-19th century was finally proved in 1989 [2, 107]. As soon as the conjecture was proved, the values of all classical Ramsey numbers of planar graphs became hale due to Walker. Four years after the proof, Steinberg and Tovey [101] not being aware of the result by Walker applied the four-color theorem and formulated the classical Ramsey numbers of planar graphs. The following theorem establishes all such Ramsey numbers of planar graphs.

Theorem 5.4.1. [101, 105] *The Ramsey numbers of planar graphs are:*

- $R_{\mathcal{P}}(2, j) = j$, $R_{\mathcal{P}}(i, 2) = i$ *where* $i \leq 4$*;*
- $R_{\mathcal{P}}(3, j) = 3j-3$*;*
- $R_{\mathcal{P}}(i, j) = 4j-3$, *where* $i \geq 4$ *and* $(i, j) \neq (4, 2)$.

For any positive integer k, let $\mathcal{G}_k$ be the class of graphs with maximum degree at most k. Staton [100] determined the exact values of $R_{\mathcal{G}_3}(3,j)$ for all $j \geq 1$. The Ramsey numbers $R_{\mathcal{G}_3}(4,j)$ for all $j \geq 1$ were obtained by Fraughnaugh and Locke [55]. In the same paper, they also determined the exact values of $R_{\mathcal{G}_4}(4,j)$ for all $j \geq 1$. The numbers $R_{\mathcal{G}_4}(3,j)$ for all $j \geq 1$ had previously been obtained by Fraughnaugh Jones [56].

Matthews in 1985 [85] studied the classical Ramsey numbers of claw-free graphs. Recall that none of the Ramsey numbers $R(3,j)$ is known when $j \geq 10$. Matthews showed that $R_{\mathcal{C}}(3,j) = \frac{5j-3}{2}$ where the index $\mathcal{C}$ denotes the class of claw-free graphs. Matthews also proved that $R_{\mathcal{C}}(i,3) = R(i,3)$. It means that the values of $R_{\mathcal{C}}(i,3)$ are unknown for $i \geq 10$. This is an evidence to the fact that the Ramsey numbers of graph classes might not be symmetric. To the best of our knowledge, these are the only results on the classical Ramsey numbers of graph classes until 2012. In this Year we started a systematic investigation on classical Ramsey numbers of graph classes. We will mention an overview of the results in Subsection 5.4.3.

5.4.2 Other Ramsey numbers of graph classes

Other 2-color Ramsey numbers have been also studied on graph classes. Recall that there are not many known results on 2-color Ramsey numbers of cycle and cliques of general graphs. Therefore, here we only mention the results on 2-color Ramsey numbers of graph classes when G_1 is a cycle and G_2 is a clique. In the case that $G_1 = C_4$ and $G_2 \in \{K_5, K_6, K_7\}$ the exact values of Ramsey numbers of planar graphs have been determined.

Theorem 5.4.2. [9, 10, 111]

- $R_{\mathcal{P}}(C_4, K_5) = 13$,
- $R_{\mathcal{P}}(C_4, K_6) = 17$,
- $R_{\mathcal{P}}(C_4, K_7) = 20$.

Bielak [9] has proved an analogous lower bound to the bound that she had for $R(C_4, K_t)$.

Theorem 5.4.3. [9] *For each integer $t \geq 5$, $R_{\mathcal{P}}(C_4, K_{t+1}) \geq 3t + [\frac{t}{5}] + 1$.*

5.4.3 More on the classical Ramsey numbers of graph classes

One of the papers that constitutes the scientific contribution of this thesis, paper VI [5], is about the classical Ramsey numbers of graph classes. For simplicity, hereafter we will use the phrase Ramsey number instead of the phrase classical Ramsey number. Our main goal of this work is to find the classes of graphs for which we manage to determine all their Ramsey numbers. Since finding all Ramsey numbers of claw-free graphs is as hard as general graphs, we concentrated on subclasses of claw-free graphs to see which subclasses are suitable for our purpose. For some subclasses of claw-free graphs such as line graphs, fuzzy circular interval graphs, fuzzy linear interval graphs and long circular interval graphs we determined all their Ramsey numbers. We also found all Ramsey numbers of circular arc graphs, cactus graphs, perfect graphs and many of its subclasses like chordal graphs, interval graphs, split graphs, bipartite graphs, and forests. Here we would like to show how to obtain all Ramsey numbers of line graphs.

Let us denote the class of all line graphs by $\mathcal{L}$. It is easy to see that $R_{\mathcal{L}}(1,j)=1$ and $R_{\mathcal{L}}(2,j)=j$. The first non-trivial case is when $i=3$.

Theorem 5.4.4. [85] *For every integer $j\geq 1$, $R_{\mathcal{L}}(3,j)=\lfloor(5j-3)/2\rfloor$.*

The following two results are used in order to determine the Ramsey numbers $R_{\mathcal{L}}(i,j)$ for $i\geq 4$.

Lemma 5.4.1. *Let H be a graph, let $G=L(H)$ be the line graph of H, and let $i\geq 4$ and $j\geq 1$ be two integers. Then H has a vertex of degree at least i if and only if G has a clique of size i. Moreover, H has a matching of size j if and only if G has an independent set of size j.*

Theorem 5.4.5. [3, 6, 26] *Let $i\geq 4$ and $j\geq 1$ be two integers, and let H be an arbitrary graph such that $\Delta(H)<i$ and $\nu(H)<j$. Then*

$$|E(H)|\leq \begin{cases} i(j-1)-(t+r)+1 & \text{if } i=2k \\ i(j-1)-r+1 & \text{if } i=2k+1, \end{cases}$$

where $j=tk+r$, $t\geq 0$ and $1\leq r\leq k$, and this bound is tight.

Note that for every $i\geq 4$ and $j\geq 1$, the values of t,k and r are uniquely defined.

Lemma 5.4.1 and Theorem 5.4.5 yield the formula for all the Ramsey numbers of line graphs when $i\geq 4$. Therefore, to calculate the Ramsey numbers of line graphs it suffices to prove Theorem 5.4.5. The theorem was first proved in 1976 by Chvátal and Hansen [26]. The method that they applied to prove the theorem has a flavor of linear programming. In 2008 a new proof of the theorem was given by Balachandran and Khare [3]. Their proof is somehow structural; however after finding a formula as an upper bound for the number of edges of such graphs, they use optimization methods to maximize the value of the formula without any reference to the graph. Their motivation to produce such a result was to obtain the extremal value of the following extremal graph theory problem: what is the least number of edges m such that every graph on m edges has a vertex of degree i or a matching of size j?

While working on a preliminary version of paper VI, before becoming aware of the existing proofs, we proved Theorem 5.4.5 from scratch using only graph theoretic arguments. Since this proof does not appear in VI [6], we present it here. The rest of this chapter is dedicated to this proof.

The following definition is to simplify and shorten the notations and argument in the proof.

Definition 34. *For every pair of integers $i\geq 4$ and $j\geq 1$,*

$$\rho(i,j)=\begin{cases} i(j-1)-(t+r)+1 & \text{if } i=2k \\ i(j-1)-r+1 & \text{if } i=2k+1, \end{cases}$$

where $j=tk+r$, $t\geq 0$ and $1\leq r\leq k$.

Therefore instead of Theorem 5.4.5, equivalently we prove the following theorem.

Theorem 5.4.6. *Let $i \geq 4$ and $j \geq 1$ be two integers. Every graph H with $\Delta(H) < i$ and $\nu(H) < j$ has at most $\rho(i,j)$ edges, and this bound is tight, i.e., there exists a graph H with $\Delta(H) < i$ and $\nu(H) < j$ that has exactly $\rho(i,j)$ edges.*

Definition of factor-critical graphs and Gallai's lemma could be used to shrink the proof significantly. As we promised to use elementary definitions, we give the complete proof. However, at the point of the proof that applying the lemma would conclude the proof, we will explain it and then we continue the proof ignoring the lemma.

Definition 35 (Factor-critical)**.** *A graph G is called* factor-critical *if for every vertex $u \in V(G)$, graph $G-u$ has a perfect matching.*

Theorem 5.4.7 (Gallai's lemma)**.** [89] *If graph G is connected and $\nu(G-u) = \nu(G)$ for each $u \in V(G)$, then G is factor-critical.*

Now we are ready to give the proof of Theorem 5.4.6.

Proof of Theorem 5.4.6. We prove the first statement of the theorem by induction on j. It is easy to verify that this statement holds when $j = 1$. For the induction hypothesis, we assume that for every $i \geq 4$ and $1 \leq \ell < j$, any graph H with $\Delta(H) < i$ and $\nu(H) < \ell$ contains at most $\rho(i,\ell)$ edges.

Now let $i \geq 4$ and $j \geq 2$ be two integers, let $k = \lfloor i/2 \rfloor$, and let $j = tk + r$, where $t \geq 0$ and $1 \leq r \leq k$. Let H be a graph with $\Delta(H) < i$ and $\nu(H) < j$. We will prove that H has at most $\rho(i,j)$ edges. We do this using a sequence of lemmas. In Lemmas 5.4.2–5.4.7, we prove that the first statement of Theorem 5.4.6 holds if H satisfies certain structural properties. After Lemma 5.4.7, we then assume that none of the Lemmas 5.4.2–5.4.7 can be applied on H. In the subsequent lemmas, we then prove several structural properties that H must have. Finally, we use this structure to show that H has at most $\rho(i,j)$ edges.

Note that we may assume, without loss of generality, that H does not contain isolated vertices.

Lemma 5.4.2. *If H is disconnected, then $|E(H)| \leq \rho(i,j)$.*

Proof. *Since H has no isolated vertices by assumption, H has two connected components H_1 and H_2 that contain at least one edge.*

Suppose H is disconnected, and let $H_1, \cdots, H_l$ be connected components of H. Define H'_1 to be H_1 and H'_2 to be $H_2 \cup \cdots \cup H_l$. Let M'_p be $M \cap E(H'_p)$, $p = 1,2$. It is clear that M'_p is maximum matching in H'_p, otherwise there will be a bigger matching than M for H. Since $|M'_p| \leq j-2$ we can apply induction hypothesis for H'_p. From the fact that $\rho(i,j)$ is an increasing function of i, it is obvious that $|E(H'_p)| \leq \rho(i,j)$, $p = 1,2$, no matter what is the maximum degree in H'_p. Thus in calculation of t_p and r_p we apply k that comes from either $i = 2k$ or $i = 2k+1$. Let us assume $|M'_p| = j_p - 1$ for $p = 1,2$, which $j_p = t_p k + r_p$ and $1 \leq r_p \leq k$. Since $|M'_1| + |M'_2| = |M|$, so $j_1 + j_2 = j+1$. We show also that $t_1 + t_2 \in \{t-1, t\}$. If $t_1 + t_2 \geq t+1$, then the equality $(t_1+t_2)k + r_1 + r_2 = tk + r + 1$ that comes from $j_1 + j_2 = j+1$ does not hold since, both r_1 and r_2 are at least one and $r \leq k$. If $t_1 + t_2 \leq t-2$ then, from the equality $(t_1+t_2)k + r_1 + r_2 = tk + r + 1$ we obtain $tk + r + 1 \leq (t-2)k + r_1 + r_2$ that leads to

$2k+r+1 \le r_1+r_2$ that is a contradiction to the fact that r_1 and r_2 are at most k and $r \ge 1$. Hence, $t_1+t_2 \in \{t-1,t\}$. We know $(t_1+t_2)k+(r_1+r_2) = j_1+j_2 = j+1$ so having $t_1+t_2 \in \{t-1,t\}$ gives

$$r_1+r_2 = j+1-(t_1+t_2)k \ge j+1-tk. \qquad (*)$$

First suppose $i = 2k+1$. Hence,

$$\begin{array}{rcll} |E(H)| & = & |E(H_1')|+|E(H_2')| & \\ & \le & [i(j_1-1)-r_1+1]+[i(j_2-1)-r_2+1] & \\ & \le & i(j_1+j_2-2)-(r_1+r_2)+2 & \\ & \le & i(j-1)-(j+1-tk)+2 & \text{; from } (*) \text{ and } j_1+j_2=j+1 \\ & = & i(j-1)-(tk+r+1-tk)+2 & ;\ j=tk+r \\ & = & i(j-1)-r+1. & \\ & = & \rho(i,j). & \end{array}$$

Now assume $i = 2k$.

$$\begin{array}{rcl} |E(H)| & = & |E(H_1')|+|E(H_2')| \\ & \le & [i(j_1-1)-(t_1+r_1)+1]+[i(j_2-1)-(t_2+r_2)+1] \\ & \le & i(j-1)-(t_1+t_2+r_1+r_2)+2 \end{array}$$

Consider two cases. First assume $t_1+t_2 = t-1$. Observe that $j_1+j_2 = j+1$ means $(t_1+t_2)k+r_1+r_2 = tk+r+1$ and then having $t_1+t_2 = t-1$ shows that $r_1+r_2 = k+r+1$. Then,

$$\begin{array}{rcl} |E(H)| & \le & i(j-1)-(t_1+t_2+r_1+r_2)+2 \\ & \le & i(j-1)-(t-1+k+r+1)+2 \\ & = & i(j-1)-(t+r)-k+2 \end{array}$$

Hence $|E(H)| \le \rho(i,j)$ since, $k \ge 2$.
So, assume as the second case $t_1+t_2 = t$. Again combining $(t_1+t_2)k+r_1+r_2 = tk+r+1$ and $t_1+t_2 = t$ gives $r_1+r_2 = r+1$ and so,

$$\begin{array}{rcl} |E(H)| & \le & i(j-1)-(t_1+t_2+r_1+r_2)+2 \\ & \le & i(j-1)-(t+r+1)+2 \\ & = & i(j-1)-(t+r)+1 \\ & = & \rho(i,j) \end{array}$$

This completes the proof of Lemma 5.4.2. △

Lemma 5.4.3. *If $\nu(H) < j-1$, then $|E(H)| \le \rho(i,j)$.*

Proof. *Suppose $\nu(H) = j' < j-1$. Let H' be the graph obtained from H by adding $j-1-j'$ connected components to H, each isomorphic to K_2. Since H' is disconnected, $|E(H')| \le \rho(i,j)$ due to Lemma 5.4.2. Hence $|E(H)| < |E(H')| \le \rho(i,j)$.* △

Lemma 5.4.4. *If H has a vertex u such that $\nu(H-u) < j-1$, then $|E(H)| \le \rho(i,j)$.*

Proof. *Suppose H has a vertex u such that $\nu(H-u) < j-1$. Since the graph $H-u$ has maximum degree at most $i-1$ and matching number at most $j-2$, the induction hypothesis ensures that*

$$|E(H-u)| \leq \begin{cases} i(j-2)-(t'+r')+1 & i=2k \\ i(j-2)-r'+1 & i=2k+1, \end{cases}$$

where $j-1 = t'k+r'$, $t' \geq 0$ and $1 \leq r' \leq k$.

Recall that $k = \lfloor i/2 \rfloor$ and $j = tk+r$, where $t \geq 0$ and $1 \leq r \leq k$. Hence $j-1 = tk-(r-1)$ when $r \geq 2$, and $j-1 = (t-1)k+k$ when $r=1$. Observe that $t'+r' \geq t+r-1$ and $r' \geq r-1$ in both cases. Hence, we have

$$|E(H-u)| \leq \begin{cases} i(j-2)-t+r+2 & i=2k \\ i(j-2)-r+2 & i=2k+1. \end{cases}$$

Since $|E(H)| = |E(H-u)|+d(u)$ and $d(u) \leq \Delta(H) \leq i-1$, it is clear that $|E(H)| \leq \rho(i,j)$. $\triangle$

Lemma 5.4.5. *If H has a maximum matching M such that there is at most one M-unsaturated vertex, then $|E(H)| \leq \rho(i,j)$.*

Proof. *If $|V(H) \setminus V(M)| = 0$, then $|V(H)| \leq 2(j-1)$. Since $\Delta(H) \leq i-1$, we have $|E(H)| \leq (i-1)(j-1) \leq \rho(i,j)$, where the last inequality follows from the fact that $k \geq 2$ and $t+r-1 \leq tk+r-1 = j-1$.*

Suppose $V(H) \setminus V(M) = \{x\}$ and $t \geq 1$. Note that

$$2|E(H)| = \sum_{v \in V(H)} d(v) = d(x) + \sum_{v \in V(M)} d(v) \leq (i-1)+2(i-1)(j-1).$$

First consider the case that $i = 2k$.

$$\begin{array}{rcll} |E(H)| & \leq & \frac{i-1}{2}+(i-1)(j-1) & \\ & = & \frac{i-1}{2}+i(j-1)-(j-1) & \\ & = & \frac{i-1}{2}+i(j-1)-(tk+r-1) & \\ & = & k-\frac{1}{2}+i(j-1)-(tk+r-1) & ;\, i=2k \\ & = & i(j-1)-(t-1)k-r+\frac{1}{2} & \\ & \leq & i(j-1)-(t-1)2-r+\frac{1}{2} & ;\, t \geq 1 \text{ and } k \geq 2 \\ & = & i(j-1)-(t+r)-t+\frac{5}{2} & \\ & \leq & i(j-1)-(t+r)+\frac{3}{2}. & ;\, t \geq 1 \end{array}$$

Since $|E(H)|$ is an integer, we have $|E(H)| \leq i(j-1)-(r+t)+1 = \rho(i,j)$.
Now assume $i = 2k+1$.

$$\begin{array}{rcll} |E(H)| & \leq & \frac{i-1}{2}+(i-1)(j-1) & \\ & = & \frac{i-1}{2}+i(j-1)-(j-1) & \\ & = & \frac{i-1}{2}+i(j-1)-(tk+r-1) & \\ & = & k+i(j-1)-(tk+r-1) & ;\, i=2k+1 \\ & = & i(j-1)-(t-1)k-r+1 & \\ & \leq & i(j-1)-r+1. & ;\, (t-1)k \geq 0 \\ & = & \rho(i,j). & \end{array}$$

Finally, suppose $V(H) \setminus V(M) = \{x\}$ and $t = 0$. Then $j \leq r \leq k$, and $|M| = j-1 \leq k-1$. For each $v \in V(M)$, $d(v) \leq 2(j-2)+2 = 2j-2 \leq 2k-2$ which yields $d(v) \leq i-2$ since, $i \geq 2k$. For every $v \in V(H) \setminus V(M)$, $d(v) \leq 2(j-1)$. Then $2|E(H)| = \sum_{v \in V(H)} = d(x) + \sum_{v \in V(M)} \leq 2(j-1) + 2(j-1)(i-1)$ so, $|E(H)| \leq (j-1)+(j-1)(i-1) = (i-1)(j-1) \leq \rho(i,j)$. △

Let us now explain how applying Gallai's lemma concludes the proof. By Lemma 5.4.4, for every vertex $u \in V(H)$ it holds that $\nu(H-u) = \nu(H)$. Therefore due to Gallai's lemma, Lemma 5.4.7, H is factor-critical. Hence for every vertex $u \in V(H)$, $H-u$ has a perfect matching. It implies that for every maximum matching M in H there is only one M-unsaturated vertex. So, Lemma 5.4.5 has already given the desired contradiction to complete the proof.

For the sake of completeness we want to present the whole proof and therefore we continue the proof without using Gallai's lemma. Getting contradiction in the proof of Lemma 5.4.5 convinces us that for any maximum matching of H there should be at least two unsaturated vertices.

Lemma 5.4.6. *If H has a maximum matching M such that there is an M-unsaturated vertex of degree 1, then $|E(H)| \leq \rho(i,j)$.*

Proof. *Suppose H has a maximum matching M such that there is an M-unsaturated vertex x that has degree 1 in H. Let u be the unique neighbor of x. Suppose $H-u$ has a matching M' of size $j-1$. Then $M' \cup \{ux\}$ is a matching of size j in H, contradicting the assumption that $\nu(H) \leq j-1$. Hence $\nu(H-u) < j-1$. Then $|E(H)| \leq \rho(i,j)$ as a result of Lemma 5.4.4.* △

Lemma 5.4.7. *If H has a maximum matching M such that there is an M-saturated vertex that has at least two M-unsaturated neighbors, then $|E(H)| \leq \rho(i,j)$.*

Proof. *Suppose H has a maximum matching M such that there is an M-saturated vertex u that has at least two M-unsaturated neighbors. Let v be the neighbor of u such that $uv \in M$. Let $R \subseteq E(H)$ be the set of edges of H that are incident with an M-unsaturated vertex, and let $T = E(H) \setminus (M \cup R)$. Note that M, R and T form a partition of $E(H)$. We define $V(M)$ to be the set of endpoints of edges of M.*

We prove that $\nu(H-u) < j-1$, which implies that $|E(H)| \leq \rho(i,j)$ due to Lemma 5.4.4. For contradiction, suppose $H-u$ has a matching M' of size $j-1$. Let Y be the subgraph of H induced by the vertices of all M'-alternating paths in H starting at vertex u and not using edges of $T \setminus M'$. Let Z be the subgraph of H induced by the vertices of $V(H) \setminus V(Y)$.

Claim 5.4.1. *There is no edge of M' between Y and Z.*

Proof of Claim 1. Let $y \in Y \setminus \{u\}$. By the definition of Y, there is an M'-alternating path P from u to y not using edges of $T \setminus M'$. Let xy be the last edge of P. We know that y is M'-saturated, as otherwise P would be an M'-augmenting path, contradicting the assumption that $\nu(H) \leq j-1$. Let $e \in M'$ be the edge that saturates y. We will show that $e \in E(Y)$. If $e = xy$, then this is true since both x and y belong to Y. Otherwise, the path obtained from P by adding the edge e is an M'-alternating path from u, and e belongs to $E(Y)$ by the definition of Y. We conclude that every vertex in $Y \setminus \{u\}$ is M'-saturated by an edge of M' both endpoints of which belong to Y. Since u is M'-unsaturated, the claim follows. ◇

Claim 5.4.2. *There is no edge of M between Y and Z.*

Proof of Claim 2. *For contradiction, suppose there is an edge $yz \in M$ with $y \in V(Y)$ and $z \in V(Z)$. By Claim 1, $yz \notin M'$. Let P be an M'-alternating path from u to y not using edges of $T \setminus M'$. Note that $yz \notin E(P)$, as otherwise z would belong to Y. Let xy be the last edge of P. Suppose $xy \in M'$. Then adding yz to P yields an M'-alternating path from u to z not using edges of $T \setminus M'$. This implies that z belongs to Y, contradicting the assumption that $z \in V(Z)$. Hence we must have $xy \notin M'$. Then $xy \in R$, as P does not use edges of $T \setminus M'$. Let x' be the vertex preceding x on P. Note that $x' \in V(M)$ and $x'x \in M'$. Also note that $x' \neq u$, since u is M'-unsaturated, and $x' \neq v$, since v is not incident with an edge of R as otherwise H would have an M-augmenting path.*

Let w be the neighbor of u on P; we either have $w = v$ or w is an M-unsaturated neighbor of u. Let w' be the neighbor of w on P. Consider the subpath uPx. We distinguish two cases, and obtain a contradiction in both cases.

First suppose $w \neq v$. Let ss' denote the edge of $E(uPx) \cap R \cap M'$ closest to x on path uPx such that $s' \in V(M)$, $s \in V(H) \setminus V(M)$ and s' is closer to x than s on path uPx. Such an edge always exists, since we can have $s = w$ and $s' = w'$. Now, consider the subpath sPx and let ll' denote the edge of $E(sPx) \cap R$ closest to s' on path uPx such that $l' \in V(M)$, $l \in V(H) \setminus V(M)$, $l \neq s$ and l' is closer to s' than l on path uPx. Note that we cannot have $s' = l' = x'$, since ss' and xx' both belong to M'. If $l' = s' \neq x'$, then $ll' \notin M'$ and there is an edge $ll'' \in R \cap M' \cap V(uPx)$, which contradicts the choice of ss'. Therefore we have $l' \neq s'$ and $ll' \in M'$, since $ll' \notin M'$ would yield the existence of the edge $ll'' \in R \cap M' \cap E(uPx)$. Notice that $s'Pl'$ contains no edge of R by construction. Moreover, by the definition of Y, $s'Pl'$ also does not contain any edge of $T \setminus M'$. Therefore $s'Pl'$ is an M-alternating path. Moreover, note that $ss', ll' \in M'$, and hence $s'Pl'$ starts and ends with edges of M. Thus sPl is an M-augmenting path, contradicting the assumption that M is a maximum matching in H.

Now suppose $w = v$. Let w'' be an M-unsaturated neighbor of u. If there exists an edge ss' in $E(uPx) \cap R \cap M'$ such that $s' \in V(M)$, $s \in V(H) \setminus V(M)$ and s' is closer to x than s, then we obtain a contradiction as before. Suppose such an edge ss' does not exist. Let $t't$ be the edge of $E(uPx) \cap R$ closest to u on path uPx, where t' appears before t on uPx. Note that $t't \in M'$. Note that such an edge $t't$ exists, as it is possible that $x' = t'$ and $x = t$. Then the path obtained from uPt by adding the edge uw'' is an M-augmenting path, contradicting the maximality of M. This completes the proof of Claim 2. ◇

Let $M'' \subseteq M'$ be the set of edges of M' that saturate vertices of Y. Recall that u has at least two M-unsaturated neighbors, and both of these belong to Y by definition. This, together with Claim 2, implies that $|V(Y)| \geq 2 \cdot |M \cap E(Y)| + 2$. As we saw in the proof of Claim 1, every vertex of $V(Y) \setminus \{u\}$ is M'-saturated. Hence $|M''| \geq |M \cap E(Y)| + 1$. Now consider the set $M^* = M'' \cup (M \cap E(Z))$. Then $|M^*| \geq |M| + 1 = j$. As a result of Claims 1 and 2, M^* is a matching. This contradicts the assumption that $\nu(H) \leq j - 1$ and completes the proof of Lemma 5.4.7. △

For the remainder of this proof, we assume that none of the above lemmas can be applied. We fix a maximum matching M of H. Let $R \subseteq E(H)$ be the set of edges of H that are incident with an M-unsaturated vertex, and let $T = E(H) \setminus (M \cup R)$. For

$p \in \{0,1,2\}$, we define the set

$$M_\ell = \{e \in M \mid \ell \text{ endpoints of } e \text{ have an } M\text{-unsaturated neighbor}\}.$$

Note that if for an edge $e \in M$ both endpoints of e have an M-unsaturated neighbor, then both endpoints are adjacent to the same M-unsaturated vertex, as otherwise there would be an M-augmenting path in H. For every $x \in V(H) \setminus V(M)$, let $M(x) = \{e \in M \mid \text{at least one endpoint of } e \text{ is adjacent to } x\}$, and let the *$M$-component* of x be the subgraph of H induced by x and the endpoints of the edges in $M(x)$. Note that the edges of M_0 do not belong to any M-component, and that every edge of $M_1 \cup M_2$ belongs to exactly one M-component due to the assumption that Lemma 5.4.7 cannot be applied.

Lemma 5.4.8. *Let $x \in V(H) \setminus V(M)$. Every edge in $M(x)$ belongs to an M-blossom whose unique M-unsaturated vertex is x.*

Proof. *Let $uv \in M(x)$. By definition, x is adjacent to at least one of the endpoint of uv. Without loss of generality, suppose u is adjacent to x. Let M' be a maximum matching in $H-u$. We know that $|M'| = j-1$, as otherwise we could apply Lemma 5.4.4. Let Y be the subgraph of H induced by the vertices of all M'-alternating paths in H starting at vertex u and not using edges of $T \setminus M'$. Note that Y contains at least one M-unsaturated vertex, namely x. If Y contains more than one M-unsaturated vertex, then we obtain a contradiction by the same arguments as in the proof of Lemma 5.4.7. Hence x is the only M-unsaturated vertex in Y.*

Let P be a longest M'-alternating path from u that start with edge ux and contains no edges of $T \setminus M'$. Let xx' be the second edge of P, where $x' \in V(M)$. We claim that P ends in vertex v. For contradiction, suppose P ends in a vertex $y \neq v$. Since we showed that x is the only M-unsaturated vertex in Y, we know that after the first two edges ux and xx', path P alternately contains edges of $M \setminus M'$ and $T \cap M'$. This means that the subpath of P from x to y is M-alternating, and that $y \in V(M)$. Vertex y is M'-saturated, as otherwise there would be an M'-augmenting path from u to y in H, contradicting the assumption that $\nu(H) \leq j-1$. Suppose the edge $e \in M'$ that saturates y belongs to P. Then $e \in T \cap M'$, which means that we can extend P by adding the edge of M incident with y. Now suppose e does not belong to P. Then the last edge of P belongs to $M \setminus M'$, which means we can extend P by adding the edge e. This contradiction shows that P ends in v. Recall that the subpath of P from x to v is M-alternating. Hence adding the edge vu to the path P yields the desired blossom in which x is the only M-unsaturated vertex. △

Lemma 5.4.9. *Let C be an M-blossom whose unique M-unsaturated vertex is x, and let $y \neq x$ be a vertex of $V(H) \setminus V(M)$. Then there is no M-alternating path from y to any vertex of C.*

Proof. *For contradiction, suppose there is an M-alternating path P from y to a vertex $z \in V(C)$. We may without loss of generality assume that none of the internal vertices of P belongs to C, as otherwise we can consider the unique subpath of P for which only the last vertex belongs to C. If $z = x$, then P is an M-augmenting path, contradicting the assumption that M is a maximum matching in H. If $z \in V(C) \notin \{x\}$, then P can be extended into an M-augmenting path from y to x by traversing the cycle C in the appropriate direction.* △

Let $x \in V(H) \setminus V(M)$ and let $M(x) = \{u_1v_1, \ldots, u_sv_s\}$. For every $p \in \{1, \ldots, s\}$, let C_p be an M-blossom containing u_pv_p and vertex x, which exists due to Lemma 5.4.8. Let X be the subgraph of H induced by the vertices of all these blossoms C_p. Note that X is a supergraph of the M-component $M(x)$. We call X the *extended M-component* of x. An induced subgraph of H is an extended M-component if it is an extended M-component of some vertex $x \in V(H) \setminus V(M)$. We point out that, unlike the M-component of x, the extended M-component of x is not uniquely defined but depends on the choice of the blossoms C_p. However, for the remainder of the proof, we fix the extended M-components of every vertex $x \in V(H) \setminus V(M)$, which will allow us to speak of *the* extended M-components.

By definition, any two M-components of H are vertex-disjoint. We now show that the same holds for any two extended M-components.

Lemma 5.4.10. *There is no M-alternating path between any two extended M-components. In particular, any two extended M-components are vertex-disjoint.*

Proof. *Let $x_1, x_2 \in V(H) \setminus V(M)$ and let X_1 and X_2 be the extended M-components of x_1 and x_2, respectively. Suppose, for contradiction, that there is an M-alternating path P from some vertex $z_1 \in V(X_1)$ to some vertex $z_2 \in V(X_2)$. We may without loss of generality assume that none of the internal vertices of P belongs to $V(X_1) \cup V(X_2)$, as otherwise we can simply consider the unique subpath of P from X_1 to X_2 that does satisfy this property. By the definition of X_p, vertex z_p is contained in an M-blossom C_p whose M-unsaturated vertex is x_p, for $p = 1, 2$. By traversing C_1 in the appropriate direction, we obtain an M-alternating path from x_1 to z_2. This contradicts Lemma 5.4.9. Note that we obtain the same contradiction if we assume X_1 and X_2 to have a common vertex z, since in that case we can take $z_1 = z_2 = z$.* $\triangle$

Let $M_0' = \{e \in M_0 \mid e \text{ does not belong to any extended } M\text{-component}\}$. A path in H is called an *M_0'-path* if all its internal vertices belong to $V(M_0')$. The following lemma implies that $M_0' \neq \emptyset$.

Lemma 5.4.11. *There exist two extended M-components X_1 and X_2 in H such that there is an M_0'-path between X_1 and X_2.*

Proof. *Since we assume Lemma 5.4.5 cannot be applied, there are at least two M-unsaturated vertices in H. By definition, every extended M-component contains exactly one M-unsaturated vertex, so H has at least two extended M-components. By Lemma 5.4.10, any two extended M-components are vertex-disjoint. Due to the assumption that Lemma 5.4.2 cannot be applied, H is connected. Since every edge of $M_1 \cup M_2$ belongs to some extended M-component and there are no edges between any two extended M-components as a result of Lemma 5.4.10, every path between any two extended M-components contains at least one vertex of $V(M_0')$. In particular, there must be two extended M-components X_1 and X_2 such that there is a path between them all whose internal vertices belong to $V(M_0')$, and such a path is an M_0'-path by definition.*
$\triangle$

Let X_1 and X_2 be two extended M-components such that there is an M_0'-path P between X_1 and X_2. Let $x_1' \in V(X_1)$ and $x_2' \in V(X_2)$ be the end-vertices of P. Recall the set T consisting of all edges in $E(H) \setminus M$ both endpoints of which belong to $V(M)$. By definition, the path P contains only edges of T and M_0'. A vertex $w \in V(P)$ is called *weird* if w is an internal vertex of P and both edges of P that are incident with w belong to T. The set of all weird vertices of P is denoted by $\mathcal{W}$. Let $w \in \mathcal{W}$. For $1 \leq \ell \leq 2$, a *shortcut* from w to X_ℓ (or, equivalently, from X_ℓ to w) is an M-alternating M'_0-path P_w from w to X_ℓ such that for every edge in $M \cap E(P_w)$ at least one endpoint belongs to $V(x'_1Pw)$. We define $\mathcal{W}_\ell \subseteq \mathcal{W}$ to be the set of weird vertices w of P that have a shortcut to X_ℓ containing the M'_0-neighbor of w. Note that $\mathcal{W}_1$ and $\mathcal{W}_2$ are not necessarily disjoint. A weird vertex w is *abnormal* if it has no shortcut to X_ℓ containing the M'_0-neighbor w' of w for $1 \leq \ell \leq 2$, i.e., if $w \in \mathcal{W} \setminus (\mathcal{W}_1 \cup \mathcal{W}_2)$.

Lemma 5.4.12. *Let X_1 and X_2 be two extended M-components. If there is an M'_0-path P between X_1 and X_2 that contains at least one abnormal vertex, then for $1 \leq \ell \leq 2$ there is a shortcut from X_ℓ to the abnormal vertex of P that is closest to X_ℓ on the path P.*

Proof. *Let P be an M'_0-path P between X_1 and X_2 that contains at least one abnormal vertex. Let x'_1 and x'_2 be the end-vertices of P, where $x'_1 \in V(X_1)$ and $x'_2 \in V(X_2)$. Let $\mathcal{W}$ be the set of weird vertices of P, and let $\mathcal{W}_1$ and $\mathcal{W}_2$ be the sets consisting of the weird vertices w that have a shortcut to X_1 and X_2, respectively, containing the M'_0-neighbor of w. By assumption, the set $\mathcal{W} \setminus (\mathcal{W}_1 \cup \mathcal{W}_2)$ of abnormal vertices of P is non-empty. For $1 \leq \ell \leq 2$, let u_ℓ be the abnormal vertex closest to X_ℓ on P, where it is possible that $u_1 = u_2$. By symmetry, it suffices to show that there is a shortcut from X_1 to u_1.*

If the subpath x'_1Pu_1 contains no weird vertices other than u_1, then this subpath is the desired shortcut. Suppose x'_1Pu_1 contains at least one weird vertex other than u_1. We claim that x'_1Pu_1 does not contain a vertex of $\mathcal{W}_2$. For contradiction, suppose x'_1Pu_1 does contain such a vertex, and let w_2 be the vertex of $\mathcal{W}_2$ that is closest to x'_1 on the path x'_1Pu_1. By the definition of $\mathcal{W}_2$, there is a shortcut P_{w_2} from X_2 to w_2 that contains the M'_0-neighbor of w_2. Note that, by definition, the paths P_{w_2} and $w_2Px'_1$ are internally vertex-disjoint. Hence, if $w_2Px'_1$ does not contain any vertex of $\mathcal{W}_1$, then the union of the paths P_{w_2} and $w_2Px'_1$ forms an M-alternating path from X_2 to X_1, contradicting Lemma 5.4.10. Suppose $w_2Px'_1$ contains at least one vertex of $\mathcal{W}_1$. Let w_1 be the vertex of $\mathcal{W}_1$ that is closest to w_2 on $w_2Px'_1$, and let P_{w_1} be a shortcut from X_1 to w_1 containing the M'_0-neighbor of w_1. Then the union of the paths P_{w_1}, w_1Pw_2 and P_{w_2} forms an M-alternating path between X_1 and X_2, again yielding a contradiction. Hence the subpath x'_1Pu_1 contains no vertex of $\mathcal{W}_2$.

Recall that x'_1Pu_1 contains at least one weird vertex other than u_1 by assumption. From the above arguments it follows that it contains at least one vertex of $\mathcal{W}_1$. Let w be the vertex of $\mathcal{W}_1$ closest to u_1 on the path x'_1Pu_1, and let P_w be a shortcut from X_1 to w containing the M'_0-neighbor of w. Then the two (internally vertex-disjoint) M-alternating M'_0-paths P_w and wPu_1 together form a shortcut from X_1 to u_1. $\triangle$

Lemma 5.4.13. *Every M'_0-path between any two extended M-components contains at least two abnormal vertices.*

Proof. *Let $x_1, x_2 \in V(H) \setminus V(M)$ and let X_1 and X_2 be the extended M-components of x_1 and x_2, respectively. Let P be an M'_0-path between a vertex $x'_1 \in V(X_1)$ and a vertex $x'_2 \in V(X_2)$. Let $\mathcal{W}$ be the set of weird vertices of P. If $|\mathcal{W}| = 0$, then P is an M-alternating path between X_1 and X_2, contradicting Lemma 5.4.10. Hence P contains at least one weird vertex, i.e., $|\mathcal{W}| \geq 1$. For $1 \leq \ell \leq 2$, let $\mathcal{W}_\ell \subseteq \mathcal{W}$ be the set of weird* *vertices w that have a shortcut to X_ℓ containing the M_0'-neighbor of w. Recall that every vertex in $\mathcal{W} \setminus (\mathcal{W}_1 \cup \mathcal{W}_2)$ is abnormal by definition.*

Assume, for contradiction, that P contains at most one abnormal vertex. We distinguish two cases.

Case 1. P contains no abnormal vertex.

In this case, we have $\mathcal{W}_1 \cup \mathcal{W}_2 = \mathcal{W}$. Recall that $|\mathcal{W}| \geq 1$. Without loss of generality, assume that $|\mathcal{W}_1| \geq 1$. Let w_1 be the vertex of $\mathcal{W}_1$ that is furthest from x_1' on P. Let P_{w_1} be a shortcut from X_1 to w_1 containing the M_0'-neighbor of w_1. Suppose the subpath w_1Px_2' contains no vertex of $\mathcal{W}_2$. Then w_1Px_2' is M-alternating. From the definition of P_{w_1} it follows that the paths P_{w_1} and w_1Px_2' are internally vertex-disjoint. Hence, the union of P_{w_1} and w_1Px_2' forms an M-alternating path from X_1 to X_2, contradicting Lemma 5.4.10. Now suppose w_1Px_2' contains at least one vertex of $\mathcal{W}_2$. Let w_2 be the vertex of $\mathcal{W}_2$ that is closest to w_1 on the path w_1Px_2', and let P_{w_2} be a shortcut from X_2 to w_2 containing the M_0'-neighbor of w_2. The three M-alternating paths P_{w_1}, w_1Pw_2 and P_{w_2} are pairwise internally vertex-disjoint by definition, so their union forms an M-alternating path between X_1 and X_2. The existence of such a path contradicts Lemma 5.4.10.

Case 2. P contains exactly one abnormal vertex.

Let w^* be the abnormal vertex of P, and let w' be the M_0'-neighbor of w^*. By Lemma 5.4.12, there exists a shortcut P_1 from X_1 to w^* and a shortcut P_2 from X_2 to w^*. Since w^* is abnormal, neither P_1 nor P_2 contains w'. As a result, the union P^* of the paths P_1 and P_2 is an M_0'-path between X_1 and X_2 whose only weird vertex is w^*. Let x_1'' and x_2'' be the end-vertices of P^*, where $x_1'' \in V(X_1)$ and $x_2'' \in V(X_2)$. Let w_1 and w_2 be the neighbors of w^* on the subpaths $x_1''P^*w^*$ and $w^*P^*x_2''$, respectively. Since w^* is a weird vertex of P^*, the two edges of P^* incident with w^* belong to T. Hence $w_1 \neq w'$ and $w_2 \neq w'$.

We now show that there is a maximum matching M^* in H such that w^* is an M^*-saturated vertex that has two M^*-unsaturated neighbors, namely w_1 and w_2. Since we assumed that Lemma 5.4.7 cannot be applied, this yields the desired contradiction.

By Lemma 5.4.8 and the definition of an extended M-component, there exists an M-blossom C_ℓ such that $x_\ell'' \in V(C_\ell)$ and x_ℓ is the unique M-unsaturated vertex of C_ℓ for $1 \leq \ell \leq 2$. We consider the subgraph of H formed by the two M-blossoms C_1 and C_2 and the M_0'-path P^* between these blossoms. In C_1, the only M-unsaturated vertex is x_1. Let $M' \subseteq E(C_1)$ be the unique maximum matching in C_1 such that x_1'' is the only M'-unsaturated vertex in C_1. Note that $|M'| = |M \cap E(C_1)|$, and that $M' = M \cap E(C_1)$ if and only if $x_1 = x_1''$. Similarly, let M'' be the unique maximum matching in C_2 such that x_2'' is the only M''-unsaturated vertex in C_2. Now let M^* be the matching obtained from M by replacing the edges of $M \cap E(C_1)$ by the edges of M', replacing the edges of $M \cap E(C_2)$ by the edges of M'', and replacing the edges of $(M \cap E(P^*)) \setminus \{w^*w'\}$ by the edges of $(E(P) \setminus M) \setminus \{w^*w_1, w^*w_2\}$. Since $|M'| = |M \cap E(C_1)|$ and $|M''| = |M \cap E(C_2)|$ and w^* is the only weird vertex of P^*, we have that $|M^*| = |M|$. Hence M^* is a maximum matching in H, and there is an M^*-saturated vertex w^* that has two M^*-unsaturated neighbors w_1 and w_2. This completes the proof of Lemma 5.4.13. $\triangle$

An edge $uv \in M_0'$ is *associated* with an extended M-component X if $N(u) \cup N(v)$ contains at least one vertex of X. If an edge $uv \in M_0'$ is not associated with any extended

M-component, then uv is said to be *unassociated*. Let $M_0'' \subseteq M_0'$ denote the set of unassociated edges.

Lemma 5.4.14. *Every edge in M_0' is associated with at most one extended M-component, and every extended M-component has at least one edge of M_0' associated with it.*

Proof. *The first statement follows from the observation that if there is an edge $uv \in M_0'$ that is adjacent to two extended M-components X_1 and X_2, then there is an M_0'-path P from X_1 to X_2 with at most one abnormal vertex. For the second statement, it suffices to show that every extended M-component is adjacent to at least one vertex in $V(M_0')$. This follows from the assumption that H is connected, H contains at least two extended M-components, and there is no edge between any two extended M-components by Lemma 5.4.10.* △

The purpose of the next lemma is to show that for some abnormal vertices in H, their M_0'-neighbors can only be adjacent to certain vertices in H.

Lemma 5.4.15. *Let X_1 and X_2 be two extended M-components, and let P be an M_0'-path between X_1 and X_2. Let u_1 be the abnormal vertex of P that is closest to X_1 on the path P, and let u_2 be the abnormal vertex of P closest to X_2. Let v_1 and v_2 be the M_0'-neighbors of u_1 and u_2, respectively. For $1 \leq \ell \leq 2$, the only possible neighbors of v_ℓ in H other than u_ℓ are endpoints of edges in M_0' that are associated with X_ℓ, and endpoints of edges in M_0''. Moreover, v_ℓ is adjacent to at most one endpoint of any edge in M_0' that is associated with X_ℓ.*

Proof. *By symmetry, it suffices to prove the statement of the lemma for v_1. Vertex v_1 is not adjacent to any vertex in X_1 or X_2 due to the assumption that u_1 is an abnormal vertex of P. By Lemma 5.4.12, there is a shortcut P_{u_1} from X_1 to u_1. Since u_1 is abnormal, P_{u_1} does not contain v_1. Suppose v_1 is adjacent to some vertex $x \in V(X)$ for some extended M-component $X \notin \{X_1, X_2\}$. Then the path obtained from P_{u_1} by adding the path u_1v_1x is an M-alternating path between X_1 and X, contradicting Lemma 5.4.10.*

Now suppose v_1 is adjacent to a vertex x that is contained in an edge $xy \in M_0'$ associated with an extended M-component $X \neq X_1$. Since xy is associated with X, at least one of the vertices in $\{x,y\}$ is adjacent to a vertex of X. If y is adjacent to a vertex $z \in V(X)$, then the path P_{u_1} together with the path u_1v_1xyz is an M-alternating path between X_1 and X, yielding a contradiction. Hence x must be adjacent to some vertex $z \in V(X)$. The union of the paths P_{u_1} and u_1v_1xz is a path between X_1 and X that has exactly one weird vertex, namely x, and therefore at most one abnormal vertex. This is a contradiction to Lemma 5.4.12.

It remains to prove that v_1 is adjacent to at most one endpoint of any edge in M_0'. For contradiction, suppose v_1 is adjacent to both endpoints of some edge $pq \in M_0'$ that is associated with X_1. Since pq is associated with X_1, at least one endpoint of pq is adjacent to a vertex $x_1' \in V(X_1)$. Without loss of generality, suppose p is adjacent to x_1'. Then the M_0'-path $x_1'pqv_1u_1$ is a shortcut from X_1 to u_1 containing v_1, contradicting the assumption that u_1 is abnormal. △

We will now use the structure of H exhibited in the lemmas above to show that

$|E(H)| \leq \rho(i,j)$. It is clear that the following holds:

$$\sum_{v\in V(H)} d(v) = \sum_{v\in V(M)} d(v) + \sum_{v\in V(H)\setminus V(M)} d(v)\,. \tag{5.1}$$

Recall that $\Delta(H) \leq i-1$ and $\nu(H) \leq j-1$. In fact, since we assume that Lemma 5.4.3 cannot be applied, we know that $\nu(H) = j-1$. In particular, we have that $|M| = j-1$. Let $\mathrm{maxm} = 2(i-1)(j-1)$ and $\mathrm{maxr} = 2(j-1)$. Then we can write

$$\sum_{v\in V(M)} d(v) \leq \mathrm{maxm}\,. \tag{5.2}$$

Since we assume that Lemma 5.4.7 cannot be applied, every vertex in $V(M)$ is adjacent to at most one vertex of $V(H)\setminus V(M)$. This, together with the definition of the sets M_0, M_1 and M_2, implies that $\sum_{v\in V(H)\setminus V(M)} d(v) = 2|M_2| + |M_1|$. Note that $\mathrm{maxr} = 2(j-1) = 2|M| = 2|M_0| + 2|M_1| + 2|M_2|$, we means we can write

$$\sum_{v\in V(H)\setminus V(M)} d(v) = \mathrm{maxr} - 2|M_0| - |M_1|\,. \tag{5.3}$$

Inequalities (5.1), (5.2) and (5.3) together yield the following:

$$\sum_{v\in V(H)} d(v) \leq \mathrm{maxm} + \mathrm{maxr} - 2|M_0| - |M_1|\,. \tag{5.4}$$

Our goal is to prove that $|E(H)| \leq \rho(i,j)$, or equivalently that the following inequality holds:

$$\sum_{v\in V(H)} d(v) \leq 2\rho(i,j)\,. \tag{5.5}$$

We distinguish two cases, depending on whether or not i is even.

CASE 1. $i = 2k$.

In this case, $\rho(i,j) = i(j-1) - (t+r) + 1$. Then we have that

$$\begin{aligned} 2\rho(i,j) &= 2i(j-1) - 2(t+r) + 2 \\ &= 2(i-1)(j-1) + 2(j-1) - 2(t+r-1) \\ &= \mathrm{maxm} + \mathrm{maxr} - 2(t+r-1)\,. \end{aligned}$$

Hence, Inequality (5.5) can be rewritten as follows:

$$\sum_{v\in V(H)} d(v) \leq \mathrm{maxm} + \mathrm{maxr} - 2(t+r-1)\,. \tag{5.6}$$

By the arguments above, in order to prove that $|E(H)| \leq \rho(i,j)$, it suffices to prove that Inequality (5.6) holds.

If $|M_0| \geq t+r-1$, then the validity of Inequality (5.6) directly follows from Inequality (5.4). Suppose that $|M_0| = t+r-1-\alpha$ for some $\alpha \geq 1$. If $|M_1| \geq 2\alpha$, then again Inequality (5.4) immediately implies that Inequality (5.6) holds. Suppose $|M_1| = 2\alpha - \beta$ for some $1 \leq \beta \leq 2\alpha$. If we substitute $|M_0| = t+r-1-\alpha$ and $|M_1| = 2\alpha - \beta$ in Equation (5.3), we get the following:

$$\sum_{v\in V(H)\setminus V(M)} d(v) = \mathrm{maxr} - 2(t+r-1) + \beta\,. \tag{5.7}$$

It follows from Equations (5.1) and (5.7) that in order to satisfy Inequality (5.6), it suffices to prove that the following inequality holds:

$$\sum_{v\in V(M)} d(v) \le \mathrm{maxm} - \beta\,. \tag{5.8}$$

Subcase 1.1. *H has at most $t-1$ extended M-components.*

Suppose H has $t-\gamma$ extended M-components for some $\gamma \ge 1$. Then $|M_0'| \ge t-\gamma$ due to Lemma 5.4.14, and consequently $|M_0| \ge |M_0'| \ge t-\gamma$. We assumed that $|M_0| = t+r-1-\alpha$ for some $\alpha \ge 1$, which means that $\alpha \le r-1+\gamma$. The extended M-component X_1 of x_1 contains at most $k-1$ edges of M_2, since otherwise the degree of x_1 would be at least $2k = i$, contradicting the assumption that $\Delta(H) \le i-1$. Hence $|M_2| \le (t-\gamma)(k-1)$. On the other hand, we have that $|M_2| = t(k-1)+\beta-\alpha$, since $|M| = j-1 = tk+r-1$ and $|M_2| = |M|-|M_0|-|M_1|$. This means that $t(k-1)+\beta-\alpha \le (t-\gamma)(k-1)$, which implies that we have the following inequality:

$$\beta-\alpha \le (1-k)\gamma. \tag{5.9}$$

Recall that $\alpha \le r-1+\gamma$. Suppose $\alpha < r-1+\gamma$. Then Inequality (5.9), together with the observation that $r \le k$, implies that $\beta \le (2-k)(\gamma-1)$. Since $k \ge 2$ and $\gamma \ge 1$, this inequality only holds if $\beta \le 0$. This contradicts the fact that $\beta \ge 1$. Hence we must have $\alpha = r-1+\gamma$. Inequality (5.9) can now be rewritten as $\beta-r+1 \le (2-k)\gamma$. Since $r \le k$, this is equivalent to $\beta-1 \le (2-k)(\gamma-1)$. Since $k \ge 2$ and $\gamma \ge 1$, this inequality shows that $\beta \le 1$, and hence $\beta = 1$.

Since $\alpha = r-1+\gamma$, we have that $|M_0| = t+r-1-\alpha = t-\gamma$. Recall that the number of extended M-components of H is $t-\gamma$ in this subcase. Hence, as a result of Lemma 5.4.14, every extended M-component has exactly one edge of M_0' associated with it and every edge of M_0 is associated with one extended M-component (note that $M_0 = M_0'$). Since H is connected, H has at least two extended M-components, and there are no edges between any two extended M-components due to Lemma 5.4.10, there exist two edges $pq, p'q' \in M_0'$ such that there is an edge e with one endpoint in $\{p,q\}$ and the other in $\{p',q'\}$. Let X and X' be the extended M-components with which pq and $p'q'$ are associated, respectively. Then there exists an M_0'-path P' from X to X' whose internal vertices belong to $\{p,q,p',q'\}$. The path P' contains at least two abnormal vertices by Lemma 5.4.13. Then P' must also contain at least two weird vertices, which implies that P' contains exactly two vertices from $\{p,q,p',q'\}$. Without loss of generality, assume that p and p' are the internal vertices of P'. Then $d(q) = 1$ as a result of Lemma 5.4.15 and the fact that $M_0'' = \emptyset$. Hence we get

$$\begin{aligned}
\textstyle\sum_{v\in V(M)} d(v) &= d(q) + \textstyle\sum_{v\in V(M)\setminus\{q\}} d(v)\\
&\le 1+(2j-3)(i-1)\\
&= \mathrm{maxm} - i + 2\\
&\le \mathrm{maxm} - \beta\,,
\end{aligned}$$

where the last inequality follows from the assumption that $i \ge 4$ and $\beta = 1$. Hence Inequality (5.8) holds in this subcase.

Subcase 1.2. *H has at least t extended M-components.*

Due to Lemma 5.4.11, there exist two extended M-components X_1 and X_2 such that there is an M_0'-path P between X_1 and X_2. This path P contains at least two abnormal vertices by Lemma 5.4.13. Let u_1 be the abnormal vertex of P that is closest to X_1 on the path P, and let u_2 be the abnormal vertex of P closest to X_2. Let v_1 and v_2 be the M_0'-neighbors of u_1 and u_2, respectively.

Since $\Delta(H) \leq i-1$ and $|V(M) \setminus \{v_1, v_2\}| = 2j-4$, we have that

$$\sum_{v \in V(M) \setminus \{v_1, v_2\}} d(v) \leq (2j-4)(i-1) = \mathrm{maxm} - 2(i-1).$$

Consequently, we have

$$\sum_{v \in V(M)} d(v) \leq \mathrm{maxm} - 2(i-1) + d(v_1) + d(v_2). \tag{5.10}$$

We will now show that $d(v_1) \leq i - 2\alpha$. By symmetry, this implies that the same inequality holds for $d(v_2)$. Let $A_1 \subseteq M_0'$ be the set of edges of M_0' that are associated with X_1. By Lemma 5.4.14, $|A_1| \geq 1$ and there at at least $t-1$ edges of $M_0' \setminus A_1$ associated with other extended M-components. Recall that M_0'' is the subset of edges of M_0' that are not associated with any extended M-component. We have $|M_0''| \leq |M_0'| - |A_1| - (t-1) \leq |M_0| - |A_1| - (t-1)$, where the last inequality follows from the observation that $|M_0'| \leq |M_0|$.

By Lemma 5.4.15, the only vertices in H to which v_1 can be adjacent are u_1, endpoints of edges in A_1 and endpoints of edges in M_0'', and v_1 cannot be adjacent to both endpoint of any edge in A_1. Hence we have $d(v_1) \leq 1 + |A_1| + 2|M_0''|$. Using the fact that $|M_0''| \leq |M_0| - |A_1| - (t-1)$ and $|M_0| = t + r - 1 - \alpha$, we get $d(v_1) \leq 1 + 2r - 2\alpha - |A_1| \leq 2r - 2\alpha$, where the last inequality follows from $|A_1| \geq 1$. Since we are in the case $i = 2k$, and $r \leq k$ by definition, we obtain $d(v_1) \leq 2k - 2\alpha = i - 2\alpha$.

If we substitute $d(v_1) \leq i - 2\alpha$ and $d(v_2) \leq i - 2\alpha$ in Inequality (5.10), we obtain the following:

$$\begin{aligned} \textstyle\sum_{v \in V(M)} d(v) &\leq \mathrm{maxm} - 2(i-1) + (i-2\alpha) + (i-2\alpha) \\ &= \mathrm{maxm} + 2 - 2\alpha - 2\alpha \\ &\leq \mathrm{maxm} - 2\alpha, \end{aligned}$$

where the last inequality holds since $\alpha \geq 1$. Since $\beta \leq 2\alpha$, we find that Inequality (5.8) holds, which in turn implies that $|E(H)| \leq \rho(i,j)$.

CASE 2. $i = 2k+1$.

The arguments used in this case closely resemble the ones used in the case where $i = 2k$. We now have $\rho(i,j) = i(j-1) - r + 1$, which implies that $2\rho(i,j) \leq \mathrm{maxm} + \mathrm{maxr} - 2(r-1)$. In order to prove Inequality (5.5), it suffices to prove that the following inequality holds:

$$\sum_{v \in V(H)} d(v) \leq \mathrm{maxm} + \mathrm{maxr} - 2(r-1). \tag{5.11}$$

It follows from Inequality (5.4) that Inequality (5.11) holds when $|M_0| \geq r-1$. Suppose $|M_0| = r - 1 - \alpha$ for some $\alpha \geq 1$. If $|M_1| \geq 2\alpha$, then again Inequality (5.4) readily implies that Inequality (5.11) holds. Suppose $|M_1| = 2\alpha - \beta$ for some $1 \leq$

$\beta \le 2\alpha$. Substituting $|M_0| = r-1-\alpha$ and $|M_1| = 2\alpha - \beta$ in Equation (5.3) yields the following equation:

$$\sum_{v \in V(H) \setminus V(M)} d(v) = \text{maxr} - 2(r-1) + \beta\,. \tag{5.12}$$

From Equations (5.1) and (5.12), it follows that in order to satisfy Inequality (5.11), it again suffices to prove the validity of Inequality (5.8).

We first show that, unlike in the case where $i = 2k$, it is not possible that H has at most $t-1$ extended M-components. For contradiction, suppose H has $t-\gamma$ extended M-components for some $\gamma \ge 1$. Since each extended M-component has at least one edge of M_0' associated with it by Lemma 5.4.14, this means that $|M_0| \ge |M_0'| \ge t - \gamma$. On the other hand, we know that $|M_0| = r-1-\alpha$, which yields the following inequality:

$$\alpha \le r-1-t-\gamma. \tag{5.13}$$

Note that every extended M-component X contains at most k edges of M_2, as otherwise the unique M-unsaturated vertex of X would have degree at least $2k+2 = i+1$, contradicting the assumption that $\Delta(H) \le i-1$. Hence $|M_2| \le (t-\gamma)k$. By assumption, $|M_0| + |M_1| = (r-1-\alpha) + (2\alpha - \beta) = r-1+\alpha-\beta$. Since $M_2 = M \setminus (M_0 \cup M_1)$ and $|M| = tk + r - 1$, we have that $|M_2| \ge tk + \beta - \alpha$. Combining this lower bound on $|M_2|$ with the above upper bound implies that $\beta - \alpha \le -\gamma k$. This, together with Inequality (5.13), yields the following:

$$\beta \le (1-k)\gamma - (1-r) - t\,. \tag{5.14}$$

Recall that $r \le k$ and $t \ge 0$. Hence Inequality (5.14) implies that $\beta \le (1-k)(\gamma-1)$. Since $\gamma \ge 1$ and $k \ge 2$, the right-hand side of this inequality is at most 0, contradicting the assumption that $\beta \ge 1$. This contradiction implies that H has at least t extended M-components.

Let X_1 and X_2 be two extended M-components of H such that there is an M_0'-path P between X_1 and X_2; these components exist due to Lemma 5.4.11. This path P contains at least two abnormal vertices by Lemma 5.4.13. Let u_1 be the abnormal vertex of P that is closest to X_1 on the path P, and let v_1 be the M_0'-neighbor of u_1. By Lemma 5.4.15, the only vertices in H to which v_1 can be adjacent are u_1, endpoints of edges in A_1 and endpoints of edges in M_0'', and v_1 cannot be adjacent to both endpoint of any edge in A_1. This implies that $d(v_1) \le 1 + |A_1| + 2|M_0''|$. Since $|M_0''| \le |M_0| - |A_1| - (t-1)$ and $|M_0| = r-1-\alpha$, we get $d(v_1) \le 1 + 2r - 2\alpha - |A_1| - 2t \le 2r - 2\alpha$, where the last inequality follows from the fact that $|A_1| \ge 1$ and $t \ge 0$. Since we assumed that $i = 2k+1$, and we know that $r \le k$, we find that $d(v_1) \le 2k - 2\alpha = (i-1) - 2\alpha$.

Recall that $\Delta(H) \le i-1$ and $|M| = j-1$, so $|V(M) \setminus \{v_1\}| = 2j-3$. Hence we have that $\sum_{v \in V(M) \setminus \{v_1\}} d(v) \le \text{maxm} - (i-1)$, which implies that

$$\sum_{v \in V(M)} d(v) = \text{maxm} - (i-1) + d(v_1)\,. \tag{5.15}$$

Equation (5.15), together with $d(v_1) \le (i-1) - 2\alpha$ and $\beta \le 2\alpha$, yields Inequality (5.8), which in turn implies that $|E(H)| \le \rho(i,j)$.

We have shown that for any pair of integers $i \geq 4$ and $j \geq 1$, every graph H with $\Delta(H) < i$ and $\nu(H) < j$ has at most $\rho(i,j)$ edges. It remains to show that for every $i \geq 4$ and $j \geq 1$, there exists a graph H with $\Delta(H) < i$ and $\nu(H) < j$ that has exactly $\rho(i,j)$ edges.

Let $i \geq 4$ and $j \geq 1$ be two integers. First suppose $i = 2k+1$. Let $H_{i,j}$ be the graph that is the disjoint union of t copies of K_i and $r-1$ copies of $K_{1,i-1}$, where $t \geq 0$ and $1 \leq r \leq k$ are such that $j = tk + r$; see Figure 5.3 for an illustration. It is clear that $\Delta(H_{i,j}) = i-1 < i$. Every matching in $H_{i,j}$ contains at most k edges of each K_i and at most one edge of each $K_{1,i-1}$, which implies that $\nu(H_{i,j}) \leq tk + r - 1 = j - 1 < j$. Since K_i contains $i(i-1)/2 = ki$ edges and $K_{1,i-1}$ contains $i-1$ edges, we have that $|E(H)| = tki + (r-1)(i-1) = i(tk+r-1) - r + 1 = i(j-1) - r + 1$, where the last equality follows from the fact that $j = tk + r$. Hence $|E(H)| = \rho(i,j)$ in this case.

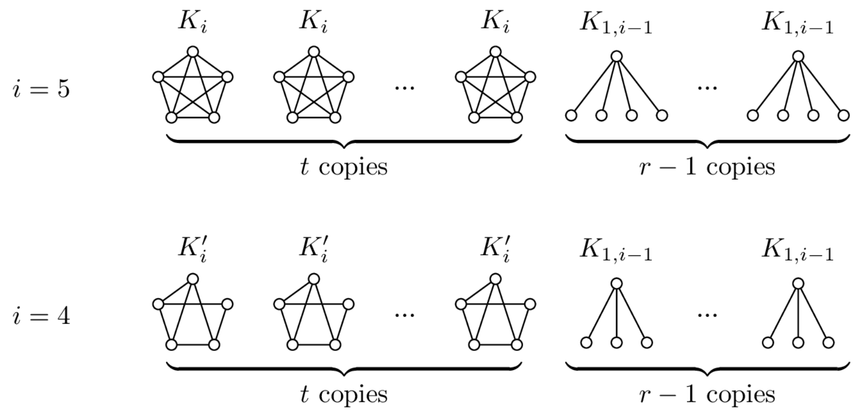


Figure 5.3: A schematic illustration of the graph $H_{i,j}$ in the case where $i = 5$ (top) and $i = 4$ (bottom).

Now suppose $i = 2k$. Let K'_i be the graph obtained from the complete graph K_i by deleting the edges of a perfect matching in K_i, and adding one new vertex that is made adjacent to exactly $i-1$ vertices in the remaining graph. We define $H_{i,j}$ to be the disjoint union of t copies of K'_i and $r-1$ copies of $K_{1,i-1}$; see Figure 5.3 for an illustration. Again, it is clear that $\Delta(H) = i-1 < i$. Every matching in H contains at most k edges of each K'_i and at most one edge of each $K_{1,i-1}$. Therefore, each matching in H has at most $tk + r - 1 = j - 1$ edges, so $\nu(H) < j$. It remains to determine the number of edges in H. The graph K'_i contains $ki - 1$ edges. Hence $|E(H)| = t(ki-1) + (r-1)(i-1) = i(tk+r-1) - t - r + 1$. Since $j = tk + r$, the number of edges in H equals $\rho(i,j)$ also in this case. This concludes the proof of Theorem **??**. $\square$

Chapter 6

Conclusion and future work

Study of graph problems on graph classes, regardless of their hardness, forms an important area of graph theory. Investigating in this direction becomes even more attractive and popular when a hard problem is solved easily for some classes of graphs. In Part I of this thesis, we provided a brief survey of graph classes, algorithmic graph problems, and their complexities to the extend that is needed for Part II. We studied briefly the techniques to cope with intractable algorithmic graph problems emphasizing the importance of restricting the instances to graph classes. We also surveyed extremal graph theory, Ramsey theory and the famous combinatorial problem of determining the Ramsey numbers of graphs. We showed the inherent hardness of this problem and discussed the advantages of defining the problem for graph classes.

In the following, related to the contents of Part I, we suggest some directions that might lead for further relevant works.

Enumeration and maximum number of objects

For some graph objects, like maximal independent sets, there exist tight upper bounds on their number on general graphs. For some of these objects, better upper bounds have been obtained on some classes of graphs. In paper V of Part II [4], we show that the number of maximal 1-regular induced subgraphs is tightly upper bounded by $3^{n/3}$ for triangle-free graphs and it is much smaller than the bound $10^{n/5}$ for general graphs.

• The maximum number of maximal 2-regular induced subgraphs of general graphs is upper bounded by $35^{n/7}$. What is the maximum number of these objects on triangle-free graphs?

There are objects for which the best known upper bounds are not tight. We have seen some of these objects in Section 4.2.3. For any of these objects improving the existent bounds yields a narrower gap. Here we point out some further works related to these objects.

• The best known upper and lower bounds for the maximum number of minimal dominating sets on general graphs are respectively, 1.7697^n and 1.5704^n. What is the tight bound on general graphs?

• The best known upper and lower bounds for the maximum number of minimal dominating sets on chordal graphs are respectively, 1.6181^n and 1.4422^n. What is the tight bound on chordal graphs?

• The best known upper bound for the maximum number of maximal induced bipartite subgraphs on general graphs is $O(1.8613^n)$ and the corresponding best known lower bound is 1.5926^n. What is the tight bound on general graphs? Are there better bounds on graph classes?

• For the maximum number of minimal feedback vertex sets in general graphs, the best known upper and lower bounds are respectively, 1.8638^n and 1.5926^n. What is the tight bound on general graphs?

• There is a large gap between the best known upper and lower bounds, 1.8638^n and 1.5926^n, for the maximum number of minimal subset feedback vertex sets on general graphs. What is the tight bound?

• For the maximum number of minimal subset feedback vertex sets on chordal graphs, there is a gap between the best known upper bound 1.6708^n and the best known lower bound 1.5848^n. What is the tight bound on chordal graphs?

Extremal graph theory and Ramsey theory

For the classical Ramsey numbers, there were few earlier results on graph classes and our contribution will be presented in Paper VI [5] of Part II. Finding more classes of graphs for which we can formulate their Ramsey numbers will be a nice continuation of our work.

• The formula for the classical Ramsey numbers of line graphs is known. What are the Ramsey numbers of quasi-line graphs which form a superclass of line graphs?

Even bigger superclass of line graphs is the class of claw-free graphs. We have seen that $R_{\mathcal{C}}(i,3) = R(i,3)$, where $\mathcal{C}$ is the class of claw-free graphs. This is an indication that it is highly unlikely to manage formulating all the Ramsey numbers of claw-free graphs easily.

• What are the classical Ramsey numbers of claw-free graphs when $i \geq 1$ and $j \geq 4$?

We can ask the same question for AT-free graphs and P_l-free graphs where $l \geq 5$.

• Let us consider the other types of Ramsey numbers. All the Ramsey numbers of trees and cliques for general graphs have been determined. Are there other interesting types of Ramsey numbers for which all the Ramsey numbers of general graphs can be determined?

• We saw that the Ramsey numbers of cycles and cliques are also hard to determine and for claw-free graphs a few of these numbers have been found. Can we determine all the Ramsey numbers of cycles and cliques for some classes of graphs?

Bibliography

[1] Agrawal, R., Imieliński, T., and Swami, A. Mining association rules between sets of items in large databases. *SIGMOD Rec. 22*, 2 (June 1993), 207–216, doi: 10.1145/170036.170072. 4

[2] Appel, K., and Haken, W. Every planar map is four-colorable. *American Mathematical Society* (1989). 5.4.1

[3] Balachandran, N., and Khare, N. Graphs with restricted valency and matching number. *Discrete Mathematics 309*, 12 (2009), 4176 – 4180, doi: http://dx.doi.org/10.1016/j.disc.2008.10.007. 5.4.5, 5.4.3

[4] Basavaraju, M., Heggernes, P., van 't Hof, P., Saei, R., and Villanger, Y. Maximal induced matchings in triangle-free graphs. In *Graph-Theoretic Concepts in Computer Science*, D. Kratsch and I. Todinca, Eds., vol. 8747 of *Lecture Notes in Computer Science*. Springer International Publishing, 2014, pp. 93–104. 1, 4.2.2, 6

[5] Belmonte, R., Heggernes, P., van 't Hof, P., Rafiey, A., and Saei, R. Graph classes and Ramsey numbers. *Discrete Applied Mathematics 173* (2014), 16 – 27, doi: 10.1016/j.dam.2014.03.016. 1, 5.4.3, 6

[6] Belmonte, R., Heggernes, P., van 't Hof, P., and Saei, R. Ramsey numbers for line graphs and perfect graphs. In *Computing and Combinatorics*, J. Gudmundsson, J. Mestre, and T. Viglas, Eds., vol. 7434 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 204–215. 5.4.5, 5.4.3

[7] Berge, C. Färbung von graphen, deren sämtliche bzw. deren ungerade kreise starr sind (zusammenfassung). *Math-Natur. Reihe, Wizz. Z. Martin-Luther-Univ. Halle-Wittenberg* (1961), 114 – 115. 1

[8] Bertossi, A. A. Dominating sets for split and bipartite graphs. *Information Processing Letters 19*, 1 (1984), 37 – 40, doi: http://dx.doi.org/10.1016/0020-0190(84)90126-1. 3.2, 3.2.2, 3.2.3

[9] Bielak, H. A note on the Ramsey number and the planar Ramsey number for C_4 and complete graphs. *Discuss. Math. Graph Theory 19*, 2 (1999), 135 – 142. 5.3, 5.3.10, 5.4.2, 5.4.3

[10] Bielak, H., and Gorgol, I. The planar Ramsey number for C_4 and K_5 is 13. *Discrete Mathematics 236*, 13 (2001), 43 – 51, doi: http://dx.doi.org/10.1016/S0012-365X(00)00429-5. Graph Theory. 5.4.2

[11] BODLAENDER, H. L., DOWNEY, R. G., FELLOWS, M. R., AND HERMELIN, D. On problems without polynomial kernels. *Journal of Computer and System Sciences 75*, 8 (2009), 423 – 434, doi: http://dx.doi.org/10.1016/j.jcss.2009.04.001. 3.4

[12] BOLLOBÁS, B. Extremal problems. In *Modern Graph Theory*, vol. 184 of *Graduate Texts in Mathematics*. Springer New York, 1998, pp. 103–144. 5.1

[13] BONDY, J. A., AND ERDŐS, P. Ramsey numbers for cycles in graphs. *Journal of Combinatorial Theory, Series B 14*, 1 (1973), 46 – 54, doi: http://dx.doi.org/10.1016/S0095-8956(73)80005-X. 5.3

[14] BRANDSTÄDT, A., LE, V. B., AND SPINRAD, J. P. *Graph Classes: A Survey*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999. 2.2

[15] BURR, S. A., ERDŐS, P., FAUDREE, R. J., P., ROUSSEAU, C. C., AND SCHELP, R. H. Ramsey numbers for the pair sparse graph-path or cycle. *Transactions of The American Mathematical Society 269*, 2 (1982), 501 – 512. 5.3

[16] BURR, S. A., ERDŐS, P., AND SPENCER, J. H. Ramsey theorems for multiple copies of graphs. *Transactions of The American Mathematical Society 209* (1975), 87 – 99. 5.3

[17] BUSS, S. R. The boolean formula value problem is in ALOGTIME. In *in Proceedings of the 19-th Annual ACM Symposium on Theory of Computing* (1987), pp. 123–131. 10

[18] BYSKOV, J. Chromatic number in time $O(2.4023^n)$ using maximal independent sets. *B R I C S Report Series*, RS-02-45 (2002). 4.1.1

[19] BYSKOV, J. M., MADSEN, B. A., AND SKJERNAA, B. On the number of maximal bipartite subgraphs of a graph. *Journal of Graph Theory 48*, 2 (2005), 127–132, doi: 10.1002/jgt.20041. 4.2.6, 4.2.7

[20] CAMERON, K. Induced matchings. *Discrete Applied Mathematics 24*, 1–3 (1989), 97 – 102, doi: http://dx.doi.org/10.1016/0166-218X(92)90275-F. 1

[21] CARDOSO, D., KAMIŃSKI, M., AND LOZIN, V. Maximum k-regular induced subgraphs. *Journal of Combinatorial Optimization 14*, 4 (2007), 455–463, doi: 10.1007/s10878-007-9045-9. 4.2.2

[22] CHUDNOVSKY, M., ROBERTSON, N., SEYMOUR, P., AND THOMAS, R. The strong perfect graph theorem. *ANNALS OF MATHEMATICS 164* (2006), 51–229. 2.2

[23] CHUDNOVSKY, M., AND SEYMOUR, P. The structure of claw-free graphs. In *PROCEEDINGS OF THE BRITISH COMBINATORIAL CONFERENCE* (2005), Cambridge University Press, pp. 153–172. 2.2

[24] CHUNG, F., AND GRAHAM, R. *Erdős on Graphs: His Legacy of Unsolved Problems*. Ak Peters Series. Taylor & Francis, 1998. 5.1

[25] CHVÁTAL, V. Tree-complete graph Ramsey numbers. *Journal of Graph Theory 1* (1977), 93. 5.3.6, 5.3.2

[26] CHVÁTAL, V., AND HANSON, . D. Degrees and matchings. *Journal of Combinatorial Theory, Series B 20*, 2 (1976), 128 – 138, doi: http://dx.doi.org/10.1016/0095-8956(76)90004-6. 5.4.5, 5.4.3

[27] CHVÁTAL, V., AND HARARY, F. Generalized Ramsey theory for graphs. III. small off-diagonal numbers. *Pacific Journal of Mathematics 41*, 2 (1972), 335 – 345. 5.3.9

[28] CLANCY, M. Some small Ramsey numbers. *J. Graph Theory 1* (1977), 89 – 91. 5.3.9

[29] CLARK, L. On cycle-star graph Ramsey numbers. *Congressus Numerantium 50* (1985), 187 – 192. 5.3

[30] COCKAYNE, E. J., AND LORIMER, P. J. On Ramsey graph numbers for stars and stripes. *Canad. Math. Bull. 18*, 1 (1975), 31 – 34. 5.3

[31] CONLON, D. Graph Ramsey theory. [Online; accessed 17-November-2014], `https://www.dpmms.cam.ac.uk/~dc340/RamseyLecture1.pdf`. 5.3.1, 5.3.2

[32] COUTURIER, J., HEGGERNES, P., VAN 'T HOF, P., AND KRATSCH, D. Minimal dominating sets in graph classes: Combinatorial bounds and enumeration. *Theor. Comput. Sci. 487* (2013), 82–94, doi: 10.1016/j.tcs.2013.03.026. 4.2.3, 4.1

[33] COUTURIER, J., AND LIEDLOFF, M. A tight bound on the number of minimal dominating sets in split graph. In *12th Cologne-Twente Workshop on Graphs and Combinatorial Optimization, Enschede, Netherlands, May 21-23, 2013.* (2013), pp. 67–70. 4.2.3, 4.1

[34] COUTURIER, J.-F., HEGGERNES, P., VAN 'T HOF, P., AND VILLANGER, Y. Maximum number of minimal feedback vertex sets in chordal graphs and cographs. In *Computing and Combinatorics*, J. Gudmundsson, J. Mestre, and T. Viglas, Eds., vol. 7434 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 133–144. 4.2.3, 4.2.9, 4.2.10

[35] DAILEY, D. P. Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete. *Discrete Mathematics 30*, 3 (1980), 289 – 293, doi: http://dx.doi.org/10.1016/0012-365X(80)90236-8. 2.3.4

[36] DICKSON, L. E. *Fermat's Last Theorem, $ax^r + by^s = cz^t$, and the congruence $x^n + y^n \equiv z^n (\pmod P)$*. Dover, 2005, pp. 731 – 776. 5.2

[37] DIEHL, C., JARD, C., AND RAMPON, J.-X. Reachability analysis on distributed executions. In *TAPSOFT'93: Theory and Practice of Software Development*, M.-C. Gaudel and J.-P. Jouannaud, Eds., vol. 668 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1993, pp. 629–643. 4

[38] DIRAC, G. A. A property of 4-chromatic graphs and some remarks on critical graphs. *Journal of The London Mathematical Society-second Series s1-27* (1952), 85–92, doi: 10.1112/jlms/s1-27.1.85. 5.3

[39] DOWNEY, R. R. G., AND FELLOWS, M. *Parameterized Complexity*. Monographs in Computer Science. Springer Verlag, 1999. 3.4.1, 3.4, 3.4

[40] EPPSTEIN, D. Small maximal independent sets and faster exact graph coloring. In *Algorithms and Data Structures*, F. Dehne, J.-R. Sack, and R. Tamassia, Eds., vol. 2125 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2001, pp. 462–470. 4.1.1

[41] EPSTEIN, L., LEVIN, A., AND WOEGINGER, G. The (weighted) metric dimension of graphs: Hard and easy cases. In *Graph-Theoretic Concepts in Computer Science*, M. Golumbic, M. Stern, A. Levy, and G. Morgenstern, Eds., vol. 7551 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 114–125. 2.3.4, 3.2

[42] ESCOFFIER, B., GOURVÈS, L., AND MONNOT, J. Complexity and approximation results for the connected vertex cover problem in graphs and hypergraphs. *Journal of Discrete Algorithms 8*, 1 (2010), 36 – 49, doi: http://dx.doi.org/10.1016/j.jda.2009.01.005. 3.2

[43] EXOO, G. Ramsey numbers, 2014. [Online; accessed 12-November-2014], `http://ginger.indstate.edu/ge/RAMSEY/`. 5.1

[44] FAUDREE, R., LAWRENCE, S. L., PARSONS, T. D., AND SCHELP, R. H. Path-cycle Ramsey numbers. *Discrete Mathematics 10*, 2 (1974), 269 – 277, doi: http://dx.doi.org/10.1016/0012-365X(74)90122-8. 5.3.8

[45] FAUDREE, R. J., AND SCHELP, R. H. All Ramsey numbers for cycles in graphs. *Discrete Mathematics 8*, 4 (1974), 313 – 329, doi: http://dx.doi.org/10.1016/0012-365X(74)90151-4. 5.3.7

[46] FAUDREE, R. J., SCHELP, R. H., AND SIMONOVITS, M. On some Ramsey type problems connected with paths, cycles and trees. *Ars Combinatoria 29A* (1990), 97 – 106. 5.3

[47] FERNAU, H., HEGGERNES, P., MEISTER, D., AND SAEI, R. Covering the edges of bipartite graphs by overlapping C_4's. Submitted. 1

[48] FERNAU, H., HEGGERNES, P., VAN 'T HOF, P., MEISTER, D., AND SAEI, R. Computing the metric dimension for chain graphs. Submitted. 1, 3.2

[49] FOMIN, F., GASPERS, S., PYATKIN, A., AND RAZGON, I. On the minimum feedback vertex set problem: Exact and enumeration algorithms. *Algorithmica 52*, 2 (2008), 293–307, doi: 10.1007/s00453-007-9152-0. 4.2.3, 4.2.8

[50] FOMIN, F., HEGGERNES, P., KRATSCH, D., PAPADOPOULOS, C., AND VILLANGER, Y. Enumerating minimal subset feedback vertex sets. *Algorithmica 69*, 1 (2014), 216–231, doi: 10.1007/s00453-012-9731-6. 1, 4.2.3

[51] Fomin, F. V., Grandoni, F., Pyatkin, A. V., and Stepanov, A. A. On maximum number of minimal dominating sets in graphs. *Electronic Notes in Discrete Mathematics 22* (2005), 157–162, doi: 10.1016/j.endm.2005.06.028. 4

[52] Fomin, F. V., Grandoni, F., Pyatkin, A. V., and Stepanov, A. A. Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications. *ACM Trans. Algorithms 5*, 1 (Dec. 2008), 9:1–9:17, doi: 10.1145/1435375.1435384. 4.2.3, 4.2.5, 4.2.3, 4.1

[53] Fomin, F. V., and Kratsch, D. *Exact Exponential Algorithms*, 1st ed. Springer-Verlag New York, Inc., New York, NY, USA, 2010. 3, 3.3, 3.3.1, 3.3.2, 3.3, 4, 2

[54] Frank, A. Some polynomial algorithms for certain graphs and hypergraphs. In *British Combinatorial Conference* (1975). 3.2

[55] Fraughnaugh, K. L., and Locke, S. C. Lower bounds on size and independence in K_4-free graphs. *Journal of graph theory 26* (1997), 61 – 71. 5.4.1

[56] Fraughnaugh Jones, K. Independence in graphs with maximum degree four. *Journal of comb. theory ser. B 37* (1984), 254 – 269. 5.4.1

[57] Fulkerson, D. R., and Gross, O. A. Incidence matrices and interval graphs. *Pacific J. Math. 15*, 3 (1965), 835–855. 2.2

[58] Garey, M., Johnson, D., and Stockmeyer, L. Some simplified NP-complete graph problems. *Theoretical Computer Science 1*, 3 (1976), 237 – 267, doi: http://dx.doi.org/10.1016/0304-3975(76)90059-1. 2.3.4

[59] Garey, M. R., and Johnson, D. S. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990. 2.3.4, 3.2

[60] Goldberg, L. A. *Efficient Algorithms for Listing Combinatorial Structures*. Cambridge University Press, 1993. Cambridge Books Online. 24

[61] Golovach, P., Heggernes, P., Kanté, M., Kratsch, D., and Villanger, Y. Minimal dominating sets in interval graphs and trees. Submitted. 4.2.3, 4.1

[62] Golovach, P. A., Heggernes, P., Kratsch, D., and Saei, R. Subset feedback vertex sets in chordal graphs. *J. Discrete Algorithms 26* (2014), 7–15, doi: 10.1016/j.jda.2013.09.005. 1, 4.2.3

[63] Golovach, P. A., Heggernes, P., Kratsch, D., and Villanger, Y. An incremental polynomial time algorithm to enumerate all minimal edge dominating sets. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I* (2013), pp. 485–496. 4.1.1, 4.1.4

[64] Gould, R. *Graph Theory*. Dover Books on Mathematics Series. Dover Publications, Incorporated, 2012. 5.1

[65] Graham, R. L., and Rothschild, B. L. Ramsey's theorem for n-dimensional arrays. *Bulletin of the American Mathematical Society 75*, 2 (03 1969), 418 – 422. 5.2

[66] Graham, R. L., Rothschild, B. L., and Spencer, J. H. *Ramsey Theory*. A Wiley-Interscience publication. Wiley, 1990. 5.2

[67] Greenwood, R. E., and Gleason, A. M. Combinatorial relations and chromatic graphs. *Canadian Journal of Mathematics 7* (1955), 1 – 7. 5.3.4, 5.3

[68] Gunopulos, D., Mannila, H., Khardon, R., and Toivonen, H. Data mining, hypergraph transversals, and machine learning (extended abstract). In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (New York, NY, USA, 1997), PODS '97, ACM, pp. 209–216. 4

[69] Gupta, S., Raman, V., and Saurabh, S. Maximum \$r\$-regular induced subgraph problem: Fast exponential algorithms and combinatorial bounds. *SIAM Journal on Discrete Mathematics 26*, 4 (2012), 1758–1780, doi: 10.1137/09077850X. 1, 4.2.2, 4.2.3, 4.2.4

[70] Hansen, P. J., and Jurs, P. C. Chemical applications of graph theory. part II. Isomer enumeration. *Journal of Chemical Education 65*, 8 (1988), 661, doi: 10.1021/ed065p661. 4

[71] Heggernes, P., van 't Hof, P., van Leeuwen, E., and Saei, R. Finding disjoint paths in split graphs. *Theory of Computing Systems* (2014), 1–20, doi: 10.1007/s00224-014-9580-6. 1

[72] Hujtera, M., and Tuza, Z. The number of maximal independent sets in triangle-free graphs. *SIAM Journal on Discrete Mathematics 6*, 2 (1993), 284–288, doi: 10.1137/0406022. 4.2.1, 4.2.2

[73] Johnson, D. S., Yannakakis, M., and Papadimitriou, C. H. On generating all maximal independent sets. *Information Processing Letters 27*, 3 (1988), 119 – 123, doi: http://dx.doi.org/10.1016/0020-0190(88)90065-8. 1, 22, 4.1.1, 4.1.5

[74] Karp, R. M. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R. Miller, J. Thatcher, and J. Bohlinger, Eds., The IBM Research Symposia Series. Springer US, 1972, pp. 85–103. 2.3.4

[75] König, D. Graphok és matrixok. *Mat. Fiz. Lapok 38* (1931), 116–119. 2.2

[76] Kuratowski, C. Sur le problème des courbes gauches en topologie. *Fundamenta Mathematicae 15*, 1 (1930), 271–283. 2.2

[77] Landman, B. M., and Robertson, A. *Ramsey Theory on the Integers*. Student mathematical library. American Mathematical Society, 2004. 5.2, 5.2.1

[78] LAWLER, E. L. A Note on the Complexity of the Chromatic Number Problem. *Information Processing Letters 5* (1976), 66–67, doi: 10.1016/0020-0190(76)90065-X. 4.1.1

[79] LAWLER, E. L., LENSTRA, J. K., AND KAN, A. H. G. R. Generating all maximal independent sets: Np-hardness and polynomial-time algorithms. *SIAM Journal on Computing 9*, 3 (1980), 558–565, doi: 10.1137/0209042. 21, 4.1.1, 4.1.1, 4.1.2, 4.1.3

[80] LI, D. Y., AND WANG, Z. J. The Ramsey number $r(mC_4, nC_4)$ (in chinese). *Journal of Shanghai Tiedao University 20* (1999), 66 – 70. 5.3

[81] LOVÁSZ, L. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B 13*, 2 (1972), 95 – 98, doi: http://dx.doi.org/10.1016/0095-8956(72)90045-7. 2.2

[82] MANIN, Y. I., AND PANCHISHKIN, A. A. *Introduction to Modern Number Theory: Fundamental Problems, Ideas and Theories*, 2nd ed. Springer Publishing Company, Incorporated, 2007. 5.2

[83] MANTEL, W. Problem 28. *Wiskundige Oppgaven 10* (1907), 60 – 61. 5.1.2

[84] MARINO, A. Algorithms for biological graphs: Analysis and enumeration. *Bulletin of the EATCS 114* (2014). 4

[85] MATTHEWS, M. Extremal results for $K_{1,3}$-free graphs. *Congressus Numerantium 49* (1985), 49 – 55. 5.4.1, 5.4.4

[86] MISRA, N., RAMAN, V., AND SAURABH, S. Lower bounds on kernelization. *Discrete Optimization 8*, 1 (2011), 110 – 128, doi: http://dx.doi.org/10.1016/j.disopt.2010.10.001. Parameterized Complexity of Discrete Optimization. 3.4

[87] MOON, J., AND MOSER, L. On cliques in graphs. *Israel Journal of Mathematics 3*, 1 (1965), 23–28, doi: 10.1007/BF02760024. 1, 4, 4.2.1

[88] PAULL, M., AND UNGER, S. Minimizing the number of states in incompletely specified sequential switching functions. *Electronic Computers, IRE Transactions on EC-8*, 3 (Sept 1959), 356–367, doi: 10.1109/TEC.1959.5222697. 4.1.1, 4.1.1

[89] PLUMMER, D., AND LOVÁSZ, L. *Matching Theory*. North-Holland Mathematics Studies. Elsevier Science, 1986. 5.4.7

[90] RADZISZOWSKI, S. Ramsey numbers involving cycles. 5.3.7, 5.3, 5.3

[91] RADZISZOWSKI, S. Small Ramsey numbers. *The Electronic Journal of Combinatorics 1994-2014* (1994), 1 – 94. 5.3, 5.1

[92] RADZISZOWSKI, S., AND TSE, K. K. A computational approach for the Ramsey numbers $r(C_4, K_n)$. *Journal of Combinatorial Mathematics and Combinatorial Computing 42* (2002), 195 – 207. 5.3

[93] ROSTA, V. On a Ramsey-type problem of J. A. Bondy and P. Erdös. II. *Journal of Combinatorial Theory, Series B 15*, 1 (1973), 105 – 120, doi: http://dx.doi.org/10.1016/0095-8956(73)90036-1. 5.3.7

[94] ROUSSEAU, C. C., AND JAYAWARDENE, C. J. The Ramsey number for a quadrilateral vs.a complete graph on six vertices. *Congressus Numerantium 123* (1997), 97 – 108. 5.3.9

[95] SCHUR, I. Über die kongruenz $x^m + y^m \equiv z^m (\pmod p)$. *Jahresber. Deutsche Math 25* (1916), 114 – 116. 5.2.3

[96] SIPSER, M. *Introduction to the Theory of Computation*, 3rd ed. Cengage learning, 2013. 2.3.1, 2.3.1, 2.3.4

[97] SIPSER, M., AND HARTMANIS, J. The history and status of the P versus NP question, 1992. 4

[98] SOIFER, A. *Ramsey Theory: Yesterday, Today, and Tomorrow*. Progress in Mathematics. Birkhäuser, 2010. 5.1, 5.2, 5.3.5

[99] SPENCER, J. H. Ramsey's theorem for spaces. *Transactions of The American Mathematical Society 249* (1979), 363 – 371. 5.2

[100] STATON, M. Some Ramsey-type numbers and the independence ratio. *Trans. Amer. Math. Soc. 256* (1979), 353 – 370. 5.4.1

[101] STEINBERG, R., AND TOVEY, C. A. Planar Ramsey numbers. *Journal of Combinatorial Theory, Series B 59*, 2 (1993), 288 – 296, doi: http://dx.doi.org/10.1006/jctb.1993.1070. 5.4.1, 5.4.1

[102] SURAHMAT, BASKORO, E., UTTUNGGADEWA, S., AND BROERSMA, H. An upper bound for the Ramsey number of a cycle of length four versus wheels. In *Combinatorial Geometry and Graph Theory*, J. Akiyama, E. Baskoro, and M. Kano, Eds., vol. 3330 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 181 – 184. 5.3

[103] TSUKIYAMA, S., IDE, M., ARIYOSHI, H., AND SHIRAKAWA, I. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing 6*, 3 (1977), 505–517, doi: 10.1137/0206036. 1, 4.1.1

[104] TURÁN, P. Eine extremalaufgabe aus der graphentheorie. *Mat. Fiz Lapook 48* (1941), 436 – 452. 5.1.1

[105] WALKER, K. The analogue of Ramsey numbers for planar graphs. *Bull. London Math. Soc. 1* (1969), 187 – 190. 5.4.1, 5.4.1

[106] WHITNEY, H. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics 54*, 1 (1932), pp. 150–168. 2.2

[107] WIKIPEDIA. Four color theorem — wikipedia, the free encyclopedia, 2014. [Online; accessed 15-November-2014], `http://en.wikipedia.org/w/index.php?title=Four_color_theorem&oldid=633835990`. 5.4.1

[108] WIKIPEDIA. Frank P. Ramsey — wikipedia, the free encyclopedia, 2014. [Online; accessed 14-November-2014], `http://en.wikipedia.org/w/index.php?title=Frank_P._Ramsey&oldid=623236196`. 5.2

[109] WILLIAMS, R. Improving exhaustive search implies superpolynomial lower bounds. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing* (New York, NY, USA, 2010), STOC '10, ACM, pp. 231–240. 4

[110] WILLIAMSON, D. P., AND SHMOYS, D. B. *The Design of Approximation Algorithms*, 1st ed. Cambridge University Press, New York, NY, USA, 2011. 11

[111] YONGQI, S., YUANSHENG, Y., XIAOHUI, L., AND YANAN, S. Note: The planar Ramsey number $PR(C_4, K_7)$. *Discrete Math. 308*, 23 (Dec. 2008), 5841–5848, doi: 10.1016/j.disc.2007.09.058. 5.4.2

Part II
Scientific results

Chapter 7

Computing the metric dimension for chain graphs

Henning Fernau, Pinar Heggernes, Pim van 't Hof, Daniel Meister, and Reza Saei

Submitted for journal publication

Computing the metric dimension for chain graphs*

Henning Fernau[2], Pinar Heggernes[1], Pim van 't Hof[1], Daniel Meister[2], and Reza Saei[1]

[1] Department of Informatics, University of Bergen, Norway,
{pinar.heggernes,pim.vanthof,reza.saeidinvar}@ii.uib.no
[2] Fachbereich IV – Informatikwissenschaften, Universität Trier, Germany,
{fernau,daniel.meister}@uni-trier.de

Abstract. The metric dimension of a graph G is the smallest size of a set R of vertices that can distinguish each vertex pair of G by the shortest-path distance to some vertex in R. Computing the metric dimension is NP-hard, even when restricting inputs to bipartite graphs. We present a linear-time algorithm for computing the metric dimension for chain graphs, which are bipartite graphs whose vertices can be ordered by neighborhood inclusion.

1 Introduction

Let G be a graph, and let $\mathrm{dist}_G(\cdot,\cdot)$ denote the shortest-path distance of G. A *resolving set* for G is a set $R \subseteq V(G)$ so that each vertex pair u, v of G has a vertex $x \in R$ satisfying $\mathrm{dist}_G(x,u) \neq \mathrm{dist}_G(x,v)$. The metric dimension problem asks for a resolving set of smallest size, and the *metric dimension* of a graph is the size of such a smallest set. Metric dimension was independently defined by Harary and Melter [16] and by Slater [22]. Metric dimension and resolving sets have applications in several technical disciplines, such as chemistry [6], robot navigation [20], network discovery and verification [2], and strategies for the Mastermind game [8]. Metric dimension was also studied for graphs of a high degree of symmetry [1,4,12].

The Metric Dimension decision problem is among the classical NP-complete problems [14]. Already Harary and Melter as well as Slater showed that the metric dimension can be computed efficiently for trees [16,22]. Díaz, Pottonen, Serna, and van Leeuwen stated that Metric Dimension is hard on bounded-degree planar graphs [9]. Epstein, Levin, and Woeginger extended the hardness of Metric Dimension to split graphs, bipartite graphs, co-bipartite graphs, and line graphs of bipartite graphs, and they showed that the metric dimension can be computed efficiently for cycles, cographs, k-edge-augmented trees, and wheels [11]. They even extended their positive results to vertex-weighted graphs, asking for a resolving set of smallest weight instead of smallest size [11]. The metric dimension can be computed in polynomial time for outer-planar graphs [9]. Approximability and inapproximability results were given by Hauptmann, Schmied, and Viehmann [18]. Further hardness results are due to Hartung and Nichterlein [17]. Finally, formulas for metric dimension are known for special graphs like trees and hypercubes [19,21].

We study the metric dimension problem on a class of bipartite graphs. We consider bipartite graphs whose vertices admit an ordering by neighborhood-inclusion. Such graphs are called chain graphs. We show that the metric dimension can be computed in linear time for chain graphs, by evaluating an easy-to-compute function. We also give an exact formula for the metric dimension of the chain graphs without false twins, that is dependent on the number of vertices only.

*This work is supported by the Research Council of Norway and the German Academic Exchange Service.

2 Graph preliminaries, resolving sets, and metric dimension

Our graphs are simple, finite, undirected. Let $G = (V, E)$ be a graph. The set of neighbors of a vertex v of G is denoted by $N_G(v)$. For a set $S \subseteq V$, $N_G(S) = \bigcup_{v \in S} N_G(v) \setminus S$. We refer the reader to the monograph by Diestel [10] for standard graph terminology and notation that we do not define here.

The *distance* of two vertices u and v of G is denoted by $\mathrm{dist}_G(u, v)$, and it is the length of a shortest u, v-path of G. If G has no u, v-path, then we let $\mathrm{dist}_G(u, v) = |V(G)|$. For a vertex triple u, v, x of G, we say that x *resolves* u and v, or x is a *resolving vertex* for u and v, if $u = v$ or $\mathrm{dist}_G(x, u) \neq \mathrm{dist}_G(x, v)$. We will mostly implicitly assume $u \neq v$ when discussing resolving vertices. A *resolving set* for G is a set R of vertices of G that has a resolving vertex for each vertex pair of G. A *minimum resolving set* for G is a resolving set for G of smallest cardinality. The *metric dimension* of G, denoted as $\dim(G)$, is the size of a minimum resolving set for G.

For every $x \in V$, the *distance relation of* x, denoted as $\sim_x$, is the binary relation on V defined as follows: for every vertex pair u, v of G, $u \sim_x v$ if and only if $\mathrm{dist}_G(x, u) = \mathrm{dist}_G(x, v)$. Note that $\sim_x$ is an equivalence relation on V. Also note that x resolves u and v if and only if u and v belong to different equivalence classes of $\sim_x$, which means that u and v appear in different levels of a breadth-first search of G with source vertex x. By $\mathfrak{Eq}(x)$, we denote the set of the equivalence classes of $\sim_x$. For $X \subseteq V$ where X is non-empty, $\sim_X$ denotes the intersection of the relations $\sim_x$ for $x \in X$, i.e., $\sim_X = \bigcap_{x \in X} \sim_x$, and $\mathfrak{Eq}(X)$ denotes the set of the equivalence classes of $\sim_X$.

A *refinement* of a partition $\mathfrak{C}$ of V is a partition $\mathfrak{D}$ of V such that every member of $\mathfrak{D}$ is contained as a subset in a member of $\mathfrak{C}$. We write $\mathfrak{D} \sqsubseteq \mathfrak{C}$, if $\mathfrak{D}$ is a refinement of $\mathfrak{C}$, or $\mathfrak{D}$ *refines* $\mathfrak{C}$. Observe the two monotonicity properties: $\mathfrak{Eq}(V) \sqsubseteq \mathfrak{Eq}(X) \sqsubseteq \mathfrak{Eq}(Y)$ for $\emptyset \subset Y \subseteq X \subseteq V$, and for $X, Y \subseteq V$ and $Z \subseteq V$, if $\mathfrak{Eq}(X) \sqsubseteq \mathfrak{Eq}(Y)$ then $\mathfrak{Eq}(X \cup Z) \sqsubseteq \mathfrak{Eq}(Y \cup Z)$.

The following lemma shows a straightforward connection between resolving sets and the equivalence classes of the distance relation.

Lemma 1. *Let $G = (V, E)$ be a graph. A set $R \subseteq V$ is a resolving set for G if and only if the members of $\mathfrak{Eq}(R)$ are of size 1 each, which means $\mathfrak{Eq}(V) = \mathfrak{Eq}(R)$.*

Let $G = (V, E)$ be a graph, and let u and v be vertices of G. We call u and v *false twins* of G, or just *twins*, if $N_G(u) = N_G(v)$. Note that false twins are non-adjacent. The false-twin relation is an equivalence relation on V, called the *twin relation*, its equivalence classes are called *twin classes*, and the twin classes can be computed in linear time (see, for instance, [15]).

Observation 1 *Let $G = (V, E)$ be a graph, and let A be a twin class of G. For every $R \subseteq V$, $A \setminus R$ is contained as a subset in a member of $\mathfrak{Eq}(R)$, which means that R cannot resolve the twins in $A \setminus R$. Thus, if R is a resolving set for G, then $|A \setminus R| \leq 1$.*

An *isolated vertex* of G is a vertex without neighbors. The metric dimension of a disconnected graph can be computed from the metric dimension of its connected components, where isolated vertices play a special role.

Proposition 1 ([11]). *Let G be a graph, let $C_1, \ldots, C_t$ be the connected components of G, and let I be the set of isolated vertices of G. Then,*

$$\dim(G) \;=\; \dim(C_1) + \cdots + \dim(C_t) + \begin{cases} 0 & , \text{ if } I = \emptyset \\ |I| - 1 & , \text{ if } I \neq \emptyset. \end{cases}$$

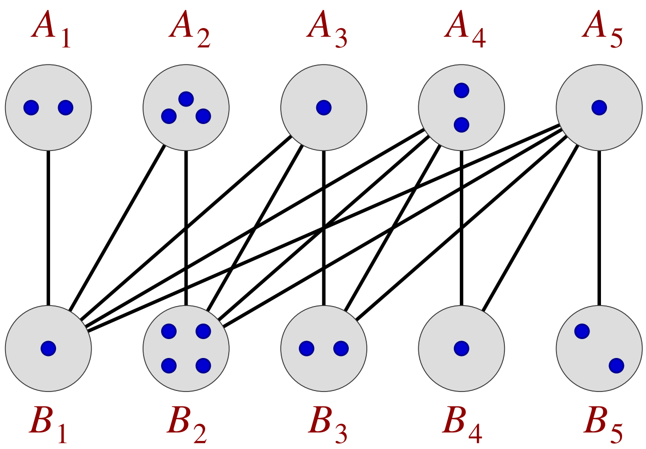


Fig. 1. The figure represents a connected chain graph. The two independent sets A and B are already partitioned into their twin classes, yielding $A_1, \dots, A_5$ for A and $B_1, \dots, B_5$ for B. The vertices in a twin class have the same neighbors. As an example, the vertices in B_3 are adjacent to the vertices in $A_3 \cup A_4 \cup A_5$. The associated cardinality sequences are $\langle 2, 3, 1, 2, 1\rangle$ for A and $\langle 1, 4, 2, 1, 2\rangle$ for B.

In view of Proposition 1, we want to point out that the metric dimension of a graph on a single vertex is 0. Due to Proposition 1, we can restrict ourselves to connected graphs. Furthermore, the metric dimension of a graph on two vertices is 1. *We henceforth assume that every graph in this paper is connected and has at least three vertices.*

In this paper, we consider a class of bipartite graphs. For convenience, we denote a bipartite graph $G = (V, E)$ as (A, B, E) where $\{A, B\}$ defines a partition of V into two independent sets. A *connected chain graph* on at least three vertices is a bipartite graph $G = (A, B, E)$ with its twin classes $A_1, \dots, A_k, B_1, \dots, B_l$ where $A_1 \cup \dots \cup A_k = A$ and $B_1 \cup \dots \cup B_l = B$ such that $\emptyset \subset N_G(A_1) \subset \dots \subset N_G(A_k) = B$ and $A = N_G(B_1) \supset \dots \supset N_G(B_l) \supset \emptyset$. It directly follows that $k = l$ must hold. An example of a connected chain graph with $2 \cdot 5$ twin classes is depicted in Figure 1.

We refer to the monograph by Brandstädt, Le, and Spinrad [3] for the standard definition and a more comprehensive treatment of chain graphs.

3 A special minimum resolving set for chain graphs

Let $G = (A, B, E)$ be a connected chain graph with its twin classes $A_1, \dots, A_k$ and $B_1, \dots, B_k$ that satisfy $N_G(A_1) \subset \dots \subset N_G(A_k)$ and $N_G(B_1) \supset \dots \supset N_G(B_k)$. We combine consecutive twin classes. For $1 \le i \le j \le k$, we denote the union $A_i \cup \dots \cup A_j$ shortly as $A_{i,\dots,j}$, and $B_{i,\dots,j} = B_i \cup \dots \cup B_j$ analogously. We also choose a "representative" of each twin class: choose $a_i \in A_i$ and $b_i \in B_i$ for every $1 \le i \le k$.

As our main combinatorial result, we show that G has a minimum resolving set of a restricted structure. We prove this result by exploiting the structure of chain graphs. The following observations will be important: $N_G(a_p) = B_{1,\dots,p}$ and $N_G(b_p) = A_{p,\dots,k}$ for every $1 \le p \le k$, and $\mathrm{dist}_G(a, a') \in \{0, 2\}$ and $\mathrm{dist}_G(b, b') \in \{0, 2\}$ and $\mathrm{dist}_G(a, b) \in \{1, 3\}$ for $a, a' \in A$ and $b, b' \in B$. Thus, $|\mathfrak{Eq}(x)| \le 4$ for every $x \in A \cup B$.

Lemma 2. *Let R be a resolving set for G, and let $1 \le p \le k$. Assume $B_p \subseteq R$ and $B_{1,\dots,p-1} \not\subseteq R$. Then, there is an index j with $1 \le j \le k$ such that $(R \setminus \{b_p\}) \cup A_j$ is a resolving set for G.*

Proof. Observe

$$\mathfrak{Eq}(b_p) \;=\; \mathfrak{Eq}\big(\{b_p\}\big) = \big\{B \setminus \{b_p\},\; \{b_p\}\big\} \;\cup\; \big\{A_{1,\ldots,p-1},\; A_{p,\ldots,k}\big\}\,.$$

Recall that for $X \subseteq A \cup B$, the vertices in X are in singleton equivalence classes in $\mathfrak{Eq}(X)$. To simplify representations in the proof, we use rest-$\mathfrak{Eq}(X)$ to denote the equivalence classes in $\mathfrak{Eq}(X)$ that do not contain vertices from X. As an example, observe

$$\text{rest-}\mathfrak{Eq}\big(\{b_p\}\big) = \big\{B \setminus \{b_p\},\; A_{1,\ldots,p-1},\; A_{p,\ldots,k}\big\}\,.$$

For the proof, we determine an index j and a subset F of $(R \setminus \{b_p\}) \cup A_j$ such that $\mathfrak{Eq}(F) \sqsubseteq \mathfrak{Eq}(b_p)$. Then, $\mathfrak{Eq}((R \setminus \{b_p\}) \cup A_j) \sqsubseteq \mathfrak{Eq}(R)$ directly follows from the monotonicity properties, and $(R \setminus \{b_p\}) \cup A_j$ is a resolving set for G due to Lemma 1.

We distinguish between two easy and two harder cases. Let $B'_p = B_p \setminus \{b_p\}$, and if $p \geq 2$, let $A'_{p-1} = A_{p-1} \cap R$ and $A''_{p-1} = A_{p-1} \setminus R$. If $p = 1$, then

$$\text{rest-}\mathfrak{Eq}\big(B'_1 \cup \{a_1\}\big) \;=\; \big\{B \setminus B_1, \{b_1\}\big\} \;\cup\; \big\{A \setminus \{a_1\}\big\}\,,$$

and if $p \geq 2$ and $A'_{p-1} \neq \emptyset$, then

$$\text{rest-}\mathfrak{Eq}\big(B'_p \cup A'_{p-1} \cup A_{p,\ldots,k}\big) \;\sqsubseteq\; \big\{B_{1,\ldots,p-1},\; \{b_p\},\; B_{p+1,\ldots,k}\big\} \;\cup\; \big\{A_{1,\ldots,p-2} \cup A''_{p-1}\big\}\,.$$

So, if $p = 1$, then $\mathfrak{Eq}\big(B'_1 \cup \{a_1\}\big) \sqsubseteq \mathfrak{Eq}(b_1)$, and $(R \setminus \{b_1\}) \cup A_1$ is a resolving set for G, and if $p \geq 2$ and $A'_{p-1} \neq \emptyset$ and $A_{p,\ldots,k} \subseteq R \cup A_j$ for some $p \leq j \leq k$, then $(R \setminus \{b_p\}) \cup A_j$ is a resolving set for G. This completes the two easy cases.

We assume that no easy case applies, which means $p \geq 2$, and $A'_{p-1} = \emptyset$ or $A_{p,\ldots,k} \not\subseteq R \cup A_j$ for every $p \leq j \leq k$. Our two harder cases distinguish between $A'_{p-1} \neq \emptyset$ and $A'_{p-1} = \emptyset$.

As the first harder case, assume $A'_{p-1} \neq \emptyset$. Let q with $p \leq q \leq k$ be smallest such that $A_q \not\subseteq R$, and let r with $q < r \leq k$ be smallest such that $A_r \not\subseteq R$; observe that q and r indeed exist. Let $u \in A_q \setminus R$ and $v \in A_r \setminus R$. Since R is a resolving set for G, there is a vertex $z \in R$ resolving u and v. Observe $z \in B_{q+1,\ldots,r}$. Then,

$$\begin{aligned}&\text{rest-}\mathfrak{Eq}\big(B'_p \cup A'_{p-1} \cup A_{p,\ldots,r-1} \cup \{z\}\big)\\ &\sqsubseteq\; \big\{B_{1,\ldots,p-1},\; \{b_p\},\; (B_{p+1,\ldots,k}) \setminus \{z\}\big\} \;\cup\; \big\{A_{1,\ldots,p-2} \cup A''_{p-1},\; A_{r,\ldots,k}\big\}\,,\end{aligned}$$

and $\mathfrak{Eq}\big(B'_p \cup A'_{p-1} \cup A_{p,\ldots,r-1} \cup \{z\}\big) \sqsubseteq \mathfrak{Eq}(b_p)$, and $(R \setminus \{b_p\}) \cup A_q$ is a resolving set for G.

As the second harder case, assume $A'_{p-1} = \emptyset$. This means $A''_{p-1} = A_{p-1}$ by the definition of A''_{p-1} and $A_{p-1} = \{a_{p-1}\}$ due to Observation 1. We consider the vertices in A and in B separately. We begin with the vertices in A. Let s be smallest with $1 \leq s \leq p-1$ such that $A_{s,\ldots,p-1} \subseteq R \cup A_{p-1}$. If $s = 1$, then

$$\text{rest-}\mathfrak{Eq}\big(A_{1,\ldots,p-1}\big) \;\sqsubseteq\; \big\{B_{1,\ldots,p-1},\; B_{p,\ldots,k}\big\} \;\cup\; \big\{A_{p,\ldots,k}\big\}\,.$$

Assume $s \geq 2$. Let $u \in A_{s-1} \setminus R$. Since R is a resolving set for G, there is a resolving vertex z in R for u and a_{p-1}. Observe $z \in B_{s,\ldots,p-1}$. Then,

$$\text{rest-}\mathfrak{Eq}\big(A_{s,\ldots,p-1} \cup \{z\}\big) \;\sqsubseteq\; \big\{(B_{1,\ldots,p-1}) \setminus \{z\},\; B_{p,\ldots,k}\big\} \;\cup\; \big\{A_{1,\ldots,s-1},\; A_{p,\ldots,k}\big\}\,.$$

Observe that each of the two situations refines $\{B_{1,\ldots,p-1}, B_{p,\ldots,k}, A_{1,\ldots,p-1}, A_{p,\ldots,k}\}$ already, and it remains to extract $\{b_p\}$ from $B_{p,\ldots,k}$. We consider the vertices in B. If $A_p \cap R \neq \emptyset$, then

$$\text{rest-}\mathfrak{Eq}\big(B'_p \cup (A_p \cap R)\big) \;\sqsubseteq\; \big\{B_{1,\ldots,p-1} \cup \{b_p\},\; B_{p+1,\ldots,k}\big\} \;\cup\; \big\{A \setminus (A_p \cap R)\big\}\,,$$

and if $B_{p+1,\ldots,k} \subseteq R$, then

$$\text{rest-}\mathfrak{Eq}\big(B'_p \cup B_{p+1,\ldots,k}\big) \;\sqsubseteq\; \big\{B_{1,\ldots,p-1} \cup \{b_p\},\; A\big\}\,.$$

Otherwise, assume $A_p \cap R = \emptyset$ and $B_{p+1,\ldots,k} \not\subseteq R$. Let q with $q > p$ be smallest such that $B_q \not\subseteq R$. Let $v \in B_{p-1} \setminus R$ and $w \in B_q \setminus R$. Recall from the assumptions about p in the lemma that v does indeed exist. Since R is a resolving set for G, there is a resolving vertex z' in R for v and w, and $z' \in A_{p-1,\ldots,q-1}$. However, since $(A_{p-1} \cup A_p) \cap R = \emptyset$, $z' \in A_{p+1,\ldots,q-1}$ follows. So,

$$\text{rest-}\mathfrak{Eq}\big(B'_p \cup B_{p+1,\ldots,q-1} \cup \{z'\}\big) \;\sqsubseteq\; \big\{B_{1,\ldots,p-1} \cup \{b_p\},\; B_{q,\ldots,k},\; A \setminus \{z'\}\big\}\,,$$

and we conclude that $(R \setminus \{b_p\}) \cup A_{p-1}$ is a resolving set for G. $\square$

From Lemma 2, we obtain our main combinatorial result about a minimum resolving set of special structure.

Proposition 2. *G has a minimum resolving set R such that $B_i \not\subseteq R$ for every $1 \leq i \leq k$.*

Proof. Let R be a minimum resolving set for G, and assume that R is chosen such that $p = \min\{i : B_i \subseteq R\} \cup \{k+1\}$ is largest possible. If $p = k+1$, then R satisfies the condition of the claim.

Otherwise, $1 \leq p \leq k$, and we can apply Lemma 2: there is an index j with $1 \leq j \leq k$ such that $(R \setminus \{b_p\}) \cup A_j$ is a resolving set for G. Since $|A_j \setminus R| \leq 1$ due to Observation 1, we observe $\big|(R \setminus \{b_p\}) \cup A_j\big| \leq \big|R\big|$, and thus, $(R \setminus \{b_p\}) \cup A_j$ is a minimum resolving set for G, that contradicts the choice of R. $\square$

4 Computing a minimum resolving set for chain graphs

We want to efficiently compute minimum resolving sets for chain graphs, and thus determine their metric dimension. We could devise a dynamic-programming algorithm based on our main combinatorial result, which is Proposition 2. However, we can do better. We explicitly define a minimum resolving set that also satisfies the structural property of Proposition 2 and explain how it can be computed in linear time.

Before we procede, note the following: for $G = (A, B, E)$ a chain graph, the number of edges of G is of order $\frac{|A|\cdot|B|}{2}$, and thus, linear running time for G is equivalent to a running time of order $(|A|+|B|)^2$. For algorithmic purposes, a more succinct representation is often chosen. In case of chain graphs, the structure of a connected chain graph is completely determined by the two cardinality sequences of the twin classes, and our metric dimension algorithm will have linear running time even on such a succinct input.

We consider a connected chain graph $G = (A, B, E)$ on at least three vertices and with its twin classes $A_1, \ldots, A_k$ and $B_1, \ldots, B_k$ where $N_G(A_1) \subset \cdots \subset N_G(A_k)$ and $N_G(B_1) \supset \cdots \supset N_G(B_k)$. The associated cardinality sequences are $\big\langle |A_1|, \ldots, |A_k| \big\rangle$ and $\big\langle |B_1|, \ldots, |B_k| \big\rangle$.

Definition 1. *Let $1 \leq p \leq q \leq k$. We call $[p, q]$ a* segment *of G if $|B_p| = \cdots = |B_q| = 1$, and $p > 1$ implies $|B_{p-1}| \geq 2$, and $q < k$ implies $|B_{q+1}| \geq 2$.*

We can say that a segment is a maximal interval of consecutive twin classes in B each of size 1. Observe that size-1 twin classes in B play a special role for our minimum resolving sets according to Proposition 2, since a corresponding resolving set would not contain any vertex from such a twin class. Segments force properties of resolving sets, as we show in the next lemma.

Lemma 3. *Let $[p, q]$ be a segment of G, and let s, t be such that $p-1 \leq s < t \leq q$ and $s \geq 1$. Let $Q \subseteq A \cup B$, and assume that $Q \cap (B_p \cup \cdots \cup B_q) = \emptyset$. If Q is a resolving set for G, then $A_s \subseteq Q$ or $A_t \subseteq Q$.*

Proof. Assume $A_s \not\subseteq Q$ and $A_t \not\subseteq Q$. Let $u \in A_s \setminus Q$ and $v \in A_t \setminus Q$. Every resolving vertex for u and v is from $\{u, v\} \cup B_{s+1} \cup \cdots \cup B_t$. Since $Q \cap (\{u, v\} \cup B_{s+1} \cup \cdots \cup B_t) = \emptyset$, Q is not a resolving set for G. □

Definition 2. *Let $1 \leq j \leq k$, and let $[p, q]$ be a segment of G.*

1) *We call j a* high position *if $|A_j| \geq 2$ and $|B_j| \geq 2$ and if $j < k$ then $|B_{j+1}| \geq 2$.*
2) *We call j a* heavy position *in $[p, q]$ if $p - 1 \leq j \leq q$ and $|A_j| \geq 2$.*
3) *We call $[p, q]$* good *if there is a heavy position in $[p, q]$.*

We can say that a high position is like a heavy position that is not in the "range" of a segment.

We now prove a lower bound on the metric dimension of G. To state this bound, we need some more notations.

- Let g be the number of good segments of G, and let h be the number of high positions.
- Let $K = A_k$ if $|A_k| = 1$ and $[p, k]$ is not a good segment of G for every $1 \leq p \leq k$, and let $K = \emptyset$ otherwise.

We clarify these notions for the chain graph of Figure 1: $[1, 1]$ and $[4, 4]$ are segments, and 1 is a heavy position in $[1, 1]$ and 4 is a heavy position in $[4, 4]$, and $K = A_5$. Moreover, 2 is a high position, since $|A_2|, |B_2|, |B_3| \geq 2$, and it is the only high position.

Lemma 4. *The metric dimension of G is at least $|A \cup B| - k - g - h - |K|$.*

Proof. Let R^* be a minimum resolving set for G that satisfies the structural property of Proposition 2. Hence, $|R^* \cap B| = |B| - k$, and $|A_j \setminus R^*| \leq 1$ for every $1 \leq j \leq k$ due to Observation 1, so that $|A| - k \leq |R^* \cap A| \leq |A|$.

Let $L = \{j : 1 \leq j \leq k \text{ and } A_j \not\subseteq R^*\}$, and observe

$$|L| = |A \setminus R^*| = |A \setminus (R^* \cap A)| = |A| - |R^* \cap A|\,.$$

If $|L| \leq g + h + |K|$, then

$$\begin{aligned}\dim(G) &= |R^*| = |R^* \cap A| + |R^* \cap B| \\ &= |A| - |L| + |B| - k \;\geq\; |A| + |B| - k - g - h - |K|\,,\end{aligned}$$

which is the desired lower bound. We verify $|L| \leq g + h + |K|$.

Claim. $|A_j| \geq 2$ for every $j \in L \setminus \{k\}$.

Proof. Let $j \in L \setminus \{k\}$, and consider $u \in B_j \setminus R^*$ and $v \in B_{j+1} \setminus R^*$, that exist according to the assumptions about R^*. If $A_j \cap R^* = \emptyset$, then R^* contains no resolving vertex for u and v, so that $A_j \cap R^* \neq \emptyset$ follows, and $A_j \not\subseteq R^*$ implies $|A_j| \geq 2$. ◊

If $|K| = 0$, then let $L' = L$, and if $|K| = 1$, then let $L' = L \setminus \{k\}$. Observe that $|L| = |L'| + |K|$.

We show $|L'| \leq g + h$, which implies $|L| = |L'| + |K| \leq g + h + |K|$. Let $r \in L'$, and assume that r is not a high position. If $r < k$, then $|A_r| \geq 2$ by the Claim, and $|B_r| = 1$ or $|B_{r+1}| = 1$, and if $r = k$, then $|B_k| = 1$. It follows: G has a segment $[p, q]$ such that $p - 1 \leq r \leq q$. Observe:

$[p, q]$ is a good segment of G, because of $|A_r| \geq 2$ in case when $r < k$, and because of $|K| = 0$ in case when $r = k$.

We show $L' \cap \{p-1, \ldots, q\} = \{r\}$. If $|L' \cap \{p-1, \ldots, q\}| \geq 2$, then there are $s, t \in L'$ such that $p - 1 \leq s < t \leq q$ and $A_s \not\subseteq R^*$ and $A_t \not\subseteq R^*$, and R^* is not a resolving set for G due to Lemma 3, a contradiction. So, $|L' \cap \{p-1, \ldots, q\}| = 1$, and $r \in L' \cap \{p-1, \ldots, q\}$ implies the result. □

We now show the complementary result of Lemma 4, that is an upper bound on the metric dimension of G. We prove the upper bound by explicitly describing a minimum resolving set for G. Arbitrarily choose representatives $a_i \in A_i$ and $b_i \in B_i$ for every $1 \leq i \leq k$. We define the following sets of vertices of G:

$$H = \{a_j : 1 \leq j \leq k \text{ and } j \text{ is a high position}\}$$
$$S = \{a_j : j \text{ is a selected heavy position in a good segment}\}$$
$$R = (A \cup B) \setminus \big(\{b_1, \ldots, b_k\} \cup H \cup S \cup K\big) .$$

Recall that each good segment contains a heavy position. For each good segment, a heavy position is (arbitrarily) selected and the representative is contained in S. Thus, there is a bijection between S and the set of the good segments. For the case of Figure 1: $H = \{a_2\}$ and $S = \{a_1, a_4\}$ and $K = A_5$. The resulting set R would be of size $(9+10) - (5-1-2-1) = 10$.

Proposition 3. *The metric dimension of G is equal to $|A \cup B| - k - g - h - |K|$, and R is a minimum resolving set for G.*

Proof. Clearly, $|R| = |A \cup B| - |\{b_1, \ldots, b_k\}| - |H| - |S| - |K| = |A \cup B| - k - h - g - |K|$. So, if R is a resolving set for G, then R is a minimum resolving set for G due to Lemma 4, and the result follows.

We verify that R is indeed a resolving set for G.

Claim. $R \neq \emptyset$, and $\mathfrak{Eq}(R) \sqsubseteq \{A, B\}$.

Proof. If $R = \emptyset$, then $A \subseteq H \cup S \cup K \subseteq \{a_1, \ldots, a_k\}$ and $B \subseteq \{b_1, \ldots, b_k\}$. The definition of high and heavy positions implies $H \cup S = \emptyset$, so that $A \subseteq K \subseteq \{a_k\}$ follows, and therefore $k = 1$. Since $|A \cup B| \geq 3$ and $B = B_1$, $|B_1| \geq 2$ follows, and $B_1 \not\subseteq \{b_1\}$ yields a contradiction.

The second part of the claim follows from the distance properties of connected bipartite graphs. ◊

As a consequence of the result of the claim, it remains to provide a resolving vertex from R for every vertex pair from $H \cup S \cup K$ and from $\{b_1, \ldots, b_k\}$.

We consider $H \cup S \cup K$. Recall from the definition of R that $|B_j| \geq 2$ implies $B_j \cap R \neq \emptyset$. Let $1 \leq s < t \leq k$, and assume $a_s, a_t \in H \cup S \cup K$. Since $s < k$, $a_s \notin K$, and thus, $a_s \in H \cup S$, and $|A_s| \geq 2$. If $a_t \in H$, then $|B_t| \geq 2$, and each vertex in $R \cap B_t$ resolves a_s and a_t. If $a_t \in S$, then t is a heavy position in a good segment $[p, q]$ of G. Since only one heavy position in $[p, q]$ is selected and $|A_s| \geq 2$, $s \leq p-2$ follows. This means $p \geq 3$ and $|B_{p-1}| \geq 2$, and each vertex in $R \cap B_{p-1}$ resolves a_s and a_t. Finally, assume $a_t \in K$. This means $A_k = \{a_k\}$, and $[p, k]$ for every $1 \leq p \leq k$ is not a good segment of G. If $|B_k| \geq 2$, then each vertex in $R \cap B_k$ resolves a_s and a_k. Otherwise, if $|B_k| = 1$, then there is some p such that $[p, k]$ is a segment of G. Since $[p, k]$ is not a good segment of G, there is no heavy position in $[p, k]$, so that $s < p - 2$ and $|B_{p-1}| \geq 2$ follows, and R has a resolving vertex for a_s and a_k.

We consider $\{b_1, \ldots, b_k\}$. Let $1 \leq s < t \leq k$. If $|A_s| \geq 2$, then R has a vertex in A_s that resolves b_s and b_t. Otherwise, $|A_s| = 1$, i.e., $A_s = \{a_s\}$. Observe that s is not a high position and that s is not a heavy position in any good segment. Moreover, $a_s \notin K$ because of $s < k$. Thus, $a_s \in R$, and a_s resolves b_s and b_t. □

We are ready to present our main result.

Theorem 1. *Given a chain graph G, a minimum resolving set for G can be constructed in linear time. As a consequence, the* Metric Dimension *problem can be solved in linear time on chain graphs.*

Proof. Due to Proposition 1 and the subsequent discussion, it suffices to consider connected chain graphs on at least three vertices, and we can apply Proposition 3.

The twin classes of a given chain graph G, namely $A_1, \ldots, A_k, B_1, \ldots, B_k$, can be computed in linear time [15]. The segments, the high positions, the heavy positions of the segments, and K can be determined in $\mathcal{O}(k)$ time from the cardinality sequences. So, a minimum resolving set R as in Proposition 3 can be constructed in linear time. □

Our second result, an exact formula for the metric dimension of special chain graphs, is a direct consequence of Proposition 3.

Theorem 2. *Let $G = (A, B, E)$ be a connected chain graph on at least four vertices. If each twin class of G is of size 1, then* $\dim(G) = \frac{|A \cup B|}{2} - 1$.

Proof. We apply Proposition 3, and it suffices to observe: $|A \cup B| = 2k$, and $g = h = 0$ and $|K| = 1$, so that $\dim(G) = |A \cup B| - k - g - h - |K| = k - 1$. □

5 Variants of metric dimension

Several variants of metric dimension are considered in the literature. Chartrand, Saenpholphat, and Zhang define the *independent resolving number* [7], which asks for the smallest size of a resolving set that is also an independent set, if such a resolving set exists. Other variants of metric dimension vary the metric itself. As an example, the *adjacency metric dimension* of a graph G, $\dim_A(G)$, asks for a set of vertices of G of smallest size that resolves the vertex pairs of G with respect to $\mathrm{dist}_G^{\leq 2}(x, y) = \min\{\mathrm{dist}_G(x, y), 2\}$; we refer to [13,19] for more information.

The concept of a locating-dominating set combines the two above variations. A set of vertices of a graph G is *locating-dominating* if it is a dominating set of G and an adjacency-resolving set for G at the same time [5,23]. The associated parameter is the *location-domination number*, and it is denoted as γ_L.

Lemma 5. *For every connected graph G,* $\dim(G) \leq \dim_A(G) \leq \gamma_L(G) \leq \dim_A(G) + 1$.

Proof. The first two inequalities, $\dim(G) \leq \dim_A(G) \leq \gamma_L(G)$, are straightforward consequences of the definitions.

For verifying the third inequality, $\gamma_L(G) \leq \dim_A(G) + 1$, let R be an adjacency-resolving set for G. If G has a vertex pair u, v where $u \neq v$ and no neighbor of u and v is contained in R, then no vertex in R adjacency-resolves u, v, which is not possible. So, at most one vertex w of G has no neighbor in R, and R itself or $R \cup \{w\}$ is a dominating set of G and adjacency-resolving for G at the same time. □

The example of induced paths of arbitrary length shows that adjacency metric dimension may be much larger than metric dimension. If we restrict to chain graphs, this large gap is bounded. We verify this bound informally. Let $G = (A, B, E)$ be a connected chain graph and let R be a resolving set for G. If R does not adjacency-resolve a vertex pair u, v, then $\mathrm{dist}_G^{\leq 2}(x, u) = \mathrm{dist}_G^{\leq 2}(x, v)$ for every $x \in R$ but $\mathrm{dist}_G(y, u) \neq \mathrm{dist}_G(y, v)$ for some $y \in R$.

Thus, and by symmetry, we can assume $1 < \mathrm{dist}_G(y,u) < \mathrm{dist}_G(y,v) < 4$, and it follows: $u \in A$ if and only if $v \in B$. It suffices to increment R by a vertex from $B_1 \cup A_k$ (with the meanings as in Figure 1), and $\dim_A(G) \le \dim(G) + 1$ follows.

We reconsider our restriction to connected chain graphs without twins, that admits exact formulas for adjacency metric dimension and the location-domination number.

Corollary 1. *Let $G = (A, B, E)$ be a connected chain graph on at least six vertices whose twin classes are of size 1. Then,* $\dim_A(G) = \frac{|A \cup B|}{2} - 1$ *and* $\gamma_L(G) = \frac{|A \cup B|}{2}$.

Proof. Let $a_1, \ldots, a_k, b_1, \ldots, b_k$ be the representatives of the twin classes of G, as they were chosen at the beginning of Section 3, and where $A = \{a_1, \ldots, a_k\}$ and $B = \{b_1, \ldots, b_k\}$. Note that $\{a_1, \ldots, a_{k-1}\}$ is the minimum resolving set of G considered in Proposition 3. It is an exercise to verify that also $\{a_2, \ldots, a_k\}$ is a resolving set for G, which proves $\dim_A(G) \le k - 1 = \frac{|A \cup B|}{2} - 1$.

For the location-domination number, observe that $\{a_1, \ldots, a_k\}$ is a locating-dominating set for G, proving $\gamma_L(G) \le k$, and a counting argument of the following form proves optimality (recall the notations from Section 3): for R a locating-dominating set for G, if $R \cap A_1 = \emptyset$, then $B_1 \subseteq R$, and for $1 \le s < t \le k$, if $R \cap (A_s \cup A_t) = \emptyset$, then $R \cap B_{s+1,\ldots,t} \neq \emptyset$. $\square$

Acknowledgement

We are grateful to two anonymous reviewers, whose suggestions and detailed comments helped improve the paper.

References

1. R. F. Bailey and P. J. Cameron. Base size, metric dimension and other invariants of groups and graphs. *Bulletin of the London Mathematical Society*, 43:209–242, 2011.
2. Z. Beerliova, F. Eberhard, T. Erlebach, A. Hall, M. Hoffmann, M. Mihalák, and L. Shankar Ram. Network discovery and verification. *IEEE Journal on Selected Areas in Communications*, 24:2168–2181, 2006.
3. A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph classes: a survey.* SIAM, Philadelphia, 1999.
4. J. Cáceres, C. Hernando, M. Mora, I. M. Pelayo, M. L. Puertas, C. Seara, and D. R. Wood. On the Metric Dimension of Cartesian Products of Graphs. *SIAM Journal on Discrete Mathematics*, 21(2):423–441, 2007.
5. I. Charon, O. Hudry, and A. Lobstein. Minimizing the size of an identifying or locating-dominating code in a graph is NP-hard. *Theoretical Computer Science*, 290(3):2109–2120, 2003.
6. G. Chartrand, L. Eroh, M. A. Johnson, and O. R. Oellermann. Resolvability in graphs and the metric dimension of a graph. *Discrete Applied Mathematics*, 105(1–3):99–113, 2000.
7. G. Chartrand, V. Saenpholphat, and P. Zhang. The independent resolving number of a graph. *Mathematica Bohemica*, 128:379–393, 2003.
8. V. Chvátal. Mastermind. *Combinatorica*, 3:325–329, 1983.
9. J. Díaz, O. Pottonen, M. Serna, and E. J. van Leeuwen. On the Complexity of Metric Dimension. *Proceedings of ESA 2012*, Springer LNCS, 7501:419–430, 2012.
10. R. Diestel. *Graph theory.* Springer, Graduate Texts in Mathematics, 173, 2010.
11. L. Epstein, A. Levin, and G. J. Woeginger. The (Weighted) Metric Dimension of Graphs: Hard and Easy Cases. *Proceedings of WG 2012*, Springer LNCS, 7551:114–125, 2012.
12. M. Fehr, S. Gosselin, O. R. Oellermann. The metric dimension of Cayley digraphs. *Discrete Mathematics*, 306:31–41, 2006.
13. H. Fernau and J. A. Rodríguez-Velázquez. Notions of Metric Dimension of Corona Products: Combinatorial and Computational Results. *Proceedings of CSR 2014*, Springer LNCS, 8476:153–166, 2014.

14. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1979.
15. M. Habib, R. M. McConnell, C. Paul, L. Viennot. Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science*, 234:59–84, 2000.
16. F. Harary and R. A. Melter. The metric dimension of a graph. *Ars Combinatoria*, 2:191–195, 1976.
17. S. Hartung and A. Nichterlein. On the Parameterized and Approximation Hardness of Metric Dimension. *Proceedings of CCC 2013*, IEEE, pp. 266–276, 2013.
18. M. Hauptmann, R. Schmied, and C. Viehmann. Approximation complexity of Metric Dimension problem. *Journal of Discrete Algorithms*, 14:214–222, 2012.
19. M. Jannesari and B. Omoomi. The metric dimension of the lexicographic product of graphs. *Discrete Mathematics*, 312(22):3349–3356, 2012.
20. S. Khuller, B. Raghavachari, and A. Rosenfeld. Landmarks in graphs. *Discrete Applied Mathematics*, 70:217–229, 1996.
21. A. Sebő and E. Tannier. On metric generators of graphs. *Mathematics of Operations Research*, 29(2):383–393, 2004.
22. P. J. Slater. Leaves of trees. *Congressus Numerantium*, 14:549–559, 1975.
23. P. J. Slater. Dominating and reference sets in a graph. *Journal of Mathematical and Physical Sciences*, 22:445–455, 1988.

Chapter 8

Covering the edges of bipartite graphs by overlapping C_4's

Henning Fernau, Pinar Heggernes, Pim van 't Hof, Daniel Meister, and Reza Saei

Submitted for journal publication

Covering the edges of bipartite graphs by overlapping C_4's *

Henning Fernau[2], Pinar Heggernes[1], Daniel Meister[2], and Reza Saei[1]

[1] Department of Informatics, University of Bergen, Norway,
{pinar.heggernes,pim.vanthof,reza.saeidinvar}@ii.uib.no
[2] Theoretical Computer Science, University of Trier, Germany, {fernau, daniel.meister}@uni-trier.de

Abstract. A C_4-cover of a graph can be seen as a partition of its edge set into edge-set subsets of chordless cycles on four vertices. Computing a C_4-cover of smallest size is already NP-hard on bipartite graphs. We determine classes of bipartite graphs for which the computation problem becomes tractable. Among our results are an explicit formula for the size of smallest C_4-covers of complete bipartite graphs and a linear-time algorithm for computing smallest C_4-covers of trees.

1 Introduction

Graph cover problems are a well-studied and popular research field in graph algorithms and graph combinatorics, that have theoretical as well as practical applications [2,35]. In this paper, we study a graph cover variant of a special flavour.

A chordless cycle is a Hamiltonian graph with an inclusion-minimal edge set. The chordless cycle of length 4 is on four vertices and is denoted as C_4; it is the (unique) 2-regular graph on four vertices, and it is equivalent to the complete bipartite graph $K_{2,2}$. We use copies of C_4 to cover the edge set of graphs. A C_4-cover of a graph is a collection of copies of C_4 such that each edge of the graph occurs in at least one of the copies of C_4. Hochbaum and Levin introduced this C_4-cover problem and studied computational questions about computing the size of smallest C_4-covers [23]. The combinatorial problem is motivated by design questions for optical networks [23,27]. Among other results, Hochbaum and Levin showed that the C_4-cover problem is NP-hard on bipartite graphs, and they gave a $\left(\frac{13}{10}+\varepsilon\right)$-approximation algorithm for the C_4-cover problem on bipartite graphs [23]. Chalopin and Paulusma studied computational complexity aspects in a more general context [9].

The intractability result by Hochbaum and Levin for bipartite graphs and the applications of the problem naturally motivate the search for tractable cases. This is what we are doing in this paper. We consider classes of bipartite graphs and aim at showing tractability of their C_4-cover problems. As our main result, we give a full formula for the size of smallest C_4-covers of complete bipartite graphs. This result seems surprising as a main result, since complete bipartite graphs have an easy-enough combinatorial structure that allows for straightforward solutions of most graph problems. We will see – it is a bit counterintuitive – that both the upper bound and the matching lower bound for complete bipartite graphs turn out non-trivial, which makes the problem interesting and relevant to be studied.

Motivated by the level of complexity of the C_4-cover problem on complete bipartite graphs, we extend our research, and study optimal C_4-covers on further classes of bipartite graphs. Noteworthy results are an explicit formula for the size of smallest C_4-covers of simple chain

*This work was supported by the Research Council of Norway and the German Academic Exchange Service.

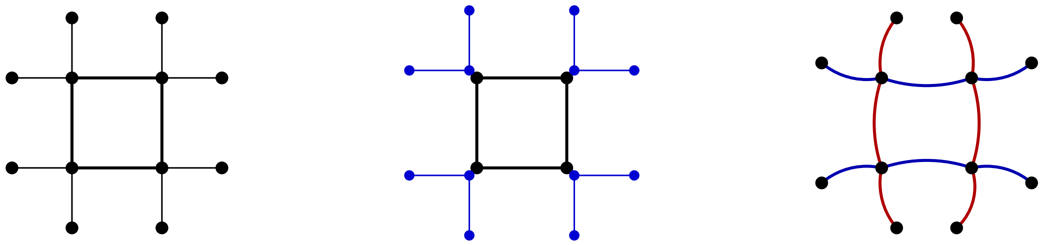


Fig. 1. A C_4 with two pendant vertices per vertex to the left, and two possible C_4-covers. The middle C_4-cover uses a C_4 and four copies of P_3 and is of size 5, and the right-side C_4-cover, using four copies of P_4, is of size 4.

graphs and a partial formula for a more general class of graphs that we call complete four-classes graphs. As a complementary result, we give a linear-time algorithm for computing a smallest C_4-cover of trees. This algorithm computes a P_4-packing of largest size in linear time, contrasting with the NP-hardness of the P_4-packing problem on arbitrary graphs [13,24].

In our context, for a fixed graph H, an H-packing of a graph G embeds copies of H into G in an edge-disjoint manner. We can say that an H-packing desires to determine a subgraph of G that is the edge-disjoint union of copies of H. The implied computational problem seeks to find a largest H-packing. Related is the study of H-decompositions that aims to find an H-packing such that in fact all edges of G are within some copy of the packing. Clearly, H-decompositions (if they exist) are always maximum H-packings. Conversely, an H-cover of G desires to embed G into the union of copies of H, and these copies may overlap in vertices and edges. Now, the related computational problem is a minimization problem: we aim to find a cover that uses the last number of copies of H. Again, H-decompositions (if they exist) provide optimum solutions. The structure of C_4 and the relationship between cover and packing problems suggests the following natural approach to computing a C_4-cover, that we discuss in Section 2 in detail: compute a largest C_4-packing of the given graph G, and then compute a largest P_4-packing of the remainder graph. An easy example, a C_4 with two pendant vertices at each vertex, as it is depicted and discussed in Figure 1, shows that this natural approach must fail on general bipartite graphs. In addition, even if largest C_4-packings exist that are extendable into smallest C_4-covers, other largest C_4-packings may exist that are not extendable into smallest C_4-covers, and the algorithmic problem of determining an extendable C_4-packing still remains; a brief discussion of these problems and examples are given in Section 5. This indicates that these seemingly related packing and covering problems prove to be fundamentally different in many aspects.

In Section 3, we present our main result, the full formula for the size of smallest C_4-covers of complete bipartite graphs. Our results and the given discussions there will show that the approach via packings sketched above works out for complete bipartite graphs, however making a solution unnecessarily complex. The size and structure of largest C_4-packings and of C_4-decompositions of complete bipartite (and similar) graphs have already been the subject of investigations [3,6,10,31,32,33]. We can apply these results to obtain a C_4-cover as an upper bound. A useful lower bound is not implied by any of these results, and the lower bound in Section 3 is one of our main combinatorial results, that we consider of independent interest, as also the proof idea might transfer to similar situations.

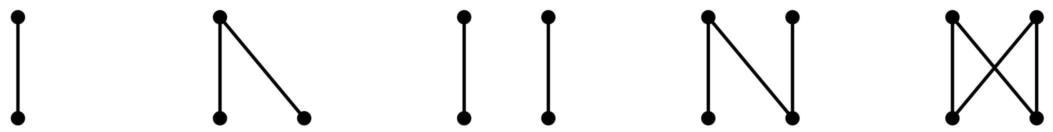


Fig. 2. Five pairwise non-isomorphic subgraphs of C_4 that are induced by non-empty sets of edges of C_4. Noteworthy are the graphs P_4 and C_4, that are the fourth and fifth graph, respectively.

2 Graph preliminaries, the C_4-cover number, and packings

We consider simple, finite, undirected graphs. A graph G is an ordered pair $G = (V, E)$, where $V = V(G)$ is the *vertex set* and $E = E(G)$ is the *edge set* of G. Edges are denoted as uv, which means that u and v are *adjacent* or *neighbours*, and u and v are *incident* to uv. If u and v are not neighbours in G then they are *non-adjacent.* For u a vertex of G, the *neighbourhood* of u, $N_G(u)$, is the set of the neighbours of u in G, and the *degree* of u is $|N_G(u)|$. A graph H is a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. For $F \subseteq E(G)$, the subgraph of G *induced* by F, $G[F]$, has as vertex set the set of vertices of G that are incident to an edge in F, and the edge set of $G[F]$ is F.

Let G and H be graphs, that may share vertices and edges. The *union* of G and H, $G \cup H$, is the graph $(V(G) \cup V(H), E(G) \cup E(H))$. Observe that G and H are subgraphs of $G \cup H$. We say that H is *embeddable* into G if there is an injective mapping $\varphi : V(H) \to V(G)$ that preserves edges, which means for each vertex pair x, y of H that $xy \in E(H)$ implies $\varphi(x)\varphi(y) \in E(G)$. Observe that H is embeddable into G if and only if H is isomorphic to a subgraph of G.

Our results are about classes of bipartite graphs. Let G be a graph. A set X of vertices of G is an *independent set* of G if the vertices in X are pairwise non-adjacent in G. A graph is called *bipartite* if its vertex set can be partitioned into two independent sets. We call such a vertex set partition a *bipartite partition.* The classes of bipartite graphs that we consider will be introduced and defined in the sections of their first appearance.

For our purposes and applications, we define a class of graphs that we call *perfect C_4-graphs.* Intuitively, a perfect C_4-graph consists of overlapping copies of C_4. Formally, the class of perfect C_4-graphs is inductively defined, and we associate an induction-depth parameter: for every set V of vertices, $(V, \emptyset)$ is a perfect C_4-graph of order 0, and for $k \geq 0$ and Q a perfect C_4-graph of order at most k and C a copy of C_4, $Q \cup C$ is a perfect C_4-graph of order at most $k+1$. It is an easy observation that a perfect C_4-graph of order at most k has at most $4k$ edges and can have an arbitrary number of vertices. It is a similarly easy observation that a graph Q is a perfect C_4-graph of order at most k if and only if there are non-empty $F_1, \ldots, F_k \subseteq E(Q)$ such that $F_1 \cup \cdots \cup F_k = E(Q)$ and each of $Q[F_1], \ldots, Q[F_k]$ is embeddable into C_4. Observe that each of $Q[F_1], \ldots, Q[F_k]$ is isomorphic to one of the graphs depicted in Figure 2, and observe that $F_1, \ldots, F_k$ can be chosen pairwise disjoint.

The graph parameter that we are going to study in this paper is the C_4-cover number. The *C_4-cover number* of a graph G, denoted as $\mathrm{cn}(G)$, is the smallest integer k such that G is embeddable into a perfect C_4-graph of order at most k. It is a consequence of above's observation that, for $k \geq 0$, the claim of $\mathrm{cn}(G) \leq k$ is equivalent to the existence of $F_1, \ldots, F_k \subseteq E(G)$ such that $F_1 \cup \cdots \cup F_k = E(G)$ and each of $G[F_1], \ldots, G[F_k]$ is embeddable into C_4 [23]. Note that an edge of G may be contained in more than one of the sets $F_1, \ldots, F_k$, so that an edge of G may be *covered* multiplely.

We relate the C_4-cover problem to packing problems. For an input graph G and a pattern graph H, an *H-packing* of G is a collection $\mathcal{C}$ of pairwise disjoint subsets of $E(G)$ such that for each $F \in \mathcal{C}$, F induces a copy of H in G. By $\mathrm{pack}_H(G)$, we denote the largest size of an

H-packing of G, that is called the *H-packing number* of G. By P_4, we denote the chordless path on four vertices, that is the fourth graph of Figure 2. We use the C_4-packing and P_4-packing number, pack_{C_4} and pack_{P_4}.

Proposition 1. *Let G be a graph. The C_4-cover number of G, $\text{cn}(G)$, is equal to the minimum number of $\left\lceil \frac{1}{2}\big(|E(G)| - \text{pack}_{P_4}(G[Y])\big)\right\rceil - \text{pack}_{C_4}(G[X])$ among all sets X, Y where $X, Y \subseteq E(G)$ and $X \cap Y = \emptyset$.*

Proof. For the lower bound, let $\mathcal{C}$ be a C_4-cover of G, where the sets in $\mathcal{C}$ are assumed pairwise disjoint. We define a partition (X, Y, Z) of $E(G)$:

$$
\begin{aligned}
X &=_{\text{def}} \bigcup_{F \in \mathcal{C} \text{ where } |F|=4} F \\
Y &=_{\text{def}} \bigcup_{F \in \mathcal{C} \text{ where } |F|=3} F \\
Z &=_{\text{def}} \bigcup_{F \in \mathcal{C} \text{ where } |F|\leq 2} F\,;
\end{aligned}
$$

note that each of X, Y, Z may be empty. Also note $|\mathcal{C}| \geq \frac{|X|}{4} + \frac{|Y|}{3} + \left\lceil \frac{|Z|}{2} \right\rceil$, and $\text{pack}_{C_4}(G[X]) = \frac{|X|}{4}$ and $\text{pack}_{P_4}(G[Y]) = \frac{|Y|}{3}$, and $Z = E(G) \setminus (X \cup Y)$. Thus,

$$
\begin{aligned}
|\mathcal{C}| \;\geq\; \left\lceil \frac{|Z|}{2} \right\rceil + \frac{|Y|}{3} + \frac{|X|}{4} &= \left\lceil \frac{|E(G)| - |X| - |Y|}{2} \right\rceil + \frac{|Y|}{3} + \frac{|X|}{4} \\
&= \left\lceil \frac{|E(G)|}{2} - \frac{|Y|}{6} \right\rceil - \frac{|X|}{4} \\
&= \left\lceil \frac{1}{2}\big(|E(G)| - \text{pack}_{P_4}(G[Y])\big) \right\rceil - \text{pack}_{C_4}(G[X])\,;
\end{aligned}
$$

for the correctness of this evaluation, recall that $|X|$ is a multiple of 4 and $|Y|$ is a multiple of 3.

For the converse upper bound, let X, Y be disjoint subsets of $E(G)$. Clearly, G has a C_4-cover of the following size:

$$
\text{pack}_{C_4}(G[X]) + \text{pack}_{P_4}(G[Y]) + \left\lceil \frac{|E(G)| - 4 \cdot \text{pack}_{C_4}(G[X]) - 3 \cdot \text{pack}_{P_4}(G[Y])}{2} \right\rceil ,
$$

which proves the bound.

Proposition 1 interprets our C_4-cover problem as a multivariate packing problem. It explains that the two sets X and Y must be chosen carefully and related to each other, and that $\text{pack}_{C_4}(G[X])$ and $\text{pack}_{P_4}(G[Y])$ are assigned different weights in a solution. Most of our results will apply Proposition 1 as a design basis.

We apply Proposition 1 to obtain some easy results. The expression assumes its smallest possible value in case of $\text{pack}_{C_4}(G) = \frac{1}{4}|E(G)|$, and it assumes its largest value for $X = Y = \emptyset$. This yields this inequality chain:

$$
\text{pack}_{C_4}(G) \;\leq\; \frac{|E(G)|}{4} \;\leq\; \text{cn}(G) \;\leq\; \left\lceil \frac{|E(G)|}{2} \right\rceil .
$$

As a consequence, there is an obvious linear-time 2-approximation algorithm for computing the C_4-cover number of arbitrary graphs.

The C_4-cover numbers of some basic graph classes are easy. The chordless paths and chordless cycles on k vertices are denoted as P_k and C_k, respectively, where $k \geq 3$ is assumed in case of cycles. Since $\mathrm{pack}_{C_4}(P_k) = 0$ for every $k \geq 1$, and since $\mathrm{pack}_{C_4}(C_k) = 0$ if and only if $k \neq 4$, the following two equalities follow by an application of Proposition 1:

$$\mathrm{cn}(P_k) \;=\; \left\lceil \frac{k-1}{3} \right\rceil \qquad \text{and} \qquad \mathrm{cn}(C_k) \;=\; \begin{cases} 2 & \text{, if } k = 3 \\ 1 & \text{, if } k = 4 \\ \mathrm{cn}(P_{k+1}) & \text{, if } k \geq 5\,, \end{cases}$$

where $\mathrm{pack}_{P_4}(P_k) = \left\lfloor \frac{k-1}{3} \right\rfloor$.

3 Complete bipartite graphs

Let n and m be integers with $n, m \geq 1$. A *complete bipartite graph* $K_{n,m}$ is a bipartite graph G with a bipartite partition $\{A, B\}$ such that $|A| = n$ and $|B| = m$ and $E(G) = A \times B = \{ab : a \in A \text{ and } b \in B\}$. In this section, we determine the C_4-cover number of complete bipartite graphs, by giving a full formula. We prove this result by explicitly constructing C_4-covers and through a lower bound, that is a consequence of our main combinatorial result of this section.

For the upper bound, a known result about cycle packings of complete bipartite graphs seems useful.

Theorem 1 ([10]). *Let n, m be integers with $n \geq m \geq 1$. Then,*

$$\mathrm{pack}_{C_4}(K_{n,m}) \;=\; \begin{cases} \left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lfloor \frac{m}{2} \right\rfloor & \text{, if } nm \equiv 0 \ (2) \\ \left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lfloor \frac{m}{2} \right\rfloor + \left\lfloor \frac{m}{4} \right\rfloor & \text{, if } nm \equiv 1 \ (2)\,. \end{cases}$$

Recall from the definition of $K_{n,m}$ that nm is equal to the number of edges of $K_{n,m}$. If nm is even then n or m is even, and a C_4-packing of $K_{n,m}$ of size at least $\left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lfloor \frac{m}{2} \right\rfloor$ is straightforward by pairing vertices [10]. Such a C_4-packing yields our first upper bound.

Lemma 1. *Let n, m be integers with $n, m \geq 1$ and where $nm \equiv 0$ (2). Then,* $\mathrm{cn}(K_{n,m}) \leq \left\lceil \frac{n}{2} \right\rceil \cdot \left\lceil \frac{m}{2} \right\rceil$.

Proof. Since n or m is even, we may assume without loss of generality that n is even. Due to Theorem 1, $\mathrm{pack}_{C_4}(K_{n,m}) = \left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lfloor \frac{m}{2} \right\rfloor = \frac{n}{2} \cdot \left\lfloor \frac{m}{2} \right\rfloor$, and by an application of Proposition 1, where we choose $Y = \emptyset$, we obtain $\mathrm{cn}(K_{n,m}) \leq \frac{nm}{2} - \frac{n}{2} \cdot \left\lfloor \frac{m}{2} \right\rfloor = \frac{n}{2} \cdot \left\lceil \frac{m}{2} \right\rceil = \left\lceil \frac{n}{2} \right\rceil \cdot \left\lceil \frac{m}{2} \right\rceil$.

The second case of Theorem 1, if nm is odd, requires a more involved construction in order to gain the amount of $\left\lfloor \frac{m}{4} \right\rfloor$ additional cycles. Furthermore, an analysis of the remaining uncovered edges becomes necessary. Since we are not (necessarily) interested in C_4-packings of largest size, we can provide a suitable C_4-cover that is easy to construct and verify. We discuss an alternative approach below.

Lemma 2. *Let n, m be integers with $n \geq m \geq 1$ and where $nm \equiv 1$ (2). Then,* $\mathrm{cn}(K_{n,m}) \leq \left\lceil \frac{n(m+1)}{4} \right\rceil$.

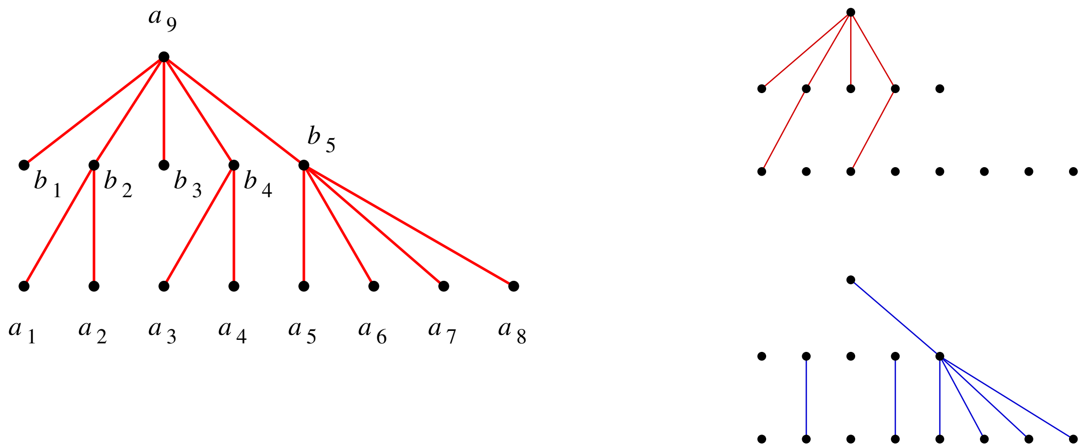


Fig. 3. The figures illustrate the structure of a C_4-cover of a $K_{n,m}$ where n and m are odd, for $K_{9,5}$ as an example, as it is constructed in the proof of Lemma 2. The left-side figure shows the edges of the remainder graph, and the two right-side figures show the two subsequently considered cases.

Proof. Let $\{A, B\}$ be a bipartite partition of $K_{n,m}$, where $|A| = n$ and $|B| = m$, and assume $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_m\}$. We define a partition (X, Y) of $E(K_{n,m})$. Define

$$X =_{\text{def}} \bigcup_{1\leq i<m \text{ where } i \text{ is odd}} \{a_i, a_{i+1}\} \times \{b_1, \dots, b_i, b_{i+2}, \dots, b_m\} \cup \Big(\{a_m, \dots, a_{n-1}\} \times \{b_1, \dots, b_{m-1}\}\Big)$$

$$Y =_{\text{def}} \bigcup_{1\leq i<m \text{ where } i \text{ is odd}} \{a_i, a_{i+1}\} \times \{b_{i+1}\} \cup \Big(\{a_m, \dots, a_{n-1}\} \times \{b_m\}\Big) \cup \Big(\{a_n\} \times \{b_1, \dots, b_m\}\Big).$$

We call $K_{n,m}[Y]$ the *remainder graph*, and its structure is shown in the left-side example of Figure 3.

Observe $\text{pack}_{C_4}(K_{n,m}[X]) \geq \lfloor \frac{n}{2} \rfloor \cdot \text{pack}_{C_4}(K_{2,m-1}) \geq \frac{n-1}{2} \cdot \frac{m-1}{2}$, and consulting the upper right-side example of Figure 3, it is not difficult to verify $\text{pack}_{P_4}(K_{n,m}[Y]) \geq \frac{m-1}{2}$. We apply Proposition 1, and obtain

$$\begin{aligned} \text{cn}(K_{n,m}) &\leq \left\lceil \frac{nm - \text{pack}_{P_4}(K_{n,m}[Y])}{2} \right\rceil - \text{pack}_{C_4}(K_{n,m}[X]) \\ &\leq \left\lceil \frac{nm}{2} - \frac{m-1}{4} \right\rceil - \frac{(n-1)(m-1)}{4} = \left\lceil \frac{nm}{2} - \frac{n(m-1)}{4} \right\rceil = \left\lceil \frac{n(m+1)}{4} \right\rceil, \end{aligned}$$

which proves the lemma.

Billington, Fu, Rodger in [6] study the structure of the remainder graphs of largest C_4-packings of complete bipartite graphs. If nm is odd, the remainder graphs belong to two families, and the following result is a direct consequence, where L is the edge set of the remainder graph:

$$\text{pack}_{P_4}(K_{n,m}[L]) = \begin{cases} 0 & , \text{ if } m \equiv 1 \ (4) \\ 1 & , \text{ if } m \equiv 3 \ (4). \end{cases}$$

Applying Proposition 1 and evaluating the expression will yield the result of Lemma 2, however by relying on stronger and more complex results than we are in need of. It is an interesting observation in this context that our construction in the proof of Lemma 2 aims at a maximization of pack_{P_4}, whereas the sketched alternative approach would minimize pack_{P_4}.

Neither of the two approaches proves optimality of the obtained C_4-covers.

We show that our upper bounds on the C_4-cover number of $K_{n,m}$ are optimal, by proving a matching lower bound. As a combinatorial tool, we define an auxiliary graph. Let G be a graph, and let $\mathcal{C}$ be a C_4-cover of G. The auxiliary graph $\mathrm{aux}(G, \mathcal{C})$ is as follows:

v) $\mathrm{aux}(G, \mathcal{C})$ has as vertex set those vertices x of G for which there is $F \in \mathcal{C}$ such that x is a vertex of degree 1 in $G[F]$;
e) for a vertex pair u, v of $\mathrm{aux}(G, \mathcal{C})$, u and v are adjacent if and only if there is $F \in \mathcal{C}$ such that: either u and v are adjacent in $G[F]$, or $|F| = 3$ and u and v are vertices of degree 1 in $G[F]$.

Informally, we can say that u and v are adjacent in $\mathrm{aux}(G, \mathcal{C})$ if there is $F \in \mathcal{C}$ such that u and v are the end vertices of a path of $G[F]$ of odd length. For an intuitive understanding of $\mathrm{aux}(G, \mathcal{C})$, if we say that each F with $|F| = 4$ "covers edges optimally" then we can say that the auxiliary graph represents information about some F that do not cover edges optimally.

Lemma 3. *Let G be a graph, and let $\mathcal{C}$ be a C_4-cover of G. Let K be an independent set of $\mathrm{aux}(G, \mathcal{C})$. Then, $|\mathcal{C}| \geq \frac{1}{4}|E(G)| + \frac{1}{4}|K|$.*

Proof. Let $u_1, \ldots, u_t$ be the vertices in K. We show $|E(G)| \leq 4|\mathcal{C}| - t$, which directly proves the claim.

For $1 \leq i \leq t$, let F_i be in $\mathcal{C}$ such that u_i is a vertex of degree 1 in $G[F_i]$. Observe that $F_1, \ldots, F_t$ exist according to the definition of $\mathrm{aux}(G, \mathcal{C})$. Clearly, $|F_i| \leq 3$, and thus, $|F_1| + \cdots + |F_t| \leq 3t$. If $F_1, \ldots, F_t$ are pairwise different then

$$\big|E(G)\big| \;=\; \sum_{F \in \mathcal{C}} \big|F\big| \;\leq\; 4\big|\mathcal{C}\big| - t\,,$$

and we can already conclude. Otherwise, some set occurs more than once.

For a contradiction, suppose that there are $1 \leq i < j < k \leq t$ such that $F_i = F_j = F_k$. This means that u_i, u_j, u_k are vertices of degree 1 of $G[F_i]$, and this means $|F_i| = 2$ and that $G[F_i]$ has four vertices of degree 1 each, and so, two out of u_i, u_j, u_k are adjacent in $G[F_i]$, thus adjacent in $\mathrm{aux}(G, \mathcal{C})$, a contradiction to the choice of K as an independent set of $\mathrm{aux}(G, \mathcal{C})$. Hence, each of $F_1, \ldots, F_t$ occurs at most twice.

Let $1 \leq i < j \leq t$ be such that $F_i = F_j$. Then, u_i and u_j are vertices of degree 1 in $G[F_i]$. If $|F_i| = 3$ then u_i and u_j are adjacent in $\mathrm{aux}(G, \mathcal{C})$, which gives a contradiction, and thus, we can conclude $|F_i| \leq 2$.

Let $\mathcal{F}_1$ be the set of those among $F_1, \ldots, F_t$ that occur exactly once, and let $\mathcal{F}_2$ be the set of those among $F_1, \ldots, F_t$ that occur at least twice. Then, $t = \big|\mathcal{F}_1\big| + 2 \cdot \big|\mathcal{F}_2\big|$, and

$$\begin{aligned}\big|F_1 \cup \cdots \cup F_t\big| &= \sum_{F \in \mathcal{F}_1} \big|F\big| + \sum_{F \in \mathcal{F}_2} \big|F\big| \\ &\leq 3 \cdot \big|\mathcal{F}_1\big| + 2 \cdot \big|\mathcal{F}_2\big| \\ &= 4 \cdot \big|\mathcal{F}_1 \cup \mathcal{F}_2\big| - \big|\mathcal{F}_1\big| - 2 \cdot \big|\mathcal{F}_2\big| \;=\; 4 \cdot \big|\{F_1, \ldots, F_t\}\big| - t\,,\end{aligned}$$

which proves the bound.

Corollary 1. *Let G be a bipartite graph with bipartite partition $\{A, B\}$, and let $\mathrm{odd}_G(A)$ be the set of vertices of odd degree in A. Then,* $\mathrm{cn}(G) \geq \frac{1}{4}\big(|E(G)| + |\mathrm{odd}_G(A)|\big)$.

Proof. Let $\mathcal{C}$ be a C_4-cover of G. Let x be a vertex in $\mathrm{odd}_G(A)$, and consider $F \in \mathcal{C}$. Since each vertex of $G[F]$ has degree at most 2, there is $F' \in \mathcal{C}$ such that x is a vertex of $G[F']$ of degree 1, and x is a vertex of $\mathrm{aux}(G, \mathcal{C})$. Next, let x, y be an adjacent vertex pair of $\mathrm{aux}(G, \mathcal{C})$. Then, there is $F'' \in \mathcal{C}$ such that x, y is an adjacent vertex pair of $G[F'']$ or $|F''| = 3$, and x and y are the two vertices of $G[F'']$ of degree 1. In both cases, $x \in A$ and $y \in B$, or $x \in B$ and $y \in A$. This means that the vertices from A in $\mathrm{aux}(G, \mathcal{C})$ form an independent set of $\mathrm{aux}(G, \mathcal{C})$, and the vertices of odd degree in particular. So, $|\mathcal{C}| \geq \frac{1}{4}|E(G)| + \frac{1}{4}|\mathrm{odd}_G(A)|$ due to Lemma 3.

We combine our results, and obtain the full formula for the C_4-cover number of complete bipartite graphs.

Theorem 2. *Let n, m be integers with $n \geq m \geq 1$. Then,*

$$\mathrm{cn}(K_{n,m}) \;=\; \begin{cases} \left\lceil \frac{n}{2} \right\rceil \cdot \left\lceil \frac{m}{2} \right\rceil & \text{, if } nm \equiv 0 \ (2) \\ \left\lceil \frac{n(m+1)}{4} \right\rceil & \text{, if } nm \equiv 1 \ (2)\,. \end{cases}$$

Proof. The upper bounds on $\mathrm{cn}(K_{n,m})$ are due to Lemmas 1 and 2.

We prove the lower bounds. If n and m are even then $\mathrm{cn}(K_{n,m}) \geq \frac{1}{4} \cdot |E(K_{n,m})| = \frac{nm}{4}$ proves the matching lower bound already. Otherwise, by symmetry, assume that m is odd. Let $\{A, B\}$ be a bipartite partition of $K_{n,m}$, where $|A| = n$ and $|B| = m$. Each vertex in A is of degree m, and thus, it is a vertex of odd degree, and $\mathrm{odd}_{K_{n,m}}(A) = A$. We apply Corollary 1:

$$\mathrm{cn}(K_{n,m}) \;\geq\; \frac{|E(K_{n,m})| + |\mathrm{odd}_{K_{n,m}}(A)|}{4} \;=\; \frac{nm+n}{4} \;=\; \frac{n(m+1)}{4}\,,$$

which proves the matching lower bounds for the two remaining cases.

4 Simple chain graphs and trees

In the preceding section, we studied complete bipartite graphs, which are the bipartite graphs with maximal edge sets. In this section, we consider two other classes of bipartite graphs: graphs that contain half of the possible edges, and the minimal connected bipartite graphs. These are respectively the simple chain graphs and the trees.

A bipartite graph G with bipartite partition $\{A, B\}$ is called a *chain graph* if the vertices in A can be ordered as $\langle a_1, \ldots, a_n \rangle$ such that $N_G(a_1) \supseteq \cdots \supseteq N_G(a_n)$. It is known and easy to verify that B then admits an ordering $\langle b_1, \ldots, b_m \rangle$ such that $N_G(b_1) \subseteq \cdots \subseteq N_G(b_m)$ [7]. We call a chain graph a *simple chain graph* if all neighbourhoods are non-empty and all neighbourhood inclusions are proper. The *order* of a simple chain graph is the size of A. Observe that $|A| = |B|$ must hold for simple chain graphs. Two examples of simple chain graphs are depicted in Figure 4.

Theorem 3. *Let G be a simple chain graph of order n. Then,*

$$cn(G) = \left\lceil \frac{n^2 + 2n}{8} \right\rceil .$$

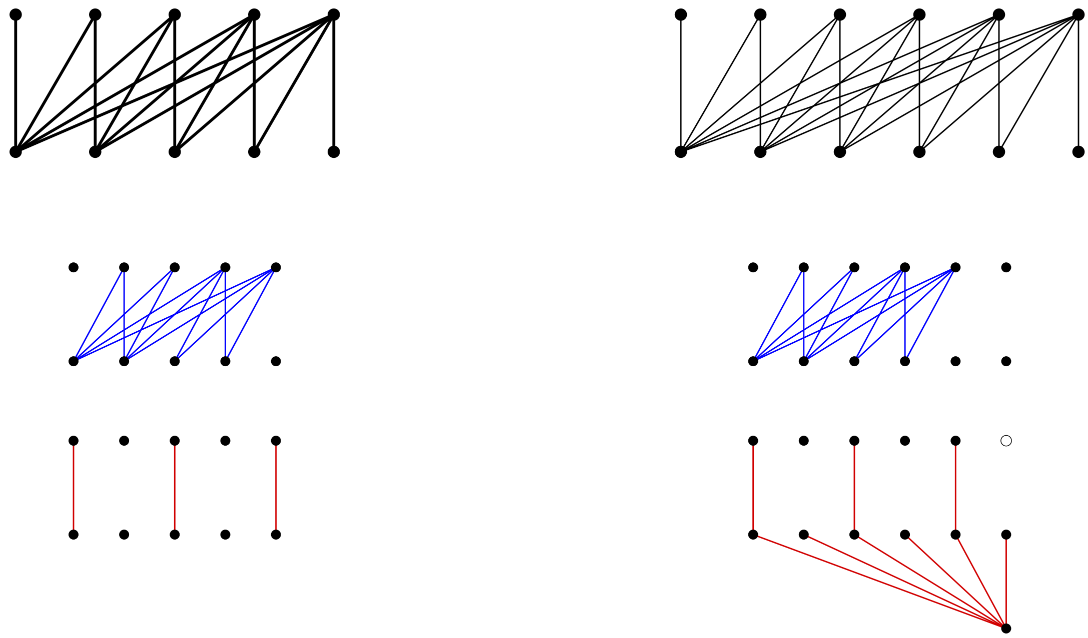


Fig. 4. The left-side figures show a simple chain graph of order 5 on top and below the two cases for a C_4-cover, as constructed in the proof of Theorem 3. Analogously, a simple chain graph of order 6 and the construction of a C_4-cover in the right-side figures.

Proof. Let $\{A, B\}$ be a bipartite partition of G, where $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ and $N_G(a_1) \supset \cdots \supset N_G(a_n)$ and $N_G(b_1) \subset \cdots \subset N_G(b_n)$. The lower bound is proved by a straightforward application of Corollary 1:

$$\begin{aligned}\mathrm{cn}(G) &\geq \frac{1}{4}|E(G)| + \frac{1}{4}|\mathrm{odd}_G(A)| \\ &= \frac{1}{4}\sum_{i=1}^{n} i + \frac{1}{4} \cdot \left\lceil \frac{n}{2} \right\rceil \;=\; \frac{1}{4} \cdot \frac{n(n+1)}{2} + \frac{1}{4} \cdot \left\lceil \frac{n}{2} \right\rceil \;=\; \frac{1}{4} \cdot \left\lceil \frac{n(n+2)}{2} \right\rceil .\end{aligned}$$

Next, we prove the upper bound. Choose $n' = n$ if n is odd, and $n' = n - 1$ if n is even. We partition $E(G)$ as follows:

$$\begin{aligned}&\Big\{\{a_i, a_{i+1}\} \times \{b_{i+1}, \ldots, b_{n'}\} : \quad 1 \leq i \leq n-2 \text{ where } i \text{ is odd}\Big\} \\ \cup &\Big\{\{a_i b_i : \ 1 \leq i \leq n \text{ and } i \text{ is odd}\}\Big\} \\ \cup &\Big\{\{a_1, \ldots, a_n\} \times \{b_n\} : \ \text{if } n \text{ is even}\Big\}.\end{aligned}$$

Thus, if n is even, consult the right-side figures of Figure 4, then

$$\begin{aligned}\mathrm{cn}(G) &\leq \sum_{i=1}^{\frac{n}{2}-1} \mathrm{cn}(K_{2,2i}) + \frac{n}{2} \\ &= \sum_{i=1}^{\frac{n}{2}-1} \frac{2 \cdot 2i}{4} + \frac{n}{2} \;=\; \sum_{i=1}^{\frac{n}{2}-1} i + \frac{n}{2} \;=\; \frac{1}{2} \cdot \left(\frac{n}{2} - 1\right) \cdot \frac{n}{2} + \frac{n}{2} \;=\; \frac{1}{2} \cdot \frac{n+2}{2} \cdot \frac{n}{2} ,\end{aligned}$$

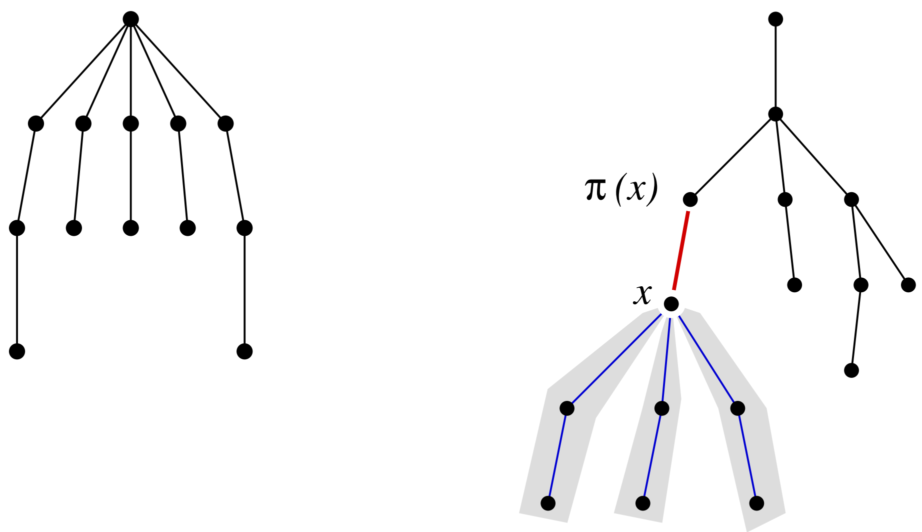


Fig. 5. The left-side figure shows a tree, whose C_4-cover number is 5, and the right-side figure illustrates the subtree edges of the edge $e = x\pi(x)$, as defined in the proof of Theorem 4.

and if n is odd, consult the left-side figures of Figure 4, then

$$\begin{aligned}\mathrm{cn}(G) &\leq \sum_{i=1}^{\frac{n-1}{2}} \mathrm{cn}(K_{2,2i}) + \left\lceil \frac{\lceil \frac{n}{2} \rceil}{2} \right\rceil \\ &= \sum_{i=1}^{\frac{n-1}{2}} i + \left\lceil \frac{n}{4} \right\rceil = \frac{1(n-1)(n+1)}{2 \cdot 2 \cdot 2} + \left\lceil \frac{n}{4} \right\rceil = \frac{n^2-1}{8} + \left\lceil \frac{n}{4} \right\rceil = \left\lceil \frac{n^2+2n}{8} - \frac{1}{8} \right\rceil .\end{aligned}$$

This completes the proof.

We now turn to trees. A *tree* is a graph whose vertices admit an ordering $\langle x_1, \ldots, x_n \rangle$ such that each vertex x_i for $1 < i \leq n$ has a unique neighbour among $x_1, \ldots, x_{i-1}$. The unique neighbour is often called the *predecessor*, and we therefore call $\langle x_1, \ldots, x_n \rangle$ a *predecessor ordering*. Trees have already appeared in previous sections as subcases, for example in Figures 3 and 4.

The C_4-cover number of trees cannot be expressed as a formula analogous to our previous results. Instead, the C_4-cover number of trees is obtained algorithmically. Before, consider the left-side tree of Figure 5 as an easy example: the tree has twelve edges, its C_4-cover number is 5, and the lower bound result of Corollary 1 yields $\frac{12+4}{4} = 4$.

Theorem 4. *The C_4-cover number of trees can be computed in linear time.*

Proof. Let T be a tree, that has at least two vertices. Since $\mathrm{pack}_{C_4}(T) = 0$ obviously, it suffices to compute $\mathrm{pack}_{P_4}(T)$ due to Proposition 1. We do this in a bottom-up fashion on T.

We fix a predecessor ordering $\langle x_1, \ldots, x_n \rangle$ for T, and we assume that x_1 has exactly one neighbour in T. For every vertex x of T with $x \neq x_1$, let $\pi(x)$ be the predecessor of x. Recall that for each edge e of T, there is a vertex x of T with $x \neq x_1$ such that $e = x\pi(x)$, i.e., e is the connecting edge between x and its predecessor $\pi(x)$.

We associate with each edge its *subtree edges*. Let e be an edge of T with $e = x\pi(x)$, and define:

$$E(e) =_{\mathrm{def}} \{e\} \cup \bigcup_{y \in N_T(x) \setminus \{\pi(x)\}} E(yx) ;$$

for an illustration, consult the right-side graph of Figure 5.

To each edge $e = x\pi(x)$ of T, we assign the triple $t(e) = \big(a(e), b(e), c(e)\big)$ with the following meaning:

$$a(e) = \mathrm{pack}_{P_4}(T[E(e)])$$
$$b(e) = \mathrm{pack}_{P_4}(T[E(e) \setminus \{e\}])$$
$$c(e) = \begin{cases} \max_{y \in N_T(x)\setminus\{\pi(x)\}} \mathrm{pack}_{P_4}(T[E(e) \setminus \{e, yx\}]) & , \text{ if } E(e) \neq \{e\} \\ -1 & , \text{ if } E(e) = \{e\}\,. \end{cases}$$

The following is easy to see: $a(e) \geq b(e) \geq c(e) \geq a(e) - 1$. We particularly consider the following two cases about the triples: $a(e) = b(e) = c(e)$ and $a(e) = b(e) > c(e)$. We show how to obtain $\big(a(e), b(e), c(e)\big)$ inductively. Let $e = x\pi(x)$ be an edge of T. If $E(e) = \{e\}$ then $t(e) = \big(a(e), b(e), c(e)\big) = \big(0, 0, -1\big)$. Otherwise, $E(e) \neq \{e\}$.

Let $y_1, \ldots, y_r$ be the neighbours of x in T that are different from $\pi(x)$. We classify $y_1, \ldots, y_r$, where $e_i = y_i x$:

$$Z_1 =_{\mathrm{def}} \big\{y_i : 1 \leq i \leq r \text{ and } a(e_i) = b(e_i) = c(e_i)\big\}$$
$$Z_2 =_{\mathrm{def}} \big\{y_i : 1 \leq i \leq r \text{ and } a(e_i) = b(e_i) > c(e_i)\big\}\,.$$

Then,

$$a(e) = \sum_{i=1}^{r} a(e_i) \;+\; \begin{cases} |Z_1| & , \text{ if } |Z_1| \leq |Z_2| + 1 \\ 1 + \left\lfloor \frac{|Z_1|-1+|Z_2|}{2} \right\rfloor & , \text{ if } |Z_1| > |Z_2| + 1 \end{cases}$$

$$b(e) = \sum_{i=1}^{r} a(e_i) \;+\; \begin{cases} |Z_1| & , \text{ if } |Z_1| \leq |Z_2| \\ \left\lfloor \frac{|Z_1|+|Z_2|}{2} \right\rfloor & , \text{ if } |Z_1| > |Z_2| \end{cases}$$

$$c(e) = \sum_{i=1}^{r} a(e_i) \;+\; \begin{cases} |Z_1| & , \text{ if } |Z_1| \leq |Z_2| - 1 \\ \left\lfloor \frac{|Z_1|+|Z_2|-1}{2} \right\rfloor & , \text{ if } |Z_1| > |Z_2| - 1\,. \end{cases}$$

The correctness of these equalities is not difficult to verify; we give a brief justification. In each of the three equalities, a P_4-packing $\mathcal{F}$ for $E(e)$ is considered, and it is distinguished between sets F in $\mathcal{F}$ containing edges from exactly one of $E(e_1), \ldots, E(e_r), \{e\}$ or exactly two of them; F cannot contain edges from at least three of $E(e_1), \ldots, E(e_r), \{e\}$. The maybe only remarkable case is $c(e)$ for $|Z_1| = |Z_2| = 0$. In this case, $c(e) = a(e_1) + \cdots + a(e_r) - 1$.

It is straightforward to observe that $t(e)$ for each edge of T can be obtained in overall linear time according to the given procedure, and since $E(x_1x_2) = E(T)$, $a(x_1x_2)$ is the size of a largest P_4-packing of T, and we conclude the linear-time algorithm for computing the C_4-cover number of T.

5 Extending complete bipartite graphs

In this section, we particularly focus on our lower-bound result of Corollary 1, and illustrate its power. We already applied this lower bound to prove the correctness of the C_4-cover numbers of complete bipartite graphs (Theorem 2) and simple chain graphs (Theorem 3). We also gave an example before Theorem 4 that shows its limitation, since it fails on simple trees. Here, we define and study a graph class that extends complete bipartite graphs.

A *complete four-classes graph* is a graph G whose vertex set can be partitioned into four sets A, B, C, D such that $E(G) = (A \times B) \cup (B \times C) \cup (C \times D)$. With a complete four-classes graph, we associate the vertex set partition ordering $\langle A, B, C, D\rangle$. Note that a complete four-classes graph is the union of three complete bipartite graphs, namely the complete bipartite graphs with edge sets $A \times B$ and $B \times C$ and $C \times D$, or it is the union of two complete bipartite graphs, namely with edge sets $(A \cup C) \times B$ and $C \times D$. Since $A \cup C$ and $B \cup D$ are independent sets, complete four-classes graphs are bipartite.

Theorem 5. *Let G be a complete four-classes graph with associated vertex set partition ordering $\langle A, B, C, D\rangle$, and let $a = |A|$, $b = |B|$, $c = |C|$, and $d = |D|$. The results of the table hold*

If				**Then**
$\lvert A\rvert$	$\lvert B\rvert$	$\lvert C\rvert$	$\lvert D\rvert$	$cn(G)$
even	*even*			$= \mathrm{cn}(K_{a,b}) + \mathrm{cn}(K_{b+d,c})$
	odd	*even*	*odd*	$= \mathrm{cn}(K_{a,b}) + \mathrm{cn}(K_{b+d,c})$
even	*odd*	*odd*	*even*	$\leq \left\lceil \frac{\lvert V(G)\rvert+\lvert E(G)\rvert-1}{4} \right\rceil - \frac{\min\{a,\ b-1,\ c-1,\ d\}}{2}$

and equality for the last case holds if $a = b - 1 = c - 1 = d$.

Proof. The upper bound of $\mathrm{cn}(G) \leq \mathrm{cn}(K_{a,b}) + \mathrm{cn}(K_{b+d,c})$ is obvious. For the lower bound, we apply Corollary 1 with bipartite partition $\big\{(A \cup C), (B \cup D)\big\}$ and we consider $\mathrm{odd}_G(A \cup C)$ and $\mathrm{odd}_G(B \cup D)$. We summarise the results in the next table:

$\lvert A\rvert$	$\lvert B\rvert$	$\lvert C\rvert$	$\lvert D\rvert$	$\mathrm{odd}_G(A \cup C)$	$\mathrm{odd}_G(B \cup D)$
even	even	even	even	$\emptyset$	$\emptyset$
even	even	even	odd	C	$\emptyset$
even	even	odd	even	$\emptyset$	$B \cup D$
even	even	odd	odd	C	$B \cup D$
even	odd	even	odd	A	$\emptyset$
odd	odd	even	odd	A	B
even	odd	odd	even	$A \cup C$	$B \cup D$

which proves that the upper and the lower bounds match for the first six cases.

We consider the remaining case, when A and D are of even cardinality and B and C are of odd cardinality. Let $m =_{\mathrm{def}} \min\{a, b-1, c-1, d\}$, and let $A', A'', B', B'', \{u\}, C', C'', \{v\}, D', D''$ be pairwise disjoint sets such that $|A'| = |B'| = |C'| = |D'| = m$ and $A = A' \cup A''$ and $B = B' \cup B'' \cup \{u\}$ and $C = C' \cup C'' \cup \{v\}$ and $D = D' \cup D''$. Consider this partition of $E(G)$:

$$\begin{aligned} &\Big\{A \times (B' \cup B''),\ (B' \cup B'') \times (C' \cup C''),\ (C' \cup C'') \times D\Big\} \\ \cup &\Big\{A' \times \{u\},\ \{u\} \times C'\Big\} \cup \Big\{B' \times \{v\},\ \{v\} \times D'\Big\} \\ \cup &\Big\{A'' \times \{u\},\ \{u\} \times C''\Big\} \cup \Big\{B'' \times \{v\},\ \{v\} \times D''\Big\} \cup \Big\{\{uv\}\Big\}, \end{aligned}$$

that partitions $E(G)$ into the edge set of a complete four-classes graph with associated vertex set partition ordering $\langle A, (B' \cup B''), (C' \cup C''), D\rangle$ and a remainder graph. The remainder graph is induced by the edges incident to u and v, that we partitioned into two classes. We obtain a C_4-cover of G by the union of C_4-covers of the three partial graphs, yielding as an upper bound:

$$\begin{aligned}
\operatorname{cn}(G) &\leq \frac{a(b-1)+(b-1)(c-1)+(c-1)d}{4} + \left\lceil \frac{4m}{2} \right\rceil \\
&\quad + \left\lceil \frac{(a+c-1)+(b-1+d)+1-4m}{2} \right\rceil \\
&= \left\lceil \frac{ab+bc+cd}{4} - \frac{a+b+c+d-1}{4} + \frac{a+b+c+d-1}{2} \right\rceil \\
&= \left\lceil \frac{ab+bc+cd}{4} + \frac{a+b+c+d-1}{4} \right\rceil .
\end{aligned}$$

We can improve this C_4-cover by combining the C_4-covers of the first and the second partial graph, as it is illustrated in the upper right-side figure of Figure 6. This yields as an improved upper bound:

$$\begin{aligned}
\operatorname{cn}(G) &\leq \frac{a(b-1)+(b-1)(c-1)+(c-1)d}{4} + 2m - \frac{m}{2} \\
&\quad + \left\lceil \frac{(a+c-1)+(b-1+d)+1-4m}{2} \right\rceil \\
&= \left\lceil \frac{ab+bc+cd}{4} + \frac{a+b+c+d-1}{4} \right\rceil - \frac{m}{2} \\
&= \left\lceil \frac{|E(G)|+|V(G)|-1}{4} \right\rceil - \frac{m}{2} .
\end{aligned}$$

And if $m = a = b-1 = c-1 = d$ then $(a+b+c+d)-1-2m = a+c$, and the upper and the lower bounds match.

The last case of Theorem 5, that is not fully resolved, can illustrate the difficulties of the C_4-cover problem already on graphs with a simple structure, such as our complete four-classes graphs. We consider as an example a complete four-classes graph with vertex set partition cardinality ordering $\langle 6, 3, 3, 6\rangle$. Theorem 5 yields $\left\lceil \frac{18+45-1}{4} \right\rceil - 1 = 15$ as an upper bound on the C_4-cover number, and Corollary 1 yields $\left\lceil \frac{45+9}{4} \right\rceil = 14$ as a lower bound. In Figure 6, we show the construction of a C_4-cover of smallest size, covering the edge set by a C_4-cover of size $5+4+5 = 14$.

Similar ideas for constructing C_4-covers as described in Figure 6 provide better than straightforward C_4-covers also for the remaining and unconsidered cases of complete four-classes graphs, that are when A, B, C, D are all of odd cardinality, when A is the only even-sized class, and when B and C are of even size only. We leave these cases open.

The lessons to be learned from this section are at least twofold: (1) We have shown that the combinatorics of still quite simple bipartite graphs is (possibly surprisingly) rich; moreover, the lower bound results derived earlier are essential to prove optimality of our constructions. (2) The indicated ideas how to deal with bottleneck situations stemming from how and where to put P_4 subgraphs could be a cornerstone of an algorithmic treatment of more general extensions of complete bipartite graphs, where finding an explicit combinatorial formula appears to be out of reach.

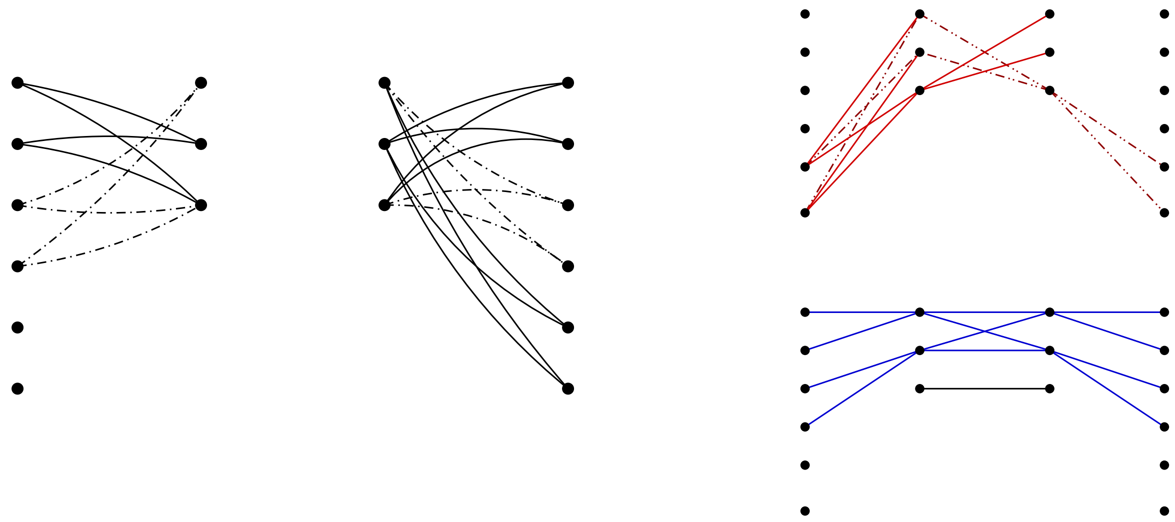


Fig. 6. The construction of a C_4-cover for a complete four-classes graph with associated vertex set partition cardinality ordering $\langle 6, 3, 3, 6 \rangle$. The figures show the three steps of the construction of a C_4-cover of smallest size, as they are discussed at the end of Section 5.

6 Concluding remarks

We studied the C_4-cover problem on bipartite graphs, with the goal of identifying reasons that make the problem hard. To this end, we determined the C_4-cover numbers of particular bipartite graphs for which most combinatorial graph problems admit simple solutions, whereas the C_4-cover problem turns out surprisingly challenging. Particularly noteworthy are our results for complete bipartite graphs and their extension class of complete four-classes graphs. We obtained combinatorial results as well as algorithmic results.

There are further studied cover problems that are related to the C_4-cover problem. In addition to the C_4-cover problem, Hochbaum and Levin also considered the more general $K_{q,q}$-cover problem [23], that is to cover the edges by copies of $K_{q,q}$, thus generalising the C_4-cover problem to balanced complete bipartite graphs of arbitrary but fixed size. Fleischner et al. asked to cover the edges of a graph by arbitrary complete bipartite graphs [14], and Orlin asked to cover the edges of a graph by complete graphs [29]. As a special case, covering the edges of chordal graphs with triangles is hard [17].

The characterisation result of Proposition 1 establishes the C_4-cover problem as a packing problem that maximizes a weight function. We gave an efficient algorithm for trees, and we are not aware of exact algorithms for this problem on any graph class not considered in this paper. A natural graph class to consider and extend our research would be the chain graphs. We expect that an algorithm for this class or its correctness proof will be highly non-trivial, as the complete four-classes graphs form a subclass of chain graphs, and even for these not all cases are completely understood.

An interesting problem on the side would be to study the behaviour of our joint $\{C_4, P_4\}$-packing and the classical C_4- and P_4-packing problems, that are all hard for general graphs [23,24,13]. Also, it would be interesting to characterize the class of graphs for which the approach to first find the biggest C_4-packing and then the biggest P_4-packing for the remainder graph always leads to optimal C_4-coverings. Related problems were considered by several authors who studied the decomposition of complete tripartite and complete bipartite graphs into edge-set disjoint copies of cycles [5,11,31]. In the spirit of [34], we could also address our problem as finding the smallest $\{C_4, P_4, 2K_2, P_3, K_2\}$-decomposition of a graph, which leads to natural more general questions. Another related topic is the investigation of edge covers by connected constrained bipartite subgraphs [1,28].

However, H-cover problems pose a type of combinatorial and computational problem with quite a special flavour. This type of problems deserves further studies. For instance, can we hope for a classification of computationally hard situations in terms of H for H-cover problems, as was obtained (at least partially) for packing problems of various types and for different variants of computational hardness, see [4,12,13,24,25,26,30]? Likewise, a good combinatorial understanding of H-cover problems is lacking. Another perspective is possible given by considering related graph-editing problems that have become quite popular recently [15,16,18,19,20,21,22]. For instance, our problem appears to be related to the question of adding as few edges as possible to the input graph so that the resulting multigraph possesses a C_4-decomposition; the only difference being that added edges might have different weights in our setting. To our knowledge, edit problems into multigraphs have not yet been considered at all, but they make perfect sense, returning once more to the original motivation of this problem [23,27], as the networks showing up in practice are in fact multigraphs. Also, cycle decompositions of multigraph have been studied in the literature on combinatorial designs [8].

References

1. J. Amilhastre, M.-C. Vilarem, and P. Janssen. Complexity of minimum biclique cover and minimum biclique decomposition for bipartite domino-free graphs. *Discrete Applied Mathematics*, 86(2-3):125–144, 1998.
2. M. Ashley, T. Berger-Wolf, P. Berman, W. Chaovalitwongse, B. DasGupta, M. Y. Kao. On approximating four covering and packing problems. *Journal of Computer and System Sciences*, 75:287–302, 2009.
3. L. W. Beinecke. A Survey of Packings and Coverings of Graphs. *The Many Facets of Graph Theory*, Lecture Notes in Mathematics, 110:45–53, Springer, 1969.
4. A. Bernáth and Z. Király. On the tractability of some natural packing, covering and partitioning problems. In *Proceedings of the 7th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications, RIMS, Kyoto University, Japan*, pages 19–28, 2011.
5. E. J. Billington. Decomposing complete tripartite graphs into cycles of lengths 3 and 4. *Discrete Mathematics*, 197/198:123–135, 1999.
6. E. J. Billington, H.-L. Fu, C. A. Rodger. Packing complete multipartite graphs with 4-cycles. *Journal of Combinatorial Designs*, 9:107–127, 2001.
7. A. Brandstädt, V. B. Le, J. P. Spinrad. *Graph Classes: A Survey.* SIAM Monographs on Discrete Mathematics and Applications, SIAM, 1999.
8. D. Bryant, D. Horsley, B. Maenhaut, and B. R. Smith. Cycle decompositions of complete multigraphs. *Journal of Combinatorial Designs*, 19(1):42–69, 2011.
9. J. Chalopin and D. Paulusma. Packing bipartite graphs with covers of complete bipartite graphs. *Discrete Applied Mathematics*, 168:40–50, 2014.
10. G. Chartrand, D. Geller, S. Hedetniemi. Graphs with Forbidden Subgraphs. *Journal of Combinatorial Theory, Series B*, 10:12–41, 1971.
11. C.-C. Chou, C.-M. Fu, W.-C. Huang. Decomposition of $K_{m,n}$ into short cycles. *Discrete Mathematics*, 197/198:195–203, 1999.
12. H. Dell and D. Marx. Kernelization of packing problems. In Y. Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 68–81. SIAM, 2012.
13. D. Dor and M. Tarsi. Graph decomposition is NP-complete: A complete proof of Holyer's conjecture. *SIAM Journal on Computing*, 26:1166–1187, 1997.
14. H. Fleischner, E. Mujuni, D. Paulusma, S. Szeider. Covering graphs with few complete bipartite subgraphs. *Theoretical Computer Science*, 410:2045–2053, 2009.

15. F. V. Fomin, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and Y. Villanger. Tight bounds for parameterized complexity of cluster editing with a small number of clusters. *Journal of Computer and System Sciences*, 80(7):1430–1447, 2014.
16. F. V. Fomin and Y. Villanger. Subexponential parameterized algorithm for minimum fill-in. *SIAM Journal on Computing*, 42(6):2197–2216, 2013.
17. V. Guruswami, C. Pandu Rangan, M.-S. Chang, G. J. Chang, C. K. Wong. The K_r-packing problem. *Computing*, 66:79–89, 2001.
18. P. Heggernes and F. Mancini. Minimal split completions. *Discrete Applied Mathematics*, 157(12):2659–2669, 2009.
19. P. Heggernes and C. Papadopoulos. Single-edge monotonic sequences of graphs and linear-time algorithms for minimal completions and deletions. *Theoretical Computer Science*, 410(1):1–15, 2009.
20. P. Heggernes, P. van 't Hof, B. Lévêque, D. Lokshtanov, and C. Paul. Contracting graphs to paths and trees. *Algorithmica*, 68(1):109–132, 2014.
21. P. Heggernes, P. van 't Hof, B. Lévêque, and C. Paul. Contracting chordal graphs and bipartite graphs to paths and trees. *Discrete Applied Mathematics*, 164:444–449, 2014.
22. P. Heggernes, P. van 't Hof, D. Lokshtanov, and C. Paul. Obtaining a bipartite graph by contracting few edges. *SIAM Journal of Discrete Mathematics*, 27(4):2143–2156, 2013.
23. D. S. Hochbaum and A. Levin. Covering the edges of bipartite graphs using $K_{2,2}$ graphs. *Theoretical Computer Science*, 411:1–9, 2010.
24. I. Holyer. The NP-completeness of some edge-partition problems. *SIAM Journal on Computing*, 10:713–717, 1981.
25. D. G. Kirkpatrick and P. Hell. On the completeness of a generalized matching problem. In *ACM Symposium on Theory of Computing STOC*, pages 240–245, 1978.
26. D. G. Kirkpatrick and P. Hell. On the complexity of general graph factor problems. *SIAM Journal on Computing*, 12:601–609, 1983.
27. A. Levin. Private communication, 2013.
28. L. Liberti, L. Alfandari, and M.-C. Plateau. Edge cover by connected bipartite subgraphs. *Annals of Operations Research*, 188(1):307–329, 2011.
29. J. Orlin. Contentment in graph theory: Covering graphs with cliques. *Indagationes Mathematicae (Proceedings)*, 80:406–424, 1977.
30. M. D. Plummer. Graph factors and factorization: 1985–2003: A survey. *Discrete Mathematics*, 307(7–8):791 – 821, 2007.
31. J. L. Ramírez-Alfonsín. Cycle decompositions of complete and complete multipartite graphs. *Australasian Journal of Combinatorics*, 11:233–238, 1995.
32. J. Schönheim and A. Bialostocki. Packing and Covering of the Complete Graph with 4-Cycles. *Canadian Mathematical Bulletin*, 18:703–708, 1975.
33. D. Sotteau. Decomposition of $K_{m,n}$ ($K^*_{m,n}$) into Cycles (Circuits) of Length $2k$. *Journal of Combinatorial Theory, Series B*, 30:75–81, 1981.
34. T. Sousa. Decompositions of graphs into 5-cycles and other small graphs. *The Electronic Journal of Combinatorics*, 12 (#R49), 2005.
35. R. Yuster. Combinatorial and computational aspects of graph packing and graph decomposition. *Computer Science Review*, 1:12–26, 2007.

A Verifying alternative calculations for Lemma 2

If $m \equiv 1$ (4) is the case, we obtain due to Proposition 1:

$$\begin{aligned}
\mathrm{cn}(K_{n,m}) &\leq \left\lceil \frac{|E(K_{n,m})|}{2} \right\rceil - \mathrm{pack}_{C_4}(K_{n,m}[X]) \\
&= \left\lceil \frac{nm}{2} \right\rceil - \left\lfloor \frac{n}{2} \right\rfloor \cdot \frac{m-1}{2} - \frac{m-1}{4} \\
&= \left\lceil \frac{nm}{2} \right\rceil - 2\left\lfloor \frac{n}{2} \right\rfloor \cdot \frac{m-1}{4} - \frac{m-1}{4} \\
&= \left\lceil \frac{nm}{2} \right\rceil - (2\left\lfloor \frac{n}{2} \right\rfloor + 1) \cdot \frac{m-1}{4} \\
&= \left\lceil \frac{nm}{2} \right\rceil - \frac{nm-n}{4} \\
&= \left\lceil \frac{2nm}{4} \right\rceil - \frac{nm-n}{4} \\
&= \left\lceil \frac{nm+n}{4} \right\rceil
\end{aligned}$$

If $m \equiv 3$ (4) is the case, we obtain due to Proposition 1:

$$\begin{aligned}
\mathrm{cn}(K_{n,m}) &\leq \left\lceil \frac{|E(K_{n,m})|-1}{2} \right\rceil - \mathrm{pack}_{C_4}(K_{n,m}[X]) \\
&= \left\lceil \frac{nm-1}{2} \right\rceil - \frac{(n-1)(m-1)}{4} - \frac{m-3}{4} \\
&= \left\lceil \frac{nm-1}{2} \right\rceil - \frac{n(m-1)-2}{4} \\
&= \left\lceil \frac{2nm-2-n(m-1)+2}{4} \right\rceil \\
&= \left\lceil \frac{nm+n}{4} \right\rceil
\end{aligned}$$

Chapter 9

Finding Disjoint Paths in Split Graphs

Pinar Heggernes, Pim van ’t Hof, Erik Jan van Leeuwen, and Reza Saei

Theory of Computing Systems. To appear.
(A preliminary version appeared in SOFSEM 2014, LNCS 8327: 315-326, Springer.)

Finding Disjoint Paths in Split Graphs*

Pinar Heggernes[1], Pim van 't Hof[1], Erik Jan van Leeuwen[2], and Reza Saei[1]

[1] Department of Informatics, University of Bergen, Norway,
{pinar.heggernes,pim.vanthof,reza.saeidinvar}@ii.uib.no
[2] MPI für Informatik, Campus E1 4, 66123 Saarbrücken, Germany, erikjan@mpi-inf.mpg.de

Abstract. The well-known Disjoint Paths problem takes as input a graph G and a set of k pairs of terminals in G, and the task is to decide whether there exists a collection of k pairwise vertex-disjoint paths in G such that the vertices in each terminal pair are connected to each other by one of the paths. This problem is known to be NP-complete, even when restricted to planar graphs or interval graphs. Moreover, although the problem is fixed-parameter tractable when parameterized by k due to a celebrated result by Robertson and Seymour, it is known not to admit a polynomial kernel unless NP $\subseteq$ coNP/poly. We prove that Disjoint Paths remains NP-complete on split graphs, and show that the problem admits a kernel with $O(k^2)$ vertices when restricted to this graph class. We furthermore prove that, on split graphs, the edge-disjoint variant of the problem is also NP-complete and admits a kernel with $O(k^3)$ vertices. To the best of our knowledge, our kernelization results are the first non-trivial kernelization results for these problems on graph classes.

1 Introduction

Finding vertex-disjoint or edge-disjoint paths with specified endpoints is a classical and fundamental problem in algorithmic graph theory and combinatorial optimization, with applications in such areas as VLSI layout, transportation networks, and network reliability; see, for example, the surveys by Frank [9] and by Vygen [26]. The Vertex-Disjoint Paths problem takes as input a graph G and a set of k pairs of terminals in G, and the task is to decide whether there exists a collection of k pairwise vertex-disjoint paths in G such that the vertices in each terminal pair are connected to each other by one of the paths. The Edge-Disjoint Paths problem is defined analagously, but here the task is to decide whether there exist k pairwise edge-disjoint paths instead of vertex-disjoint paths.

The Vertex-Disjoint Paths problem was shown to be NP-complete by Karp [15], one year before Even et al. [8] proved that the same holds for Edge-Disjoint Paths. A celebrated result by Robertson and Seymour [24], obtained as part of their groundbreaking graph minors theory, states that the Vertex-Disjoint Paths problem can be solved in $O(n^3)$ time for every fixed k. This implies that Edge-Disjoint Paths can be solved in $O(m^3)$ time for every fixed k. As a recent development, an $O(n^2)$-time algorithm for each of the problems, for every fixed k, was obtained by Kawarabayashi, Kobayashi and Reed [16]. The above results show that both problems are fixed-parameter tractable when parameterized by the number of terminal pairs. On the negative side, Bodlaender, Thomassé and Yeo [3] showed that, under the same parameterization, the Vertex-Disjoint Paths problem does not admit a polynomial kernel, unless NP $\subseteq$ coNP/poly.

Both problems have been intensively studied on graph classes. A trivial reduction from Edge-Disjoint Paths to Vertex-Disjoint Paths implies that the latter is NP-complete on line graphs. By a slightly more complicated argument, Edge-Disjoint Paths can also be

*This research is supported by the Research Council of Norway.

shown to be NP-complete on line graphs[1]. It is known that both problems remain NP-complete when restricted to planar graphs [18,19]. On the positive side, Vertex-Disjoint Paths can be solved in linear time for every fixed k on planar graphs [23], or more generally, on graphs of bounded genus [7,17]. Interestingly, Vertex-Disjoint Paths can be solved in polynomial-time on graphs of bounded treewidth [22], while Edge-Disjoint Paths is NP-complete on series-parallel graphs [21], and thus on graphs of treewidth at most 2. Gurski and Wanke [11] proved that the line graph of a graph of treewidth at most k has clique-width at most $2k+2$, and therefore Vertex-Disjoint Paths (following [11]) and Edge-Disjoint Paths (using the reduction of footnote 1) are NP-complete on graphs of clique-width at most 6. On the other hand, Vertex-Disjoint Paths can be solved in linear time on graphs of clique-width at most 2 [11]. Natarajan and Sprague [20] proved the NP-completeness of Vertex-Disjoint Paths on interval graphs, and thus also on all superclasses of interval graphs such as circular-arc graphs and chordal graphs. On chordal graphs, Vertex-Disjoint Paths is linear-time solvable for each fixed k [14].

Given the fact that the Vertex-Disjoint Paths problem is unlikely to admit a polynomial kernel on general graphs, and the amount of known results for both problems on graph classes, it is surprising that no kernelization result has been known on either problem when restricted to graph classes. Interestingly, even the classical complexity status of both problems has been open on split graphs, i.e., graphs whose vertex set can be partitioned into a clique and an independent set, which form a well-studied graph class and a famous subclass of chordal graphs [4,10].

We present the first non-trivial kernelization algorithms for the Vertex-Disjoint Paths and Edge-Disjoint Paths problems on graph classes by showing that the problems admit kernels with $O(k^2)$ and $O(k^3)$ vertices, respectively, on split graphs. To complement these results, we prove that both problems remain NP-complete on this graph class.

2 Preliminaries

All the graphs considered in this paper are finite, simple, and undirected. We refer to the monograph by Diestel [5] for graph terminology and notation not defined below. A *split graph* is a graph whose vertex set can be partitioned into a clique C and an independent set I, either of which may be empty; such a partition (C, I) is called a *split partition*. Note that, in general, a split graph can have more than one split partition. However, split graphs can be recognized in linear time, and a split partition can be found in linear time if one exists [12]. A *chordal graph* is a graph for which no set of four or more vertices induces a cycle, or more formally, a graph G is chordal if for any $X \subseteq V(G)$ with $|X| \geq 4$, the graph $(X, E(G) \cap (X \times X))$ is not a cycle. Observe that any split graph is chordal.

Let G be a graph. For any vertex v in G, we write $N_G(v)$ to denote the neighborhood of v, and $d_G(v) = |N_G(v)|$ to denote the degree of v. Given a path P in G and a vertex $v \in V(G)$, we say that P *visits* v if $v \in V(P)$.

The two problems we consider in this paper are formally defined as follows:

[1] We briefly sketch a reduction from Edge-Disjoint Paths on general graphs to Edge-Disjoint Paths on line graphs. Replace every edge uv of the original graph G by a three-vertex path u, x_{uv}, v (where x_{uv} is a new vertex), and take the line graph of the resulting graph. Then each edge uv of G corresponds to an edge of the line graph (namely, the edge between the vertices corresponding to edges ux_{uv} and $x_{uv}v$). This creates a correspondence between paths in G and paths in the line graph, and completes the proof.

VERTEX-DISJOINT PATHS	
Instance:	A graph G, and a set $\mathcal{X} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ of k pairs of vertices in G, called *terminals*, where $s_i \neq t_i$ for each $i \in \{1, \ldots, k\}$.
Question:	Do there exist k pairwise vertex-disjoint paths $P_1, \ldots, P_k$ in G such that P_i connects s_i to t_i for each $i \in \{1, \ldots, k\}$?

EDGE-DISJOINT PATHS	
Instance:	A graph G, and a set $\mathcal{X} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ of k pairs of vertices in G, called *terminals*, where $s_i \neq t_i$ for each $i \in \{1, \ldots, k\}$.
Question:	Do there exist k pairwise edge-disjoint paths $P_1, \ldots, P_k$ in G such that P_i connects s_i to t_i for each $i \in \{1, \ldots, k\}$?

Throughout the paper, we write n and m to denote the number of vertices and edges, respectively, of the input graph G in each of the two problems. Note that in both problems, we allow different terminals to coincide. This is motivated by the close relationship between the problem of finding vertex-disjoint and edge-disjoint paths and the problem of finding topological minors and immersions in graphs, respectively [24]. For this reason, we define two paths to be *vertex-disjoint* if they are distinct, and if none of the paths contains an inner vertex of the other. Observe that this definition ensures that no terminal appears as an inner vertex of any path in a solution for VERTEX-DISJOINT PATHS. For EDGE-DISJOINT PATHS, it seems unnatural to impose such a restriction[2]. Therefore, the definition of edge-disjointness is as expected: we define two paths to be *edge-disjoint* if they do not share an edge.

Suppose $(G, \mathcal{X}, k)$ is a yes-instance of the VERTEX-DISJOINT PATHS problem. A solution $\mathcal{P} = \{P_1, \ldots, P_k\}$ for the instance $(G, \mathcal{X}, k)$ is *minimum* if there is no solution $\mathcal{Q} = \{Q_1, \ldots, Q_k\}$ for $(G, \mathcal{X})$ such that $\sum_{i=1}^{k} |E(Q_i)| < \sum_{i=1}^{k} |E(P_i)|$.

For any problem Π, two instances I_1, I_2 of Π are *equivalent* if I_1 is a yes-instance of Π if and only if I_2 is a yes-instance of Π. A *parameterized problem* is a subset $Q \subseteq \Sigma^* \times \mathbb{N}$ for some finite alphabet Σ, where the second part of the input is called the *parameter*. A parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$ is said to be *fixed-parameter tractable* if for each pair $(x, k) \in \Sigma^* \times \mathbb{N}$ it can be decided in time $f(k)\,|x|^{O(1)}$ whether $(x, k) \in Q$, for some function f that only depends on k; here, $|x|$ denotes the length of input x. We say that a parameterized problem Q has a *kernel* if there is an algorithm that transforms each instance (x, k) in time $(|x| + k)^{O(1)}$ into an instance (x', k'), such that $(x, k) \in Q$ if and only if $(x', k') \in Q$ and $|x'| + k' \leq g(k)$ for some function g. Here, g is typically an exponential function of k. If g is a polynomial or a linear function of k, then we say that the problem has a *polynomial* kernel or a *linear* kernel, respectively. We refer the interested reader to the monograph by Downey and Fellows [6] for more background on parameterized complexity. It is known that a parameterized problem is fixed-parameter tractable if and only if it is decidable and has a kernel, and several fixed-parameter tractable problems are known to have polynomial or even linear kernels. Recently, methods have been developed for proving the non-existence of polynomial kernels, under some complexity theoretical assumptions [1,2,3].

In the NP-completeness proofs in Section 3, we will reduce from a restricted variant of the SATISFIABILITY (SAT) problem. In order to define this variant, we need to introduce some terminology. Let x be a variable and c a clause of a Boolean formula φ in conjunctive normal

[2]We mention, however, that our polynomial kernel for EDGE-DISJOINT PATHS, presented in Section 4.2, can be easily modified to take this restriction into account. It suffices to insist in Lemma 8 that, instead of the degree, the number of non-terminal neighbors is at least the number of terminals on it. The rest of the proof goes through (mutatis mutandis). Also, our NP-completeness result is not influenced by this issue.

form (CNF). We say that x *appears* in c if either x or $\neg x$ is a literal of c. If x is a literal of clause c, then we say that x *appears positively* in c. Similarly, if $\neg x$ is a literal of c, then x *appears negatively* in c. Given a Boolean formula φ, we say that a variable x appears positively (respectively negatively) if there is a clause c in φ in which x appears positively (respectively negatively). The following result, which we will use to prove that Vertex-Disjoint Paths is NP-complete on split graphs, is due to Tovey [25].

Theorem 1 ([25]). *The* SAT *problem is NP-complete when restricted to CNF formulas satisfying the following three conditions:*

- *every clause contains two or three literals;*
- *every variable appears in two or three clauses;*
- *every variable appears at least once positively and at least once negatively.*

3 Finding Disjoint Paths in Split Graphs is NP-Hard

Lynch [19] gave a polynomial-time reduction from SAT to Vertex-Disjoint Paths, thereby proving the latter problem to be NP-complete in general. By modifying his reduction, he then strengthened his result and proved that Vertex-Disjoint Paths remains NP-complete when restricted to planar graphs. In this section, we first show that the reduction of Lynch can also be modified to prove that Vertex-Disjoint Paths is NP-complete on split graphs. We then show that Edge-Disjoint Paths is NP-complete on split graphs as well, using a reduction from the Edge-Disjoint Paths problem on general graphs.

We first describe the reduction from SAT to Vertex-Disjoint Paths due to Lynch [19]. Let $\varphi = c_1 \wedge c_2 \wedge \ldots \wedge c_m$ be a CNF formula, and let $v_1, \ldots, v_n$ be the variables that appear in φ. We assume that every variable appears at least once positively and at least once negatively; if this is not the case, then we can trivially reduce the instance to an equivalent instance that satisfies this property. Given the formula φ, we create an instance $(G_\varphi, \mathcal{X}_\varphi)$ of Vertex-Disjoint Paths as follows (see Figure 1 for an illustrative example).

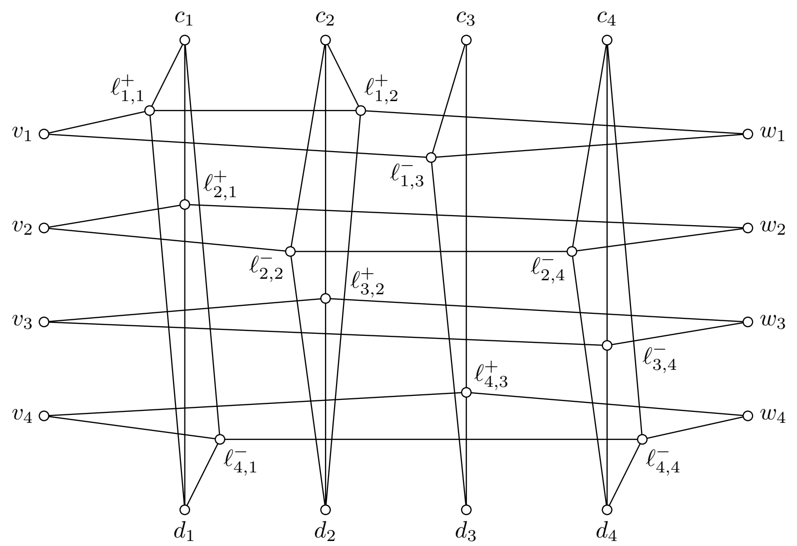


Fig. 1. The graph G_φ constructed in the reduction of Lynch from the CNF formula $\varphi = (v_1 \vee v_2 \vee \neg v_4) \wedge (\neg v_2 \vee v_3 \vee v_1) \wedge (\neg v_1 \vee v_4) \wedge (\neg v_2 \vee \neg v_3 \vee \neg v_4)$.

The vertex set of the graph G_φ consists of three types of vertices: variable vertices, clause vertices, and literal vertices. For each variable v_i in φ, we create two *variable vertices* v_i and w_i; we call (v_i, w_i) a *variable pair*. For each clause c_j, we add two *clause vertices* c_j and d_j and call (c_j, d_j) a *clause pair*. For each clause c_j, we also add a *literal vertex* for each literal that appears in c_j as follows. If c_j contains a literal v_i, that is, if variable v_i appears positively in clause c_j, then we add a vertex $\ell^+_{i,j}$ to the graph, and we make this vertex adjacent to vertices c_j and d_j. Similarly, if c_j contains a literal $\neg v_i$, then we add a vertex $\ell^-_{i,j}$ and make it adjacent to both c_j and d_j. This way, we create $|c_j|$ paths of length exactly 2 between each clause pair (c_j, d_j), where $|c_j|$ is the number of literals in clause c_j.

For each $i \in \{1, \ldots, n\}$, we add edges to the graph in order to create exactly two vertex-disjoint paths between the variable pair (v_i, w_i) as follows. Let $c_{j_1}, c_{j_2}, \ldots, c_{j_p}$ be the clauses in which v_i appears positively, where $j_1 < j_2 < \cdots < j_p$. Similarly, let $c_{k_1}, c_{k_2}, \ldots, c_{k_q}$ be the clauses in which v_i appears negatively, where $k_1 < k_2 < \cdots < k_p$. Note that $p \geq 1$ and $q \geq 1$ due to the assumption that every variable appears at least once positively and at least once negatively. We now add the edges $v_i\ell^+_{i,j_1}$ and $\ell^+_{i,j_p}w_i$, as well as the edges $\ell^+_{i,j_1}\ell^+_{i,j_2}$, $\ell^+_{i,j_2}\ell^+_{i,j_3}, \ldots,$ $\ell^+_{i,j_{p-1}}\ell^+_{i,j_p}$. Let $L^+_i = v_i\ell^+_{i,j_1}\ell^+_{i,j_2}\cdots\ell^+_{i,j_{p-1}}\ell^+_{i,j_p}w_i$ denote the path between v_i and w_i that is created this way. Similarly, we add exactly those edges needed to create the path $L^-_i = v_i\ell^-_{i,k_1}\ell^-_{i,k_2}\cdots\ell^-_{i,j_{q-1}}\ell^-_{i,j_q}w_i$. This completes the construction of the graph G_φ.

Let $\mathcal{X}_\varphi$ be the set consisting of all the variable pairs and all the clause pairs in G_φ, i.e., $\mathcal{X}_\varphi = \{(v_i, w_i) \mid 1 \leq i \leq n\} \cup \{(c_j, d_j) \mid 1 \leq j \leq m\}$. The pair $(G_\varphi, \mathcal{X}_\varphi)$ is the instance of Vertex-Disjoint Paths corresponding to the instance φ of SAT.

Theorem 2 ([19]). *Let φ be a CNF formula. Then φ is satisfiable if and only if $(G_\varphi, \mathcal{X}_\varphi)$ is a yes-instance of the* Vertex-Disjoint Paths *problem.*

Proof. For the sake of completeness, we sketch the proof of Lynch [19] of this theorem. We only prove one direction; the other direction is similar.

Suppose that φ is satisfiable, and consider any truth assignment to the variables such that φ is satisfied. Construct a solution to $(G_\varphi, \mathcal{X}_\varphi)$ as follows. If variable v_i is set to true by the truth assignment, then use L^-_i to connect v_i and w_i; otherwise, use L^+_i. Since φ is satisfied, each clause c_j has a literal that is true. Say this literal is v_i. Therefore, v_i is set to true by the truth assignment, and thus vertex $\ell^+_{i,j}$ is not used to connect v_i and w_i. Hence, it can be used to connect c_j and d_j. It can now be readily verified that $(G_\varphi, \mathcal{X}_\varphi)$ is a yes-instance. □

We are now ready to prove our first result.

Theorem 3. *The* Vertex-Disjoint Paths *problem is NP-complete on split graphs.*

Proof. We reduce from the NP-complete variant of SAT defined in Theorem 1. Let $\varphi = c_1 \wedge c_2 \wedge \ldots \wedge c_m$ be a CNF formula that satisfies the three conditions mentioned in Theorem 1, and let $v_1, \ldots, v_n$ be the variables that appear in φ. Let $(G_\varphi, \mathcal{X}_\varphi)$ be the instance of Vertex-Disjoint Paths constructed from φ in the way described at the beginning of this section. Now let G be the graph obtained from G_φ by adding an edge between each pair of distinct literal vertices, i.e., by adding all the edges needed to make the literal vertices form a clique. The graph G clearly is a split graph.

We will show that $(G, \mathcal{X}_\varphi)$ is a yes-instance of Vertex-Disjoint Paths if and only if $(G_\varphi, \mathcal{X}_\varphi)$ is a yes-instance of Vertex-Disjoint Paths. Since $(G_\varphi, \mathcal{X}_\varphi)$ is a yes-instance of Vertex-Disjoint Paths if and only if the formula φ is satisfiable due to Theorem 2, this suffices to prove the theorem.

If $(G_\varphi, \mathcal{X}_\varphi)$ is a yes-instance of Vertex-Disjoint Paths, then so is $(G, \mathcal{X}_\varphi)$ due to the fact that G is a supergraph of G_φ. Hence, it remains to prove the reverse direction. Suppose

$(G, \mathcal{X}_\varphi)$ is a yes-instance of Vertex-Disjoint Paths. Let $\mathcal{P} = \{P_1, \ldots, P_n, Q_1, \ldots, Q_m\}$ be a minimum solution, where each path P_i connects the two terminals in the variable pair (v_i, w_i), and each path Q_j connects the terminals in the clause pair (c_j, d_j). We will show that all the paths in $\mathcal{P}$ exist also in the graph G_φ, implying that $\mathcal{P}$ is a solution for the instance $(G_\varphi, \mathcal{X}_\varphi)$.

Since $\mathcal{P}$ is a minimum solution and no two terminals coincide, it holds that every path in $\mathcal{P}$ is an induced path in G. By the construction of G, this implies that all the inner vertices of every path in $\mathcal{P}$ are literal vertices. Moreover, since the literal vertices form a clique in G, every path in $\mathcal{P}$ has at most two inner vertices.

Let $j \in \{1, \ldots, m\}$. Since $N_G(c_j) = N_G(d_j)$, the vertices c_j and d_j are non-adjacent, and Q_j is an induced path between c_j and d_j, the path Q_j must have length 2, and its only inner vertex is a literal vertex. Recall that we only added edges between distinct literal vertices when constructing the graph G from G_φ. Hence, the path Q_j exists in G_φ.

Now let $i \in \{1, \ldots, n\}$. We consider the path P_i between v_i and w_i. As we observed earlier, the path P_i contains at most two inner vertices, and all inner vertices of P_i are literal vertices. If P_i has exactly one inner vertex, then P_i exists in G_φ for the same reason as why the path Q_j from the previous paragraph exists in G_φ. Suppose P_i has two inner vertices. Recall the two vertex-disjoint paths L_i^+ and L_i^- between v_i and w_i, respectively, that were defined just above Theorem 2. Since v_i appears in at most three different clauses, at least once positively and at least once negatively, one of these paths has length 2, while the other path has length 2 or 3. Without loss of generality, suppose L_i^+ has length 2, and let ℓ denote the only inner vertex of L_i^+. Note that both v_i and w_i are adjacent to ℓ. Since P_i is an induced path from v_i to w_i with exactly two inner vertices, P_i cannot contain the vertex ℓ. From the construction of G, it is then clear that both inner vertices of P_i must lie on the path L_i^-. This implies that L_i^- must have length 3, and that $P_i = L_i^-$. We conclude that the path P_i exists in G_φ. ◻

We now turn our attention to Edge-Disjoint Paths on split graphs. The following lemma will be used in the proof of Theorem 4 below.

Lemma 1. *Let $(G, \mathcal{X})$ be an instance of* Edge-Disjoint Paths. *Let $\mathcal{X}'$ be a subset of $\mathcal{X}$ such that for every pair $(s, t) \in \mathcal{X}'$, it holds that $d_G(s) = d_G(t) = 1$ and there is a (unique) path P_{st} of length 3 between s and t. If the paths in $\mathcal{P}' = \{P_{st} \mid (s, t) \in \mathcal{X}'\}$ are pairwise edge-disjoint and $(G, \mathcal{X})$ is a yes-instance, then there is a solution $\mathcal{P}$ for the instance $(G, \mathcal{X})$ such that $\mathcal{P}' \subseteq \mathcal{P}$.*

Proof. Suppose the paths in $\mathcal{P}' = \{P_{st} \mid (s, t) \in \mathcal{X}'\}$ are pairwise edge-disjoint and $(G, \mathcal{X})$ is a yes-instance. Let $\mathcal{P}$ be a solution for $(G, \mathcal{X})$ that contains as many paths from $\mathcal{P}'$ as possible. We claim that $\mathcal{P}' \subseteq \mathcal{P}$. Suppose, for contradiction, that there exists a pair of terminals $(s, t) \in \mathcal{X}'$ such that $P_{st} \notin \mathcal{P}$. Let Q denote the path in $\mathcal{P}$ connecting s and t. Recall that both s and t have degree 1 in G, and let u and v be the unique neighbors of s and t, respectively. If none of the paths in $\mathcal{P}$ uses the edge uv, then the set $(\mathcal{P} \setminus Q) \cup P_{st}$ is a solution for $(G, \mathcal{X})$ containing more paths from $\mathcal{P}'$ than $\mathcal{P}$ does, contradicting the choice of $\mathcal{P}$. Hence, there must be a path $P^* \in \mathcal{P}$ that uses the edge uv. Let s^* and t^* be the two terminals that are connected by the path P^*. Let Q^* denote the path between s^* and t^* obtained from P^* by replacing the edge uv by the subpath of Q between u and v. Note that $P^* \notin \mathcal{P}'$, as otherwise the paths $P^* = P_{s^*t^*}$ and P_{st} would share the edge uv, contradicting the assumption that the paths in $\mathcal{P}'$ are pairwise edge-disjoint. Hence, the set obtained from $\mathcal{P}$ by replacing Q by the path P_{st} and replacing P^* by the path Q^* yields a solution for $(G, \mathcal{X})$ that contains one more path from $\mathcal{P}'$. This contradicts the choice of $\mathcal{P}$ and finishes the proof. ◻

We now prove the analogue of Theorem 3 for Edge-Disjoint Paths.

Theorem 4. *The* EDGE-DISJOINT PATHS *problem is NP-complete on split graphs.*

Proof. We reduce from EDGE-DISJOINT PATHS on general graphs, which is well-known to be NP-complete [18]. Let $(G, \mathcal{X})$ be an instance of EDGE-DISJOINT PATHS, where $\mathcal{X} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$. Let G' be the graph obtained from G by adding, for every $i \in \{1, \ldots, k\}$, two new vertices s'_i and t'_i as well as two edges $s'_i s_i$ and $t'_i t_i$. Let $\mathcal{X}' = \{(s'_1, t'_1), \ldots, (s'_k, t'_k)\}$. Clearly, $(G, \mathcal{X})$ is a yes-instance of EDGE-DISJOINT PATHS if and only if $(G', \mathcal{X}')$ is a yes-instance of EDGE-DISJOINT PATHS.

From G', we create a split graph G'' as follows. For every pair of vertices $u, v \in V(G)$ such that $uv \notin E(G)$, we add to G' the edge uv as well as two new terminals s_{uv} and t_{uv} that we connect to u and v respectively. Let G'' be the resulting graph, let $Q = \{(s_{uv}, t_{uv}) \mid u, v \in V(G), uv \notin E(G)\}$ be the terminal pairs that were added to G' to create G'', and let $\mathcal{X}'' = \mathcal{X}' \cup \mathcal{Q}$. We claim that $(G'', \mathcal{X}'')$ and $(G', \mathcal{X}')$ are equivalent instances of EDGE-DISJOINT PATHS. Since G'' is a split graph, this suffices to prove the theorem.

Since G'' is a supergraph of G', it is clear that if $(G', \mathcal{X}')$ is a yes-instance of EDGE-DISJOINT PATHS, then so is $(G'', \mathcal{X}'')$. For the reverse direction, suppose that $(G'', \mathcal{X}'')$ is a yes-instance. For every pair $(s_{uv}, t_{uv}) \in \mathcal{Q}$, let P_{uv} be unique path of length 3 in G'' between s_{uv} and t_{uv}, and let $\mathcal{P}'$ be the set consisting of these paths. By Lemma 1, there is a solution $\mathcal{P}$ for $(G'', \mathcal{X}'')$ such that $\mathcal{P}' \subseteq \mathcal{P}$. Note that the paths in $\mathcal{P}'$ contain all the edges that were added between non-adjacent vertices in G' in the construction of G''. This implies that for every $(s, t) \in \mathcal{X}'$, the path in $\mathcal{P}$ connecting s to t contains only edges that already existed in G'. Hence, $\mathcal{P} \setminus \mathcal{P}'$ is a solution for the instance $(G', \mathcal{X}')$. $\square$

4 Two Polynomial Kernels

In this section, we present polynomial kernels for VERTEX-DISJOINT PATHS and EDGE-DISJOINT PATHS on split graphs, parameterized by the number of terminal pairs. We will denote instances of the parameterized version of the problems by $(G, \mathcal{X}, k)$, where $k = |\mathcal{X}|$ is the parameter.

Before we present the kernels, we introduce some additional terminology. Let $(G, \mathcal{X}, k)$ be an instance of either the VERTEX-DISJOINT PATHS problem or the EDGE-DISJOINT PATHS problem, where $\mathcal{X} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$. Every vertex in $\{s_1, \ldots, s_k, t_1, \ldots, t_k\}$ is called a *terminal*. If $s_i = v$ (resp. $t_i = v$) for some $v \in V(G)$, then we say that s_i (resp. t_i) is a *terminal on* v; note that, in general, there can be more than one terminal on v. A vertex $v \in V(G)$ is a *terminal vertex* if there is at least one terminal on v, and v is a *non-terminal vertex* otherwise. Let $uv \in E(G)$. If there exists a subset $S \subseteq \{1, \ldots, k\}$ of size at least 2 such that $\{u, v\} = \{s_i, t_i\}$ for every $i \in S$, then we call uv a *heavy edge*, and $|S|$ is the *weight* of the edge uv. In other words, a heavy edge of weight s is an edge whose endpoints coincide with $s \geq 2$ pairs of terminals.

Recall that a solution $\mathcal{P}$ for an instance $(G, \mathcal{X}, k)$ of VERTEX-DISJOINT PATHS is *minimum* if there is no solution $\mathcal{Q}$ for $(G, \mathcal{X})$ that uses strictly fewer edges of G. It is easy to see that if all the terminals in $\mathcal{X}$ are distinct, then every path in a minimum solution is an induced path. In general, this is not true if we allow (pairs of) terminals to coincide. The following lemma shows that if the input graph G is chordal, then the non-induced paths in a minimum solution are of a very restricted type. This lemma will be used in the correctness proof of two reduction rules in our kernelization algorithm for VERTEX-DISJOINT PATHS on split graphs.

Lemma 2. *Let* $(G, \mathcal{X}, k)$ *be an instance of* VERTEX-DISJOINT PATHS *such that* G *is a chordal graph. If* $(G, \mathcal{X}, k)$ *is a yes-instance and* $\mathcal{P} = \{P_1, \ldots, P_k\}$ *a minimum solution, then every path* $P_i \in \mathcal{P}$ *satisfies exactly one of the following two statements:*

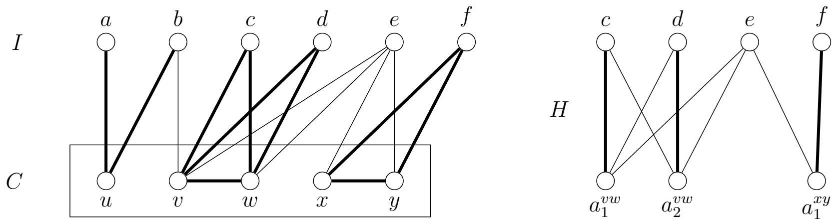


Fig. 2. The left picture shows a split graph G with split partition (C, I). To keep the picture clean, not all edges of C are drawn. Let $\mathcal{X} = \{(a, b), (v, w), (v, w), (v, w), (x, y), (x, y)\}$ and consider the instance $(G, \mathcal{X}, 6)$. Observe that edges vw and xy are heavy. The thick edges indicate a solution $\mathcal{P}$ for this instance. The right figure shows the auxiliary bipartite graph H for $(G, \mathcal{X}, 6)$. The thick edges indicate a matching M' that is induced by $\mathcal{P}$.

- *P_i is an induced path;*
- *P_i is a path of length 2, and there exists a path $P_j \in \mathcal{P}$ of length 1 whose endpoints coincide with the endpoints of P_i.*

Proof. Suppose $(G, \mathcal{X}, k)$ is a yes-instance, and let $\mathcal{P}$ be a minimum solution for this instance. If all the paths in $\mathcal{P}$ are induced, then the statement of the lemma holds. Suppose there is a path $P_i \in \mathcal{P}$ that is not induced. Then there exists an edge $xy \in E(G)$ such that $x, y \in V(P_i)$ and $xy \notin E(P_i)$. Since $\mathcal{P}$ is a minimum solution, there must exist a path $P_j \in \mathcal{P}$ such that $xy \in E(P_j)$. The paths P_i and P_j are vertex-disjoint, so by definition, x and y are the endpoints of both P_i and P_j. It remains to argue that P_i has length 2. For contradiction, suppose P_i has length at least 3. The path P_i together with the edge xy forms a cycle of length at least 4 in G. Since G is chordal, this cycle has a chord e such that at least one endpoint of e is an internal vertex of P_i. Since the paths in $\mathcal{P}$ are pairwise vertex-disjoint, the edge e is not used in any path in $\mathcal{P}$. But then we could have replaced P_i by a shorter path P_i', containing the edge e, to obtain another solution. This contradicts the minimality of $\mathcal{P}$. □

4.1 Polynomial Kernel for Vertex-Disjoint Paths on Split Graphs

Our kernelization algorithm for Vertex-Disjoint Paths on split graphs consists of four reduction rules. We fix a split partition (C^*, I^*) of the split graph G^* that forms part of the instance on which the kernelization algorithm is applied. In each of the rules below, we write $(G, \mathcal{X}, k)$ to denote the instance of Vertex-Disjoint Paths on which the rule is applied, and (C, I) denotes the unique split partition of G satisfying $I = V(G) \cap I^*$; from the description of the rules it will be clear that G indeed has such a split partition. The instance that is obtained after the application of the rule on $(G, \mathcal{X}, k)$ is denoted by $(G', \mathcal{X}', k')$. We say that a reduction rule is *safe* if $(G, \mathcal{X}, k)$ and $(G', \mathcal{X}', k')$ are equivalent instances of Vertex-Disjoint Paths.

Given an instance $(G, \mathcal{X}, k)$ of Vertex-Disjoint Paths and a split partition (C, I) of G, we can construct an auxiliary bipartite graph H as follows. Let T be the set of all terminal vertices in G. The vertex set of H consists of the independent set $I \setminus T$ and an independent set A that is constructed as follows. For every pair of vertices $v, w \in C$ such that vw is a heavy edge of weight $s \geq 2$, we add $s - 1$ vertices $a_1^{vw}, \ldots, a_{s-1}^{vw}$. The edge set of H is constructed by adding, for each $x \in I \setminus T$, an edge from x to vertices $a_1^{vw}, \ldots, a_{s-1}^{vw}$ if and only if x is adjacent to both v and w in G. An example is given in Figure 2.

Using graph H, we can now define our first reduction rule. Here, given a matching M of H, we say that $x \in I$ is *matched* by M if x is an endpoint of an edge in M.

Rule 1. *If there exists a non-terminal vertex in I, then we construct the bipartite graph H as described above, and find a maximum matching M in H. Let R be the set of vertices in $I \setminus T$ that are not matched by M. We set $G' = G - R$, $\mathcal{X}' = \mathcal{X}$, and $k' = k$.*

Lemma 3. *Rule 1 is safe.*

Proof. It is clear that if $(G', \mathcal{X}', k')$ is a yes-instance of VERTEX-DISJOINT PATHS, then $(G, \mathcal{X}, k)$ is also a yes-instance of VERTEX-DISJOINT PATHS, as G is a supergraph of G'. For the reverse direction, suppose $(G, \mathcal{X}, k)$ is a yes-instance of VERTEX-DISJOINT PATHS. Among all minimum solutions for this instance, let $\mathcal{P}$ be one for which the total number of vertices in R visited by the paths in $\mathcal{P}$ is as small as possible. We will show that $\mathcal{P}$ is a solution for $(G', \mathcal{X}', k')$, which suffices to prove the lemma.

The idea of the proof will be to show that $\mathcal{P}$ induces a matching M' in H. If a path in $\mathcal{P}$ uses a vertex in R, then by inspecting the symmetric difference of M and M', we can reroute some paths in $\mathcal{P}$ to avoid this vertex; in particular, we can reduce the total number of vertices of R that are visited by the paths in $\mathcal{P}$. Therefore, no path of $\mathcal{P}$ visits a vertex of R, proving that $\mathcal{P}$ is a solution for $(G', \mathcal{X}', k')$.

We first define a matching M' in H as follows. Consider any path $P_i \in \mathcal{P}$ that visits a vertex $x \in I \setminus T$. Since $x \notin T$, there are two vertices $v, w \in C$ such that the edges vx and xw appear consecutively on the path P_i. Hence, P_i is not induced. By Lemma 2, P_i has length 2, and its endpoints are v and w, which are also the endpoints of some path $P_j \in \mathcal{P}$ of length 1. In particular, this implies that vw is a heavy edge. Let $s \geq 2$ be the weight of vw, and consider the vertices $a_1^{vw}, \ldots, a_{s-1}^{vw}$ in the graph H. By construction, x is adjacent to each of these vertices. We arbitrarily choose an edge xa_j^{vw}, where $j \in \{1, \ldots, s-1\}$. We do the same for every other non-induced path in $\mathcal{P}$ whose middle vertex belongs to $I \setminus T$. By the construction of H, we can do this in such a way that the chosen edges form a matching. Let M' be such a matching. See Figure 2 for an illustration of such a matching M'.

Now suppose, for contradiction, that there exists a path $P_i \in \mathcal{P}$ that visits a vertex $x \in R$. Let v and w be the endpoints of P_i. As argued before, it follows from Lemma 2 that $P_i = vxw$. Hence, there exists an edge $xa_j^{vw} \in M'$ with $j \in \{1, \ldots, s-1\}$, where s is the weight of the edge vw. Let Q be a maximal path in H starting in x whose edges alternatingly belong to M' and M. Since no edge in M is incident with a vertex in R, we know that x is the only vertex of Q that belongs to R. Let $y \neq x$ be the other endpoint of Q. The last edge of Q, i.e., the edge of Q incident with y, belongs to M, as otherwise Q would be an M-augmenting path in H, contradicting the fact that M is a maximum matching in H. In particular, this implies that $y \in I \setminus T$ and y is not incident with an edge in M'. By the definition of M', we know that y is not visited by any non-induced path of $\mathcal{P}$, and thus y is not visited by any path in $\mathcal{P}$.

Let $z_1, \ldots, z_\ell$ be the consecutive vertices of Q that belong to $I \setminus T$, such that $z_1 = x$ and $z_\ell = y$. For every $j \in \{1, \ldots, \ell - 1\}$, we replace the path $P \in \mathcal{P}$ whose middle vertex is z_j by the path obtained from P by replacing z_j by z_{j+1}. Since y is not visited by any path in $\mathcal{P}$, this yields a new solution $\mathcal{P}'$. In fact, $\mathcal{P}'$ is a minimum solution, as the total number of edges in all the paths did not change. Since the paths in $\mathcal{P}'$ visit one fewer vertex of R than the paths in $\mathcal{P}$, we arrive at the desired contradiction. We conclude that no path in $\mathcal{P}$ visits a vertex in R. Therefore, $\mathcal{P}$ is a solution for $(G', \mathcal{X}', k')$. $\square$

Rule 2. *If there exists a non-terminal vertex $v \in I$ that is adjacent to a non-terminal vertex in C, then we set G' to be the graph obtained from G by deleting edge vw for every non-terminal vertex $w \in C$. We also set $\mathcal{X}' = \mathcal{X}$ and $k' = k$.*

Lemma 4. *Rule 2 is safe.*

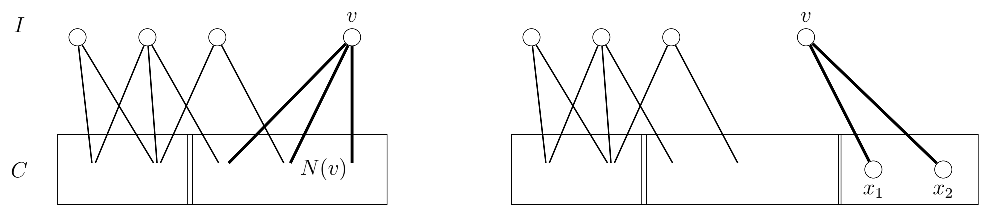


Fig. 3. Illustration of Rule 3. The left picture shows a split graph G with a split partition (C, I). Consider an instance on G with $k = 2$ such that v is a terminal of both pairs. Then $p = 2$. Since $d_G(v) = 3 \geq 2 = 2k - p$, we can indeed apply Rule 3 to v. The right picture shows the graph after the application of Rule 3.

Proof. Suppose $v \in I$ is a non-terminal vertex that is adjacent to a non-terminal vertex $w \in C$. If $(G, \mathcal{X}, k)$ is a yes-instance and $\mathcal{P}$ is a minimum solution for this instance, then no path in $\mathcal{P}$ uses the edge vw as a result of Lemma 2. Hence, it is safe to delete any such edge vw from the graph. $\square$

Rule 3. *If there exists a terminal vertex $v \in I$ with $d_G(v) \geq 2k-p$, where $p \geq 1$ is the number of terminals on v, then we set G' to be the graph obtained from G by deleting all edges incident with v, adding p new vertices $\{x_1, \ldots, x_p\}$, and making these new vertices adjacent to v, to each other, and to all the other vertices in C. We also set $\mathcal{X}' = \mathcal{X}$ and $k' = k$.*

See Figure 3 for an illustration of this rule.

Lemma 5. *Rule 3 is safe.*

Proof. Suppose there exists a terminal vertex $v \in I$ with $d_G(v) \geq 2k - p$, where $p \geq 1$ is the number of terminals on v. Let $X = \{x_1, \ldots, x_p\}$ be the set of vertices that were added during the execution of the rule. Let $Y = \{y_1, \ldots, y_p\}$ be the set of terminals on v.

Suppose $(G, \mathcal{X}, k)$ is a yes-instance of Vertex-Disjoint Paths, and let $\mathcal{P} = \{P_1, \ldots, P_k\}$ be an arbitrary solution for this instance. We construct a solution $\mathcal{P}' = \{P'_1, \ldots, P'_k\}$ for $(G', \mathcal{X}', k')$ as follows. Let $i \in \{1, \ldots, k\}$. First suppose that neither of the terminals in the pair (s_i, t_i) belongs to the set Y. Since the paths in $\mathcal{P}$ are pairwise vertex-disjoint and v is a terminal vertex, the path P_i does not contain an edge incident with v. Hence, P_i exists in G', and we set $P'_i = P_i$. Now suppose $v \in \{s_i, t_i\}$, and recall that $s_i \neq t_i$ by definition. Suppose, without loss of generality, that $v = s_i$. Then $s_i \in Y$, so $s_i = y_r$ for some $r \in \{1, \ldots, p\}$. Let vw be the first edge of the path P_i in G. We define P'_i to be the path in G' obtained from P_i by deleting the edge vw and adding the vertex x_r as well as the edges vx_r and x_rw. Let $\mathcal{P}' = \{P'_1, \ldots, P'_k\}$ denote the collection of paths in G' obtained this way. Since the paths in $\mathcal{P}$ are pairwise vertex-disjoint in G, and every vertex in $\{x_1, \ldots, x_p\}$ is visited by exactly one path in $\mathcal{P}'$, it holds that the paths in $\mathcal{P}'$ are pairwise vertex-disjoint in G'. Hence, $\mathcal{P}'$ is a solution for the instance $(G', \mathcal{X}', k')$.

For the reverse direction, suppose $(G', \mathcal{X}', k')$ is a yes-instance of Vertex-Disjoint Paths, and let $\mathcal{Q} = \{Q_1, \ldots, Q_k\}$ be a minimum solution. Let $\mathcal{Q}^* \subseteq \mathcal{Q}$ be the set of paths in $\mathcal{Q}$ that visit a vertex in the set $X = \{x_1, \ldots, x_p\}$. Since there are p terminals on v, and v has exactly p neighbors in G' (namely, the vertices of X), every path in $\mathcal{Q}^*$ has v as one of its endpoints and $|\mathcal{Q}^*| = p$. Now consider the paths in $\mathcal{Q} \setminus \mathcal{Q}^*$. Due to Lemma 2, each such path visits at most three vertices of $N_G(v)$. If a path in $\mathcal{Q} \setminus \mathcal{Q}^*$ is induced, it visits at most two vertices of $N_G(v)$. If a path $Q_i \in \mathcal{Q} \setminus \mathcal{Q}^*$ visits three vertices in $N_G(v)$, then Lemma 2 ensures that there is another path $Q_j \in \mathcal{Q} \setminus \mathcal{Q}^*$ such that Q_j has length 1 and the endpoints of Q_j coincide with the endpoints of Q_i, implying that Q_i and Q_j together visit at most three vertices of $N_G(v)$. We deduce that at most $2(k-p)$ vertices of $N_G(v)$ are visited by the $k-p$ paths in $\mathcal{Q} \setminus \mathcal{Q}^*$. Recall that $d_G(v) \geq 2k-p$. Therefore, at least p vertices of $N_G(v)$, say $z_1, \ldots, z_p$, are not visited by any path in $\mathcal{Q} \setminus \mathcal{Q}^*$.

Armed with the above observations, we construct solution $\mathcal{P} = (P_1, \ldots, P_k)$ for $(G, \mathcal{X}, k)$ as follows. For every path $Q_i \in \mathcal{Q} \setminus \mathcal{Q}^*$, we define $P_i = Q_i$. Now let $Q_i \in \mathcal{Q}^*$. The path Q_i visits v, one vertex $x_\ell \in X$, one vertex $z \in C$, and possibly one (terminal) vertex in $I \setminus \{v\}$. If $z \in N_G(v)$, then we define P_i to be the path in G obtained from Q_i by deleting x_ℓ and its two incident edges and adding the edge vz. If $z \notin N_G(v)$, then we define P_i to be the path obtained from Q_i by replacing the vertex x_ℓ by z_ℓ. It is easy to verify that $\mathcal{P}$ is a solution for the instance $(G, \mathcal{X}, k)$. $\square$

Rule 4. *If there exist more than $k-1$ non-terminal vertices in C that have no neighbors in I, then we set G' to be the graph obtained from G by deleting all but $k-1$ of those vertices. We also set $\mathcal{X}' = \mathcal{X}$ and $k' = k$.*

Lemma 6. *Rule 4 is safe.*

Proof. As in the proof of Lemma 3, the fact that G is a supergraph of G' implies that $(G, \mathcal{X}, k)$ is a yes-instance of Vertex-Disjoint Paths if $(G', \mathcal{X}', k')$ is. For the reverse direction, suppose $(G, \mathcal{X}, k)$ is a yes-instance, and let $\mathcal{P}$ be a minimum solution. Let v be a non-terminal vertex in C that has no neighbors in I. Due to Lemma 2, the only kind of path in $\mathcal{P}$ that can visit v is a path of length 2 whose endpoints are the endpoints of a heavy edge. Observe that there are at most $k-1$ such paths, as for every heavy edge pq in G, there is exactly one path in $\mathcal{P}$ of length 1 whose endpoints are p and q. Since all non-terminals in C that have no neighbors in I are true twins, we can safely delete all but $k-1$ of them from the graph. $\square$

We now prove that the above four reduction rules yield a quadratic vertex kernel for Vertex-Disjoint Paths on split graphs.

Theorem 5. *The* Vertex-Disjoint Paths *problem on split graphs has a kernel with at most $4k^2$ vertices and (thus) at most $8k^4$ edges, where k is the number of terminal pairs.*

Proof. We describe a kernelization algorithm for Vertex-Disjoint Paths on split graphs. Let $(G^*, \mathcal{X}, k)$ be an instance of Vertex-Disjoint Paths, where G^* is a split graph. We fix a split partition (C^*, I^*) of G^*. We start by applying Rule 1 once. We then exhaustively apply Rules 2 and 3. Finally, we apply Rule 4 once. Let $(G', \mathcal{X}', k')$ be the obtained instance. From the description of the reduction rules it is clear that G' is a split graph, and that $\mathcal{X}' = \mathcal{X}$ and $k' = k$. By Lemmas 3–6, $(G', \mathcal{X}', k')$ is a yes-instance of Vertex-Disjoint Paths if and only if $(G^*, \mathcal{X}, k)$ is a yes-instance. Hence, the algorithm indeed reduces any instance of Vertex-Disjoint Paths to an equivalent instance.

We now show that $|V(G')| \leq 4k^2$. Let (C', I') be the unique split partition of G' such that $I' = V(G') \cap I^*$, i.e., the independent set I' contains exactly those vertices of I^* that were not deleted during any application of the reduction rules. Note that this split partition indeed exists by the description of the reduction rules. We distinguish two cases.

First suppose that every vertex in I' is a terminal vertex. Then $|I'| \leq 2k$, and since Rule 3 cannot be applied, every vertex in I' has degree at most $2k-2$. This implies that C' contains at most $2k(2k-2)$ vertices that have at least one neighbor in I'. Since Rule 4 cannot be applied, there are at most $k-1$ vertices in C' that have no neighbor in I', and thus $|C'| \leq 4k^2 - 3k - 1$. This implies that $|V(G')| = |I'| + |C'| \leq 4k^2$, as desired.

Now suppose that I' contains at least one non-terminal vertex. Recall the graph $H = (I^* \setminus T, A, F)$ and the maximum matching M in H that are constructed during the execution of Rule 1. The assumption that I' contains a non-terminal vertex implies that $|M| \geq 1$. Each edge in M corresponds to a non-terminal vertex in I' and a terminal pair in C', and there is at least one heavy edge whose endpoints both belong to C'. This implies that I' contains $|M|$ non-terminal vertices, and there are at least $|M|+1$ terminal pairs that belong to C'. Hence, there are at most $2(k-(|M|+1))$ terminal vertices in I', and $|I'| \leq |M|+2(k-(|M|+1)) \leq 2k-|M|-2 \leq 2k-3$, where the last inequality follows from the assumption that $|M| \geq 1$. Since Rules 2 and 3 cannot be applied, every vertex in I' has degree at most $2k$. This implies that C' contains at most $2k(2k-3)$ vertices that have at least one neighbor in I', and C' contains at most $k-1$ other vertices due to Rule 4. We conclude that $|C'| \leq 4k^2-5k-1$, and thus $|V(G')| \leq 4k^2$.

It remains to argue that the above algorithm runs in polynomial time. Recall that a split partition of G^* can be found in linear time [12]. Rules 1 and 4 are applied only once. Rules 2 and 3 together are applied at most $|I^*|$ times in total, as each vertex in I^* that is not deleted during the execution of Rule 1 is considered in at most one of these rules. Rules 2, 3, and 4 can trivially be executed in polynomial time. The same holds for Rule 1, since it only takes polynomial time to construct the auxiliary bipartite graph H and find a maximum matching M in H. We conclude that the overall running time of the kernelization algorithm is polynomial. ◻

Note that because the number of vertices of the clique in the reduced graph G' contributes the most to the total number of vertices in G', we cannot prove a significantly better bound on the number of edges of G' than the trivial bound: $8k^4$. This means that the actual size of the kernel will be $O(k^4)$, even though the number of vertices is significantly smaller.

4.2 Polynomial Kernel for Edge-Disjoint Paths on Split Graphs

In this section, we present a cubic vertex kernel for the Edge-Disjoint Paths problem on split graphs. We need the following two structural lemmas.

Lemma 7. *Let $(G,\mathcal{X},k)$ be an instance of* Edge-Disjoint Paths *such that G is a complete graph. If $|V(G)| \geq 2k$, then $(G,\mathcal{X},k)$ is a yes-instance.*

Proof. An edge $vw \in E(G)$ is called *occupied by the terminal pair* (s_i,t_i) if there is a terminal pair (s_i,t_i) such that $\{v,w\} = \{s_i,t_i\}$; if vw is occupied by some terminal pair, we simply call it *occupied.* Recall that an edge vw is *heavy* if it is occupied by more than one terminal pair. If vw is occupied by exactly one terminal pair, then we call vw a *light* edge. Denote the number of light edges by ℓ_1 and the number of heavy edges by ℓ_2. Observe that $\ell_1+2\ell_2 \leq k$.

We claim that there exists a solution $\mathcal{P} = \{P_1,\ldots,P_k\}$ for the instance $(G,\mathcal{X},k)$ such that every path in $\mathcal{P}$ contains at most three vertices, and we will construct such a solution below.

For every terminal pair $(s_i,t_i) \in \mathcal{X}$ that occupies a light edge, we define P_i to be the path whose only edge is s_it_i. For every heavy edge vw, we arbitrarily choose a terminal pair (s_i,t_i) that occupies it, and again define P_i to be the path whose only edge is $vw = s_it_i$. Let $\mathcal{P}'$ be the set of paths P_i that we have defined so far, and let $\mathcal{X}'$ consist of the corresponding terminal pairs in $\mathcal{X}$. Note that every path in $\mathcal{P}'$ contains exactly two vertices. Every path in $\mathcal{P} \setminus \mathcal{P}'$ will contain exactly three vertices. For every terminal pair $(s_i,t_i) \in \mathcal{X} \setminus \mathcal{X}'$, the middle vertex of the path $P_i \in \mathcal{P} \setminus \mathcal{P}'$ is called the *bouncer* for (s_i,t_i). We now describe how we can construct the set of all bouncers.

For each terminal pair $(s_i,t_i) \in \mathcal{X} \setminus \mathcal{X}'$, we choose an arbitrary vertex of $V(G) \setminus \{s_i,t_i\}$ that is not incident with any occupied edge as the bouncer for (s_i,t_i), and we do this in such a way that the bouncers for any two pairs in $\mathcal{X} \setminus \mathcal{X}'$ are distinct. To see why this is possible, we first observe that we need to choose exactly $|\mathcal{X} \setminus \mathcal{X}'| = k-\ell_1-\ell_2$ bouncers. Since there are $\ell_1 + \ell_2$ occupied edges in G, there are at least $2k - 2(\ell_1 + \ell_2) = 2(k - \ell_1 - \ell_2)$ vertices of G that are not incident with any occupied edge. Recall that $\ell_1 + 2\ell_2 \leq k$, which implies that $k - \ell_1 - \ell_2 \geq \ell_2 \geq 0$ and thus $2(k - \ell_1 - \ell_2) \geq k - \ell_1 - \ell_2$. This means that we can indeed choose a unique bouncer for each terminal pair in $\mathcal{X} \setminus \mathcal{X}'$ in the way described above.

Now, for each terminal pair $(s_i, t_i) \in \mathcal{X} \setminus \mathcal{X}'$, let P_i be the path from s_i to its selected bouncer to t_i. Note that the paths in $\mathcal{P} = (P_1, \ldots, P_k)$ are pairwise edge-disjoint due the way we chose the bouncers, implying that $(G, \mathcal{X}, k)$ is a yes-instance. $\square$

Lemma 8. *Let $(G, \mathcal{X}, k)$ be an instance of* Edge-Disjoint Paths *such that G is a split graph with split partition (C, I), and the degree of each terminal vertex is at least the number of terminals on it. If $|C| \geq 2k$, then $(G, \mathcal{X}, k)$ is a yes-instance.*

Proof. The proof of this lemma consists of two steps: project to C, and route within C. In the first step, we project the terminals to C. Consider any terminal vertex $x \in I$. For each terminal on x, we project it to a neighbor of x in such a way that no two terminals on x are projected to the same vertex; if the terminal is s_i, denote this neighbor by s'_i, and if the terminal is t_i, denote this neighbor by t'_i. Since the degree of every terminal vertex is at least the number of terminals on it, this is indeed possible. For any terminal s_i that is on a terminal vertex in C, let $s'_i = s_i$, and for any terminal t_i that is on a terminal vertex in C, let $t'_i = t_i$. Let $\mathcal{X}' = \{(s'_i, t'_i) \mid i = 1, \ldots, k\}$, and let $G' = G - I$.

Since G' is a complete graph and $|V(G')| = |C| \geq 2k$, there exists a solution $\mathcal{P}''$ for the instance $(G', \mathcal{X}'', k'')$ due to Lemma 7, where $\mathcal{X}'' = \{(s'_i, t'_i) \in \mathcal{X}' \mid s'_i \neq t'_i\}$ and $k'' = |\mathcal{X}''|$. Let $\mathcal{P}' = \{P'_1, \ldots, P'_k\}$ be the collection of paths obtained from $\mathcal{P}''$ as follows. For every pair $(s'_i, t'_i) \in \mathcal{X}''$, we define P'_i to be the path in $\mathcal{P}''$ connecting s'_i and t'_i. For every pair (s'_i, t'_i) with $s'_i = t'_i$, we define P'_i to be the trivial path consisting of the single vertex s'_i. We now show that we can extend the paths in $\mathcal{P}'$ to obtain a solution $\mathcal{P}$ for the instance $(G, \mathcal{X}, k)$. For every $i \in \{1, \ldots, k\}$, we extend the path P'_i using the edges $s_i s'_i$ (if $s_i \neq s'_i$) and $t_i t'_i$ (if $t_i \neq t'_i$); let the resulting path be P_i. Since for every terminal vertex $x \in I$, no two terminals on x were projected to the same neighbor of x, the paths in $\mathcal{P}$ are pairwise edge-disjoint. We conclude that $(G, \mathcal{X}, k)$ is a yes-instance. $\square$

Apart from the above two lemmas, our kernelization algorithm for Edge-Disjoint Paths on split graphs will use one reduction rule. Before formulating this rule, we first define an auxiliary bipartite graph H' that is similar to but different from the graph H used in Rule 1 in the previous subsection. Given an instance $(G, \mathcal{X}, k)$ of Edge-Disjoint Paths and a split partition (C, I) of G, we define H' to be the bipartite graph whose vertex set consists of the independent set $I \setminus T$ and an independent set B that contains $4k + 1$ vertices $b_1^{vw}, \ldots, b_{4k+1}^{vw}$ for each pair v, w of vertices of C. For each $x \in I \setminus T$, we add edges from x to all of the vertices $b_1^{vw}, \ldots, b_{4k+1}^{vw}$ if and only if x is adjacent to both v and w in G. An example is given in Figure 4.

We are now ready to formulate the reduction rule. The rule takes as input an instance $(G, \mathcal{X}, k)$ of Edge-Disjoint Paths and a split partition (C, I) of G, and it returns an instance $(G', \mathcal{X}', k')$ of Edge-Disjoint Paths.

Rule A. *If there exists a non-terminal vertex in I, then we construct the bipartite graph H' as described above, and find a maximal matching M in H'. Let R be the set of vertices in $I \setminus T$ that are not matched by M. We set $G' = G - R$, $\mathcal{X}' = \mathcal{X}$, and $k' = k$.*

Lemma 9. *Rule A is safe.*

Proof. It is clear that if $(G', \mathcal{X}', k')$ is a yes-instance of Edge-Disjoint Paths, then $(G, \mathcal{X}, k)$ is also a yes-instance of Edge-Disjoint Paths, as G is a supergraph of G'. For the reverse

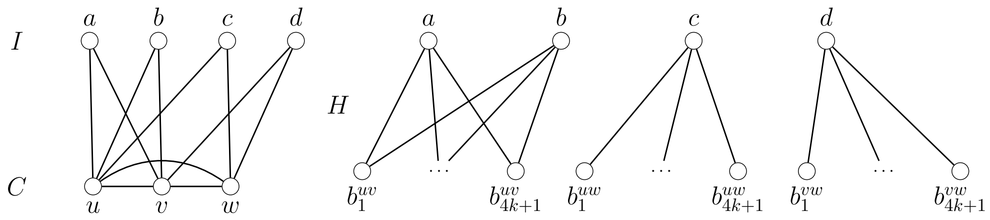


Fig. 4. The left picture shows a split graph G with split partition (C, I). Consider any instance on G such that no vertex of I is a terminal vertex. The right figure then shows the corresponding auxiliary bipartite graph H.

direction, suppose that $(G, \mathcal{X}, k)$ is a yes-instance of Edge-Disjoint Paths. Note that there exists a solution for $(G, \mathcal{X}, k)$ such that no path in the solution visits a vertex more than once. Among all such solutions, let $\mathcal{P} = (P_1, \ldots, P_k)$ be one for which the total number of visits by all paths combined to vertices from R is minimized. We claim that no path in $\mathcal{P}$ visits a vertex in R.

For contradiction, suppose that some path $P_j \in \mathcal{P}$ visits some vertex $r \in R$. Since $r \notin T$, there are two vertices $v, w \in C$ such that the edges vr and wr appear consecutively on the path P_j. As $r \in R$, it is not matched by the maximal matching M used in Rule A. Since r is adjacent to all the vertices in $\{b_1^{vw}, \ldots, b_{4k+1}^{vw}\}$ and M is a maximal matching, all the vertices in $\{b_1^{vw}, \ldots, b_{4k+1}^{vw}\}$ are matched by M, and consequently at least $4k + 1$ vertices of $I \setminus T$ that are adjacent to both v and w are matched by M. Let Z denote this set of vertices. By the choice of $\mathcal{P}$, no path of $\mathcal{P}$ visits a vertex twice. Hence, there are at most $4k$ edges of $\bigcup_{i=1}^{k} E(P_i)$ incident with v or w in G. Therefore, there exists a vertex $z \in Z$ such that $\bigcup_{i=1}^{k} E(P_i)$ contains neither the edge vz nor the edge wz. Let P_j' be the path obtained from P_j by replacing r with z and shortcutting it if necessary (i.e., if $z \in V(P_j)$). Then, $\mathcal{P}' = (P_1, \ldots, P_{j-1}, P_j', P_{j+1}, \ldots, P_k)$ is a solution for $(G, \mathcal{X}, k)$ where each path visits each vertex at most once, and where the total number of visits by all paths combined to vertices from R is at least one smaller than $\mathcal{P}$, contradicting the choice of $\mathcal{P}$. Therefore, no path of $\mathcal{P}$ visits a vertex of R. Hence, $\mathcal{P}$ is also a solution for $(G', \mathcal{X}', k')$, and thus it is a yes-instance. □

We now present our second kernelization result.

Theorem 6. *The* Edge-Disjoint Paths *problem on split graphs has a kernel with at most* $8k^3$ *vertices and at most* $16k^4$ *edges, where k is the number of terminal pairs.*

Proof. We describe a kernelization algorithm for Edge-Disjoint Paths on split graphs. Let $(G, \mathcal{X}, k)$ be an instance of Edge-Disjoint Paths, where G is a split graph on n vertices and m edges. We assume that $k \geq 1$, and we fix a split partition (C, I) of G. If the degree of any terminal vertex is less than the number of terminals on it, then $(G, \mathcal{X}, k)$ is clearly a no-instance, and the kernelization algorithm outputs a trivial no-instance. Hence, we may assume that the degree of any terminal vertex is at least the number of terminals on it. If $|C| \geq 2k$, then the algorithm returns a trivial yes-instance, which is safe due to Lemma 8. Suppose $|C| \leq 2k - 1$. If every vertex in I is a terminal vertex, then $|I| \leq 2k$ and thus $|V(G)| \leq 4k - 1$ and $|E(G)| \leq 8k^2$, so we simply return the current instance as the desired kernel.

Suppose I contains at least one non-terminal vertex. We apply Rule A. Let $(G', \mathcal{X}', k')$ denote the resulting instance. By Lemma 9, $(G', \mathcal{X}', k')$ is a yes-instance of Edge-Disjoint Paths if and only if $(G, \mathcal{X}, k)$ is a yes-instance. Let (C', I') be the split partition of $V(G')$

such that $C' \subseteq C$ and $I' \subseteq I$. Since Rule A does not change the clique part of the instance, $C = C'$, and in particular, $|C'| = |C| \leq 2k-1$ by assumption. By the construction of H', any maximal matching of H' has size at most $|B| = (4k+1) \cdot \binom{|C|}{2} \leq 8k^3 - 10k^2 + k + 1 \leq 8k^3 - 4k$. Hence, I' contains at most $8k^3 - 4k$ non-terminal vertices, and consequently $|I'| \leq 8k^3 - 2k$. We conclude that $|V(G')| = |C'| + |I'| \leq 8k^3$. To bound $|E(G')|$, we recall that C' is a clique and I' is an independent set. Therefore, $|E(G')| \leq \binom{|C'|}{2} + |C'| \cdot |I'| \leq 16k^4$.

It remains to argue that the above algorithm runs in polynomial time. Recall that a split partition of G can be found in linear time [12]. Checking the degrees of terminal vertices can be done in linear time. Therefore, it remains to analyze the time needed to apply Rule A. Note that H' has $O(n + k \cdot \binom{|C|}{2}) = O(n + k^3)$ vertices, and we can add all edges in $O(k \cdot \binom{|C|}{2} n) = O(k^3 n)$ time. Since finding a maximal matching takes linear time, the overall running time is indeed polynomial. ◻

Note that, as the number of edges of the kernel is at most $16k^4$, the actual size of the kernel will be $O(k^4)$, even though the number of vertices is significantly smaller.

5 Conclusion

We proved that Vertex-Disjoint Paths and Edge-Disjoint Paths admit kernels with $O(k^2)$ and $O(k^3)$ vertices, respectively, when restricted to split graphs. It would be interesting to investigate whether or not the problems admit *linear* vertex kernels on split graphs. Another interesting open question is whether either problem admits a polynomial kernel on chordal graphs, a well-known superclass of split graphs.

Recall that Bodlaender et al. [3] proved that Vertex-Disjoint Paths does not admit a polynomial kernel on general graphs, unless NP $\subseteq$ coNP/poly. They asked whether or not the problem admits a polynomial kernel when restricted to planar graphs. One could ask the same question about the Edge-Disjoint Paths problem. As far as we know, even the kernelization complexity of Edge-Disjoint Paths on general graphs is still open.

References

1. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. J. Comp. Syst. Sci. 75(8), 423–434 (2009)
2. Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (Meta) kernelization. In: 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, pp. 629–638. IEEE Computer Society (2009)
3. Bodlaender, H.L., Thomasse, S., Yeo, A.: Kernel bounds for disjoint cycles and disjoint paths. Theor. Comp. Sci. 412(35), 4570–4578 (2011)
4. Brandstädt, A., Le, V. B., Spinrad, J. Graph Classes: A Survey. SIAM Monographs on Discrete Mathematics and Applications (1999)
5. Diestel, R.: Graph Theory. Springer-Verlag, Electronic Edition (2005)
6. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science, Springer (1999)
7. Dvorak, Z., Král', D., Thomas, R.: Three-coloring triangle-free planar graphs in linear time. In: Mathieu, C. (ed.), SODA 2009, pp. 1176–1182. ACM-SIAM (2009)
8. Even, S., Itai, A., Shamir, A.: On the complexity of timetable and multicommodity flow problems. SIAM J. Comp. 5, 691–703 (1976)
9. Frank, A.: Packing paths, circuits, and cuts – a survey. In: Korte, B., Lovász, L., Prömel, H.J., Schrijver, A. (eds.), Paths, Flows, and VLSI-Layout, pp. 47–100. Springer-Verlag, Berlin (1990)
10. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Annals of Disc. Math. 57, Elsevier (2004).

11. Gurski, F., Wanke, E.: Vertex disjoint paths on clique-width bounded graphs. Theor. Comput. Sci. 359, 188–199 (2006)
12. Hammer, P.L., Simeone, B.: The splittance of a graph. Combinatorica 1, 275–284 (1981)
13. Heggernes, P., van 't Hof, P., van Leeuwen, E.J., Saei, R.: Finding disjoint paths in split graphs. In: Geffert, V., Preneel, B., Rovan, B., Stuller, J., Tjoa, A.M. (eds.), SOFSEM 2014. LNCS, vol. 8327, pp. 315–326 (2014)
14. Kammer, F., Tholey, T.: The k-disjoint paths problem on chordal graphs. In: Paul, C., Habib, M. (eds.), WG 2009. LNCS, vol. 5911, pp. 190–201 (2010)
15. Karp, R.M.: On the complexity of combinatorial problems. Networks 5, 45–68 (1975)
16. Kawarabayashi, K., Kobayashi, Y., Reed, B.A.: The disjoint paths problem in quadratic time. J. Comb. Theory B 102, 424–435 (2012)
17. Kobayashi, Y., Kawarabayashi, K.: Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In: Mathieu, C. (ed.), SODA 2009, pp. 1146–1155. ACM-SIAM (2009)
18. Kramer, M., van Leeuwen, J.: The complexity of wirerouting and finding minimum area layouts for arbitrary VLSI circuits. Adv. Comput. Res. 2, 129–146 (1984)
19. Lynch, J.F.: The equivalence of theorem proving and the interconnection problem ACM SIGDA Newsletter 5(3), 31–36 (1975)
20. Natarajan, S., Sprague, A.P.: Disjoint paths in circular arc graphs. Nordic J. Comp. 3, 256–270 (1996)
21. Nishizeki, T., Vygen, J., Zhou, X.: The edge-disjoint paths problem is NP-complete for series-parallel graphs. Discrete Applied Math. 115, 177–186 (2001)
22. Reed, B.A.: Tree width and tangles: A new connectivity measure and some applications. In: R.A. Bailey (ed.), Surveys in Combinatorics, pp. 87–162. Cambridge University Press (1997)
23. Reed, B.A., Robertson, N., Schrijver, A., Seymour, P.D.: Finding disjoint trees in planar graphs in linear time. In: Contemp. Math., vol. 147, pp. 295–301. Amer. Math. Soc., Providence, RI (1993)
24. Robertson, N., Seymour, P.D.: Graph minors XIII. The disjoint paths problem. J. Comb. Theory B 63(1), 65–110 (1995)
25. Tovey, C.A.: A simplified NP-complete satisfiability problem. Discrete Applied Math. 8, 85–89 (1984)
26. Vygen, J.: Disjoint paths. Technical report 94816, Research Institute for Discrete Mathematics, University of Bonn (1998)

Chapter 10

Subset Feedback Vertex Sets in Chordal Graphs

Petr A. Golovach, Pinar Heggernes, Dieter Kratsch, and Reza Saei

J. Discrete Algorithms 26 (2014), 7-15.
(A preliminary version appeared in IPEC 2012, LNCS 7535: 85 - 96, Springer.)

Subset Feedback Vertex Sets in Chordal Graphs*

Petr A. Golovach[1], Pinar Heggernes[1], Dieter Kratsch[2], and Reza Saei[1]

[1] Department of Informatics, University of Bergen, Norway,
{petr.golovach,pinar.heggernes,reza.saeidinvar}@ii.uib.no
[2] LITA, Université de Lorraine, Metz, France, kratsch@univ-metz.fr

Abstract. Given a graph $G = (V, E)$ and a set $S \subseteq V$, a set $U \subseteq V$ is a subset feedback vertex set of (G, S) if no cycle in $G[V \setminus U]$ contains a vertex of S. The Subset Feedback Vertex Set problem takes as input G, S, and an integer k, and the question is whether (G, S) has a subset feedback vertex set of cardinality or weight at most k. Both the weighted and the unweighted versions of this problem are NP-complete on chordal graphs, even on their subclass split graphs. We give an algorithm with running time $O(1.6708^n)$ that enumerates all minimal subset feedback vertex sets on chordal graphs on n vertices. As a consequence, Subset Feedback Vertex Set can be solved in time $O(1.6708^n)$ on chordal graphs, both in the weighted and in the unweighted case. As a comparison, on arbitrary graphs the fastest known algorithm for these problems has $O(1.8638^n)$ running time. We also obtain that a chordal graph G has at most 1.6708^n minimal subset feedback vertex sets, regardless of S. This narrows the gap with respect to the best known lower bound of 1.5848^n on this graph class. For arbitrary graphs, the gap is substantially wider, as the best known upper and lower bounds are 1.8638^n and 1.5927^n, respectively.

1 Introduction

Given a graph $G = (V, E)$ and a set $S \subseteq V$, a set $U \subseteq V$ is a subset feedback vertex set of (G, S) if no cycle in $G[V \setminus U]$ contains a vertex of S. A subset feedback vertex set U is minimal if no subset feedback vertex set of (G, S) is a proper subset of U. The Subset Feedback Vertex Set problem takes as input G, S, and an integer k, and the question is whether (G, S) has a subset feedback vertex set of cardinality at most k. In the weighted version of the problem, every vertex of G has a weight, and the question is whether there is a subset feedback vertex set of total weight at most k.

Subset Feedback Vertex Set was introduced by Even et al. [5], and it generalizes several well-studied problems. When $S = V$, it is equivalent to the classical Feedback Vertex Set problem [13], and when $|S| = 1$, it generalizes the Multiway Cut problem [8]. Weighted Subset Feedback Vertex Set admits a polynomial-time constant-factor approximation algorithm [5]. The unweighted version of the problem is fixed parameter tractable [3]. The only exact algorithm known for its weighted version is by Fomin et al. [8] and it runs in $O(1.8638^n)$ time and solves the problem by enumerating all minimal subset feedback vertex sets.

Let us briefly compare Subset Feedback Vertex Set to its more widely known restriction Feedback Vertex Set. The unweighted version of Feedback Vertex Set can be solved in time $O(1.7347^n)$ [10], whereas the best known algorithm for its weighted version runs in time $O(1.8638^n)$ and enumerates all minimal feedback vertex sets [6]. Feedback Vertex Set has also been studied on many graph classes, like chordal graphs and AT-free

*This work has been supported by the European Research Council, the Research Council of Norway, and the French National Research Agency.

graphs [1,18], and several positive results exist. This is not yet the case for Subset Feedback Vertex Set, and no algorithm with a running time of $O(c^n)$ such that $c < 1.8638$ is known for any significant graph class. Interestingly, whereas both the weighted and the unweighted versions of Feedback Vertex Set are solvable in polynomial time on chordal graphs [1,22], even the unweighted version of Subset Feedback Vertex Set is NP-complete on chordal graphs; in fact on their more restricted subclass split graphs, by a standard reduction from Vertex Cover [8].

In this paper we give an algorithm with running time $O(1.6708^n)$ that enumerates all minimal subset feedback vertex sets when the input graph is chordal. As a consequence, Subset Feedback Vertex Set can be solved in time $O(1.6708^n)$ on chordal graphs, both in the weighted and in the unweighted cases. Our algorithm differs completely from the $O(1.8638^n)$ time algorithm of [8] for the general case, and it heavily uses the structure of chordal graphs. Chordal graphs form one of the most studied graph classes; they have extensive practical applications in Sparse Matrix Computations [12], Computational Biology and Phylogenetics [21], and several other fields [15], and they are crucial in characterizing and understanding fundamental algorithmic tools, like treewidth.

Enumeration algorithms are central in the field of Exact Exponential Algorithms, as the running times of many exact exponential time algorithms rely on the maximum number of various objects in graphs [9]. A classical example is the widely used result of Moon and Moser [19], showing that the maximum number of maximal cliques or maximal independent sets in an n-vertex graph is $3^{n/3}$. More recently, the maximum numbers and enumeration of objects like minimal dominating sets, minimal feedback vertex sets, minimal subset feedback vertex sets, minimal separators, and potential maximal cliques, have been studied; see e.g., [6,7,8,10,16,17,20]. The maximum number of such objects in graphs have traditionally found independent interest also in graph theory and combinatorics.

The results we present in this paper give an upper bound of 1.6708^n on the maximum number of minimal subset feedback vertex sets a chordal graph can have. A tight bound on the maximum number of minimal feedback vertex sets on chordal graphs is known to be $10^{n/5} \approx 1.5848^n$ [2], and this thus gives a lower bound on the maximum number of minimal subset feedback vertex sets on chordal graphs. Consequently, our results tighten the gap between the upper and lower bounds on the maximum number of subset feedback vertex sets on chordal graphs. The corresponding gap is much larger on general graphs. There, the maximum numbers of minimal feedback and subset feedback vertex sets are both 1.8638^n [6,8], but no examples of graphs having 1.5927^n or more minimal feedback or subset feedback vertex sets are known [6]. Note that the maximum number of minimal subset feedback vertex sets can be dramatically different from the maximum number of minimal feedback vertex sets. Split graphs, which form a subclass of chordal graphs, have at most n^2 minimal feedback vertex sets, whereas they can have $3^{n/3} \approx 1.4422^n$ minimal subset feedback vertex sets [8]. These upper and lower bounds are summarized in Table 1.

In the next section we give the necessary background and notation. Our main results are presented in Section 3. We end the paper with a concluding section containing open questions.

2 Background and notation

We work with simple undirected graphs. We denote such a graph by $G = (V, E)$, where V is the set of vertices and E is the set of edges of G. We adhere to the convention that $n = |V|$. The set of neighbors of a vertex $v \in V$ is denoted by $N_G(v)$. The degree of v, $|N_G(v)|$, is denoted by $d_G(v)$. The *closed neighborhood* of v is $N_G[v] = N(v) \cup \{v\}$. We will omit the subscripts when there is no ambiguity. For a vertex subset $X \subseteq V$, the subgraph of G induced

	Max # subset feedback vertex sets		Max # feedback vertex sets	
Graph class	Upper bound	Lower bound	Upper bound	Lower bound
General	1.8638^n	1.5927^n	1.8638^n	1.5927^n
Chordal	1.6708^n	$10^{n/5} \approx 1.5848^n$	$10^{n/5}$	$10^{n/5}$
Split	1.6708^n	$3^{n/3} \approx 1.4422^n$	n^2	n^2

Table 1. Known upper and lower bounds on the maximum number of feedback vertex and subset feedback vertex sets a graph can have. The upper bounds for the maximum number of subset feedback vertex sets for chordal and split graphs are results of this paper.

by X is denoted by $G[X]$. For ease of notation, we use $G - v$ to denote the graph $G[V \setminus \{v\}]$, and $G - X$ to denote the graph $G[V \setminus X]$.

A *path* in G is a sequence of distinct vertices such that the next vertex in the sequence is adjacent to the previous vertex. A *cycle* is a path with at least three vertices such that the last vertex is in addition adjacent to the first. Given a subset $S \subseteq V$, we call a cycle an S-*cycle* if it contains a vertex of S. For a cycle or an S-cycle C, we use $V(C)$ to denote the set of vertices in C. A subset $F \subseteq V$ will be called a *forest* if $G[F]$ contains no cycle. Similarly, F is an S-forest if no cycle in $G[F]$ contains a vertex of S. A graph is *connected* if there is a path between every pair of its vertices. A maximal connected subgraph of G is called a *connected component* of G. A set $X \subseteq V$ is a *clique* if $uv \in E$ for every pair of vertices $u, v \in X$; and X is an *independent set* if $uv \notin E$ for every pair of vertices $u, v \in X$.

A *chord* of a cycle is an edge between two non consecutive vertices of the cycle. A graph is *chordal* if every cycle of length at least 4 contains a chord. It is easy to see that induced subgraphs of chordal graphs are also chordal [15]. Chordal graphs form a very well studied graph class, and they have many interesting characterizations. For our purposes a characterization via simplicial vertices will be sufficient. A vertex v is called *simplicial* if $N(v)$ is a clique. Every chordal graph has at least one simplicial vertex [4]. From this it follows that, in a chordal graph, one can repeatedly find a simplicial vertex in the remaining graph and remove it, until the graph becomes empty. The ordering in which the vertices of the starting graph have been removed in this way, is called a *perfect elimination ordering.* Interestingly, chordal graphs are exactly the graphs on which one can perform such an elimination procedure of simplicial vertices.

Theorem 1 ([11]) *A graph is chordal if and only if it has a perfect elimination ordering.*

A graph is a *split graph* if its vertex set can be partitioned into a clique and an independent set. Split graphs form a subclass of chordal graphs, as they contain no chordless cycles of length 4 or more.

Given a set $S \subseteq V$, a set $U \subseteq V$ is a *subset feedback vertex set* of (G, S) if no cycle in $G - U$ contains a vertex of S. Observe that U is a subset feedback vertex set of (G, S) if and only if $V \setminus U$ is an S-forest. If $S = V$ then U is a *feedback vertex set* of G, and $V \setminus U$ is a forest. A subset feedback vertex set U is *minimal* if no proper subset of U is a subset feedback vertex set of (G, S), and an S-forest is *maximal* if it cannot be extended to a larger S-forest by including more vertices of G. Clearly, U is a minimal subset feedback vertex set of (G, S) if and only if $V \setminus U$ is a maximal S-forest of G. Consequently, the number of minimal subset feedback vertex sets of (G, S) is equal to the number of maximal S-forests of G.

Let $\mu(G, S)$ denote the number of minimal subset feedback vertex sets of (G, S), equivalently the number of maximal S-forests of G. Observe that $\mu(G, S) = \prod_{i=1}^{t} \mu(G_i, S)$, where

$G_1, G_2, \ldots, G_t$ are the connected components of G. This is because every maximal S-forest of G is the union of maximal S-forests of the connected components of G.

Let $\mu(G) = \max\{\mu(G,S) \mid S \subseteq V\}$ denote the maximum number of minimal subset feedback vertex sets graph G can have, regardless of S. Note that $\mu(G)$ is lower bounded by the number of minimal feedback vertex set of G. Let H be the complete graph on 5 vertices. This graph has 10 minimal feedback vertex sets [2]. Let H_ℓ be the graph obtained by taking ℓ disjoint copies of H, for $\ell \geq 1$. The number of minimal feedback vertex sets of H_ℓ is thus $10^\ell = 10^{n/5} \approx 1.5848^n$. Any graph H_ℓ is chordal and hence $10^{n/5}$ is a lower bound on the number of minimal subset feedback vertex sets of chordal graphs, i.e., there is a chordal graph $G = (V, E)$ and a set $S \subseteq V$ such that (G, S) has $10^{n/5}$ minimal subset feedback vertex set. When it comes to the maximum number of minimal feedback vertex sets in chordal graphs, Couturier et al. showed that the above lower bound is also the upper bound [2]. An upper bound on the number of minimal subset feedback vertex sets of chordal graphs better than the one for general graphs has not been known until the result we present below.

3 Enumerating minimal subset feedback vertex sets in chordal graphs

In this section, we describe an algorithm that takes as input a chordal graph $G = (V, E)$ and a vertex subset $S \subseteq V$, and lists all maximal S-forests of G. Our algorithm is a recursive branching algorithm; every maximal S-forest of G will be present at some leaf of the corresponding branching tree, whereas some of the leaves might not correspond to maximal S-forests. Every recursive call has input (G', F, U, R), where F is the set of vertices of G placed so far in an S-forest of G, U is the set of vertices so far deleted from G and hence placed in the corresponding subset feedback vertex set, $R \subseteq F$ is the set of vertices that are placed in F and that are no longer relevant for making further decisions, and $G' = G - (U \cup R)$. We call the vertices in R *hidden*. The vertices in $V \setminus (U \cup F)$ are called *undecided* vertices. As G and S do not change throughout the algorithm, they are not parts of the input to the recursive calls. To summarize, (G', F, U, R) is simply an instance of the problem of listing all maximal S-forests of G containing F and not containing U, and our described algorithm solves exactly this problem. Given G and S, the main program runs this recursive algorithm on $(G, \emptyset, \emptyset, \emptyset)$.

If at some call (G', F, U, R), the graph G' has no undecided vertices, then we are at a leaf of the branching tree, and the algorithm stops after checking whether F is a maximal S-forest of G. If F is a maximal S-forest, it is added to the list of S-forests that will be output. If G' has undecided vertices, the algorithm continues, but first it checks whether F is an S-forest. If not, then the algorithm stops, discards F since it can never lead to a maximal S-forest, and no new subproblems are generated from this instance. If the algorithm continues, then since G' is chordal, we know that G' has a simplicial vertex. The algorithm chooses an arbitrary simplicial vertex v of G' and makes choices depending on v. Vertex v might already be placed in F or not; these two cases will be handled separately by the algorithm as Case 1 and Case 2 below. The following operations will be used in our algorithm:

- *Deleting* a vertex x: deletes x from G' and adds it to U. Vertex x will be permanently deleted from G' and it will be a part of the suggested subset feedback vertex set U in all subsequent subproblems.
- *Adding* a vertex x to F: adds x to F. Vertex x will be a part of F in all subsequent subproblems, and will never be considered for deletion.
- *Hiding* a vertex x of F: this operation is only applicable on some simplicial vertices of G' that are already placed in F. We apply it when x is no longer relevant for making

further decisions on the remaining vertices of $G' - F$. When x is hidden, it is added to R and removed from G' but it remains a part of F in all subsequent subproblems, and in particular it remains in $G - U$.

Throughout the algorithm we will keep the following invariant.

Invariant 1 *Let (G', F, U, R) be an instance. For any S-cycle C in $G - U$ that contains a vertex of R, there is an S-cycle C' in G' such that $V(C') = V(C) \setminus R$.*

Invariant 1 is clearly true when R is empty. Whenever we hide a vertex v, we will argue that the invariant is still true after v is hidden. The next lemma shows that we can safely ignore the vertices in R when we make further decisions on $G - U$, and hence it is safe to work on $G' = G - (U \cup R)$ instead of $G - U$.

Lemma 1. *Let (G', F, U, R) be an instance. Under Invariant 1, F' is a maximal S-forest of $G - U$ such that $F \subseteq F'$ if and only if $F' \setminus R$ is a maximal S-forest of G'.*

Proof. Let F' be a maximal S-forest of $G - U$ such that $F \subseteq F'$. Then clearly $F' \setminus R$ is an S-forest in G'. Let us argue for maximality. Since F' is maximal, for any vertex x of $G - (U \cup F')$, x is involved in an S-cycle C in $G - U$ such that $V(C) \subseteq F' \cup \{x\}$. Observe that since $R \subseteq F \subseteq F'$, any such vertex x is also a vertex in G'. By Invariant 1, x is involved in an S-cycle C' in G' such that $V(C') = V(C) \setminus R$. Since $G' = G - (U \cup R)$, it follows that $V(C') \subseteq (F' \setminus R) \cup \{x\}$. Hence x cannot be added to $F' \setminus R$, which is thus a maximal S-forest of G'.

For the other direction, assume that $F' \setminus R$ is a maximal S-forest of G'. Hence every vertex x in G' outside of $F' \setminus R$ is involved in an S-cycle C in G' such that $V(C) \subseteq (F' \setminus R) \cup \{x\}$. Since $G - U$ is a supergraph of G', C is also an S-cycle in $G - U$. Hence no more vertices can be added to F', which is thus maximal. Let us argue that F' is an S-forest. Assume for contradiction that it is not. Then a vertex y of R is involved in an S-cycle C in $G - U$ such that $V(C) \subseteq F'$. Then by Invariant 1, there is an S-cycle C' in G' such that $V(C') \subseteq F' \setminus R$, which contradicts the assumption that $F' \setminus R$ is an S-forest of G'. □

The *measure* of an instance (G', F, U, R) is the number of undecided vertices, i.e., the vertices in $G' - F$. In the beginning of the algorithm all vertices are undecided and hence the measure of $(G, \emptyset, \emptyset, \emptyset)$ is n. The measure drops by the number of vertices deleted from G' plus the number of vertices added to F. Hiding a vertex does not affect the measure of an instance. In the call with input (G', F, U, R), the algorithm will further branch into subproblems in which some vertices will be deleted from G' and some vertices will be placed in F, and the measure will drop accordingly. If at a step, we branch into t new subproblems, where the measure decreases by $c_1, c_2, \ldots, c_t$ in each subproblem, respectively, we get the *branching vector* $(c_1, c_2, \ldots, c_t)$. At each branching point, we will give the corresponding branching vector to be of help in the running time analysis.

We now describe the reduction and the branching rules of the algorithm when G' has undecided vertices and F is an S-forest. Let (G', F, U, R) be a call of the algorithm satisfying this. In the below, we let $N(v) = N_{G'}(v)$, $N[v] = N_{G'}[v]$, and $d(v) = d_{G'}(v)$. First, we state three reduction rules. These rules are applied recursively on the considered instance as long as it is possible to apply at least one of them.

It is easy to see that the first reduction rule, Rule A, is safe since $\{u, v, w\}$ forms an S-cycle, and u, w are already placed in F:

Rule A. *If in G' an undecided vertex v is adjacent to vertices $u, w \in F$ such that $uw \in E$ and $\{u, v, w\} \cap S \neq \emptyset$, then delete v, i.e., reduce to the subproblem $(G' - v, F, U \cup \{v\}, R)$.*

The following observation immediately results in the next reduction rule: Rule B.

Observation 1 *Let v be a vertex of G' such that no S-cycle of G' contains v. Then v must be added to F if it is not in F, and it is then safe to hide v.*

Proof. If vertex v is not involved in an S-cycle, then it cannot get involved in an S-cycle at later steps when more and more vertices are deleted from G' and added to U. Hence it is safe to add it to F, and due to maximality it must be added to F if it is not already in F. Assume now that $v \in F$. Recall that for any S-cycle C in $G - U$ that contains a vertex of R, there is an S-cycle C' in G' such $V(C') = V(C) \setminus R$. Because no S-cycle in G' contains v, C' is an S-cycle in $G' - v$. Hence, it is safe to hide v and add it to R. □

Rule B. *If G' has a vertex v with $d(v) \leq 1$, then add v to F if v is undecided, and when $v \in F$ then hide v, i.e., reduce to the subproblem $(G' - v, F \cup \{v\}, U, R \cup \{v\})$.*

Since G' is not empty and it is chordal, it has a simplicial vertex. With the following observation we obtain the next reduction rule: Rule C.

Observation 2 *Let v be a simplicial vertex of G'. If $N[v] \cap S = \emptyset$, then v must be added to F if it is not already in F, and it is then safe to hide v.*

Proof. Recall that by Rule B, we can assume that $d(v) \geq 2$. First we prove that if $v \notin F$, then it must be added to F. Let F' be any maximal S-forest of G' such that $F \subseteq F'$. If F' contains at most one vertex of $N(v)$, then F' must contain v by Observation 1. If F' contains two or more vertices from $N(v)$, then since none of these belongs to S-cycles in $G'[F']$, they are all pairwise adjacent, and $v \notin S$, v cannot belong to an S-cycle in $G'[F' \cup \{v\}]$. Since v has no other neighbors in G' and F' is maximal, F' must thus contain v.

We now prove that it is safe to hide v when $v \in F$. Let $C = v, x_1, \ldots, x_k, v$ be an S-cycle in G'. Since $x_1, x_k \in N(v)$, they do not belong to S. Since v does not belong to S either, a vertex from $\{x_2, \ldots, x_{k-1}\}$ belongs to S. Since v is simplicial, x_1 and x_k are adjacent. Consequently, $x_1, \ldots, x_k, x_1$ is an S-cycle in $G' - v$. It follows immediately that if $v \in F$, then we can hide it and add it to R. □

Rule C. *If there is a simplicial vertex v such that $N[v] \cap S = \emptyset$, then add v to F if v is undecided, and when $v \in F$ then hide v, i.e., reduce to the subproblem $(G' - v, F \cup \{v\}, U, R \cup \{v\})$.*

If we cannot apply any of the Rules A, B, or C, then we apply one of the branching rules below. In particular, we pick a simplicial vertex v, hence $N(v)$ is a clique. Vertex v is either undecided or it belongs to F. If v is undecided then we proceed as described in Case 1 below. If $v \in F$ then we proceed as described in Case 2 below. Notice that by Rule B, $d(v) \geq 2$ in both cases.

Case 1. The chosen simplicial vertex v is undecided

In this case, we know that $v \notin F$. However, v might be in S or not, and v might have a neighbor in F or not. We have four cases corresponding to these possibilities, and no other case is possible.

Case 1.1 $v \notin F$, $v \in S$, and $N(v) \cap F = \emptyset$.

If $d(v) = 2$ then let u_1 and u_2 be the two neighbors of v. Since $v \in S$, at most two vertices from $\{v, u_1, u_2\}$ can be added to F. Note however that, if exactly one of u_1, u_2 is added to F and the other one is deleted, then v must also be added to F by Observation 1. This implies that if v is deleted then both u_1 and u_2 must be added to F. Consequently, we branch into the following subproblems, which cover all possibilities, and we obtain $(3, 3, 3, 3)$ as the branching vector:

- Vertex v is deleted from G' and added to U; vertices u_1 and u_2 are added to F: the decrease in the measure is 3.
- Vertex u_1 is deleted from G' and added to U; vertices v and u_2 are added to F: the decrease is 3.
- Vertex u_2 is deleted from G' and added to U; vertices v and u_1 are added to F: the decrease is 3.
- Vertices u_1 and u_2 are deleted from G' and added to U; vertex v is added to F: the decrease is 3.

If $d(v) = 3$ then let u_1, u_2, u_3 be the three neighbors of v. Because $v \in S$, if v is added to F then at most one of the vertices u_1, u_2, u_3 can be included in F. As above, we will branch on the possibilities of adding v and at most one of its neighbors into F and deleting the other neighbors, or deleting v. For the choice of deleting v, we observe the following: either u_1 is added to F or u_1 is also deleted. If both v and u_1 are deleted, then both u_2 and u_3 must be added to F, by Observation 1. Consequently, we branch into the following subproblems, which cover all possibilities, and we obtain $(4, 4, 4, 4, 2, 4)$ as the branching vector:

- Vertices u_2 and u_3 are deleted from G' and added to U; vertices v and u_1 are added to F: the decrease is 4.
- Vertices u_1 and u_3 are deleted from G' and added to U; vertices v and u_2 are added to F: the decrease is 4.
- Vertices u_1 and u_2 are deleted from G' and added to U; vertices v and u_3 are added to F: the decrease is 4.
- Vertices u_1, u_2, and u_3 are deleted from G' and added to U; vertex v is added to F: the decrease is 4.
- Vertex v is deleted from G' and added to U; vertex u_1 is added to F: the decrease is 2.
- Vertices v and u_1 are deleted from G' and added to U; vertices u_2 and u_3 are added to F: the decrease is 4.

In the rest we assume that $t = d(v) \geq 4$. By the same arguments as above, either v is deleted or it is added to F with at most one of its neighbors. Consequently, we branch into the following subproblems, where $u_1, u_2, \ldots, u_t$ are the neighbors of v in G':

- Vertex v is deleted from G' and added to U; nothing else changes: the decrease in the measure is 1.
- Vertex v is added to F; all of its neighbors are deleted from G' and added to U: the decrease in the measure is $t + 1$.
- Vertices v and u_1 are added to F; all other neighbors of v are deleted from G' and added to U: the decrease is $t + 1$.
- The last step above is repeated with each of the other neighbors of v instead of u_1: the decrease is $t + 1$ in each of these $t - 1$ additional cases.

The branching vector is $(1, t + 1, t + 1, \ldots, t + 1)$, where the term $t + 1$ appears $t + 1$ times, and $t \geq 4$.

Case 1.2 $v \notin F$, $v \in S$, and $N(v) \cap F \neq \emptyset$.

As we cannot apply Rule A for the considered instance, $|N(v) \cap F| = 1$. Since $t = d(v) \geq 2$, we know that v has exactly one neighbor in F, say $u_1 \in F$, whereas the rest of its neighbors $u_2, \ldots, u_t$ are undecided. We branch into the two possibilities of adding v to F or deleting v. If we add v to F, since one neighbor is already in F then none of the $t-1$ undecided neighbors can be added, and therefore we delete them from G' and add them to U. We get the following two subproblems: $(G' - v, U \cup \{v\}, F, R)$ and $(G' - \{u_2, \ldots, u_t\}, U \cup \{u_2, \ldots, u_t\}, F \cup \{v\}, R)$. In the first subproblem the measure decreases by 1, and in the second it decreases by t. We get the branching vector $(1, t)$ with $t \geq 2$.

Case 1.3 $v \notin F$, $v \notin S$, and $N(v) \cap F = \emptyset$.

Since we cannot apply Rule C, v has at least one neighbor belonging to S.

If $d(v) = 2$, let u_1 and u_2 be the neighbors of v. Since u_1 or u_2 belongs to S, we know that at most two vertices from $\{v, u_1, u_2\}$ can be added to F. Consequently, this case is identical to the subcase of Case 1.1 handling $d(v) = 2$. We branch into the same subproblems and we obtain $(3, 3, 3, 3)$ as the branching vector.

If $d(v) = 3$, let u_1, u_2, u_3 be the neighbors of v. Assume without loss of generality that $u_1 \in S$. This case is very similar to the subcase of Case 1.1 handling $d(v) = 3$, but now we branch on u_1 instead of v. If u_1 is added to F then at most one of v, u_2, u_3 can be added to F. If u_1 is added to F and u_2, u_3 are added to U then v should be added to F, by Observation 1. If u_1 is deleted then either v is added to F or v is also deleted. If v is also deleted then both u_2 and u_3 must be added to F, by Observation 1. Consequently, we branch into the following subproblems, which cover all possibilities, and we obtain $(4, 4, 4, 2, 4)$ as the branching vector:

- Vertices u_2 and u_3 are deleted from G' and added to U; vertices u_1 and v are added to F: the decrease is 4.
- Vertices v and u_3 are deleted from G' and added to U; vertices u_1 and u_2 are added to F: the decrease is 4.
- Vertices v and u_2 are deleted from G' and added to U; vertices u_1 and u_3 are added to F: the decrease is 4.
- Vertex u_1 is deleted from G' and added to U; vertex v is added to F: the decrease is 2.
- Vertices u_1 and v are deleted from G' and added to U; vertices u_2 and u_3 are added to F: the decrease is 4.

If $t = d(v) \geq 4$, then let $u_1, u_2, \ldots, u_t$ be the neighbors of v in G', and assume without loss of generality that $u_1 \in S$. We will branch on the two possibilities of adding u_1 to F and deleting u_1. If we add u_1 to F then we can add at most one other vertex of $N[v]$ to F and all others must be deleted. If u_1 is added to F and $u_2, \ldots, u_t$ are added to U then v should be added to F, by Observation 1. Consequently, we branch into the following subproblems:

- Vertex u_1 is deleted from G' and added to U; nothing else changes: the decrease in the measure is 1.
- Vertices u_1 and v are added to F; all other neighbors of v are deleted from G' and added to U: the decrease is $t + 1$.
- Vertices u_1 and u_2 are added to F; v and all other neighbors of v are deleted from G' and added to U: the decrease is $t + 1$.
- The last step above is repeated with each of the neighbors $u_3, \ldots, u_t$ of v instead of u_2: the decrease is $t + 1$ in each of these $t - 2$ additional cases.

The branching vector is $(1, t + 1, t + 1, \ldots, t + 1)$, where the term $t + 1$ appears t times, with $t \geq 4$.

Case 1.4 $v \notin F$, $v \notin S$, and $N(v) \cap F \neq \emptyset$.

As we cannot apply Rule C, $N(v) \cap S \neq \emptyset$. Suppose that $|N(v) \cap F| \geq 2$. If there is a vertex $u \in (N(v) \setminus F) \cap S$, then Rule A can be applied for u. Consequently, there is a vertex $u \in N(v) \cap F \cap S$, but then Rule A can be applied for v. It means that v has exactly one neighbor u in F. We take action depending on whether or not u belongs to S:

If $u \in S$, then at most one more vertex from $N[v]$ can be added to F, and all others must be deleted from G' and added to U. We get $t = d(v)$ subproblems in each of which a vertex of $N[v] \setminus \{u\}$ is added to F and all others are deleted from G' and added to U. Observe that we do not get a subproblem where all vertices of $N[v] \setminus \{u\}$ are deleted from G', due to Observation 1. Thus we get $(t, \ldots, t)$ as the branching vector, where the term t is repeated t times, and $t \geq 2$.

If $u \notin S$, then we know that v has another neighbor $w \in S$. We branch into two subproblems resulting from adding w to F or deleting w from G'. If we add w to F, then since u is also in F, no other vertex from $N[v]$ can be added to F and hence they must all be deleted from G' and added to U. We get a subproblem in which the measure decreases by $t = d(v)$. In the other subproblem we simply delete w from G' and add it to U; the decrease is 1. Hence we get $(1, t)$ as the branching vector for this case, where $t \geq 2$.

Case 2. The chosen simplicial vertex v belongs to F

In addition to belonging to F, v might either belong to S or not. Our algorithm takes action depending on this.

Case 2.1 $v \in F$ and $v \in S$.

Because $G[F]$ has no S-cycles, $|N(v) \cap F| \leq 1$. If $N(v) \cap F \neq \emptyset$, then Rule A can be applied for the vertices $N(v) \setminus F$. It follows that $N(v) \cap F = \emptyset$. Since $v \in S$ and $v \in F$, at most one vertex of $N(v)$ can be added to F, regardless of how many of these are in S.

If $d(v) = 2$ then let u and w be the two neighbors of v. We branch on the two possibilities of either adding u to the S-forest F or adding u to the subset feedback vertex set U. In the latter subproblem we delete u from G' and add it to U; the decrease is 1. In the first subproblem, we add u to F, and consequently we must delete w from G' and add it to U; the decrease is 2. We get $(1, 2)$ as the branching vector.

If $t = d(v) \geq 3$ then we branch into the possibilities of adding exactly one vertex of $N(v)$ to F and deleting all others from G', or deleting all vertices of $N(v)$ from G'. We get t subproblems in which one vertex is added to F and all other vertices of $N(v)$ are deleted from G' and added to U, and one subproblem in which all vertices of $N(v)$ are deleted from G' and added to U. In each of these $t + 1$ subproblems the decrease is t. Hence we get $(t, t, t, \ldots, t)$ as the branching vector, where the term t is repeated $t + 1$ times, and $t \geq 3$.

Case 2.2 $v \in F$ and $v \notin S$.

Suppose that $N(v) \cap F \neq \emptyset$. If a neighbor u of v is both in F and in S, then all other neighbors of v are undecided, since $G[F]$ has no S-cycles. Then we can apply Rule A for these neighbors of v. If there is $u \in (N(v) \cap F) \setminus S$, then Rule A can be applied for all $w \in N(v) \cap S$. It means that $N(v) \cap S = \emptyset$, but in this case we can apply Rule C. Therefore, $N(v) \cap F = \emptyset$. Because we cannot apply Rule C, v has at least one neighbor that is undecided and belongs to S.

Recall that $t = d(v) \geq 2$, and let $u_1, u_2, \ldots, u_t$ be the neighbors of v, and assume without loss of generality that $u_1 \in S$. We branch into the two possibilities of either deleting u_1 from G' and adding it to U, or adding u_1 to F. In the latter case, no other neighbor of $N(v)$ can be added to F, since they all form S-cycles with v and u_1, and hence they must all be deleted from G' and added to U. We get one subproblem where the decrease is 1, and one subproblem where the decrease is t. This gives us the branching vector $(1, t)$ with $t \geq 2$.

The description of the algorithm is now complete, and we are ready to conclude our main result.

Theorem 2 *All minimal subset feedback vertex sets of a chordal graph on n vertices can be listed in $O(1.6708^n)$ time.*

Proof. We claim that the algorithm described before the statement of the theorem is exactly an algorithm achieving the statement of the theorem. The correctness of the described algorithm follows from Invariant 1, Lemma 1, Observations 1, 2, and the arguments given for each case, observing that we have taken care of all possible cases. It remains to analyse the running time.

Let G be a chordal graph on n vertices and let S be any set of vertices in G, that form the input to the main program that calls our recursive algorithm. In each of the branching rules, the measure decreases as described, and in each of the reduction rules, either the measure decreases or at least one vertex of F is deleted from G'. When all vertices of G' are either in U or in F, then the recurrence stops. At this point we need to check whether F is a maximal S-forest of G. This can easily be done in polynomial time; F is an S-forest if and only if every vertex of $S \cap F$ is incident in G to edges that are bridges. Maximality is also easy to check since if a subset X of $V \setminus F$ can be added to F to obtain a larger S-forest, then also a single vertex of X can be added, so we can repeatedly check possible extensions by single vertices. Consequently, the running time will be upper bounded by the number of leaves in the search tree.

For the analysis of the running time $T(n)$, we use terminology that is standard for recursive branching algorithms [9]. In particular, a branching vector $(c_1, c_2, \ldots, c_t)$ results in the recurrence $T(n) \leq T(n-c_1) + T(n-c_2) + \ldots + T(n-c_t)$. In this case $T(n) = O^*(\alpha^n)$, where α is the unique positive real root of $x^n - x^{n-c_1} - \ldots - x^{n-c_t} = 0$ [9], and the O^*-notation suppresses polynomial factors. The number α is called the *branching number* of this branching vector. It is common to round α to the fourth digit after the decimal point. By rounding the last digit up, we can use O-notation instead of O^*-notation [9]. As different branching vectors are involved at different steps of our algorithm, the branching vector with the highest branching number gives an upper bound on $T(n)$.

We now list the branching vectors that have appeared during the description of the algorithm, in the order of first appearance. We give the branching number for each of them; however we do not include here the explicit calculations.

- $(3, 3, 3, 3)$: the branching number is ≈ 1.5875.
- $(4, 4, 4, 4, 2, 4)$: the branching number is ≈ 1.6707.
- $(1, t, t, t, t, \ldots, t)$, where the term t appears t times, and $t \geq 5$: $(1, 5, 5, 5, 5, 5)$ gives the maximum branching number for this vector, which is ≈ 1.6594.
- $(1, t)$, $t \geq 2$: $(1, 2)$ gives the maximum branching number for this branching vector, which is ≈ 1.6181.
- $(4, 4, 4, 2, 4)$: the branching number is ≈ 1.6005.
- $(1, t, t, t, t, \ldots, t)$, where the term t appears $t-1$ times, and $t \geq 5$: $(1, 5, 5, 5, 5)$ gives the maximum branching number for this vector, which is ≈ 1.60041.
- $(t, \ldots, t)$, where the term t is repeated t times, and $t \geq 2$: $(3, 3, 3)$ gives the maximum branching number for this vector, which is ≈ 1.4423.
- $(1, 2)$: the branching number is ≈ 1.6181.

– $(t, t, \ldots, t)$, where the term t is repeated $t+1$ times, and $t \geq 3$: $(3,3,3,3)$ gives the maximum branching number for this vector, which is ≈ 1.5875.

The largest branching number is ≈ 1.6707, and it is obtained for $(4,4,4,4,2,4)$. Thus the running time of our algorithm is $O(1.6708^n)$. □

Two corollaries follow from the above result.

Corollary 1. *Both weighted and unweighted versions of* Subset Feedback Vertex Set *can be solved in* $O(1.6708^n)$ *time on chordal graphs.*

Proof. Observe that any subset feedback vertex set of minimum cardinality or minimum weight is a minimal subset feedback vertex set. Hence we can check the cardinality or weight of each generated minimal subset feedback vertex set, and compare the smallest one with the given bound k of the input. □

Theorem 2 immediately implies that a chordal graph on n vertices has $O(1.6708^n)$ minimal subset feedback vertex sets. To avoid the constant hidden in the big-O notation, we prove the following corollary.

Corollary 2. *A chordal graph on n vertices has at most 1.6708^n minimal subset feedback vertex sets.*

Proof. Let G be a chordal graph on n vertices and let S be any set of vertices in G. From the correctness of the described algorithm it follows that $\mu(G,S)$ is upper bounded by the number of leaves in the search tree corresponding to the algorithm. Let $L(n)$ denote the maximum number of leaves in a search tree resulting from running our algorithm on an n-vertex chordal graph $G=(V,E)$ and a subset $S \subseteq V$. From the description of the algorithm we see that $L(n)$ satisfies the following inequalities; we list them in the same order as the corresponding branching vectors in the proof of Theorem 2.

1. $L(n) \leq 4\, L(n-3)$.
2. $L(n) \leq 5\, L(n-4) + L(n-2)$.
3. $L(n) \leq t\, L(n-t) + L(n-1)$, where $t \geq 5$.
4. $L(n) \leq L(n-t) + L(n-1)$, where $t \geq 2$.
5. $L(n) \leq 4\, L(n-4) + L(n-2)$.
6. $L(n) \leq (t-1)\, L(n-t) + L(n-1)$, where $t \geq 5$.
7. $L(n) \leq t\, L(n-t)$, where $t \geq 2$.
8. $L(n) \leq L(n-2) + L(n-1)$.
9. $L(n) \leq (t+1)\, L(n-t)$, where $t \geq 3$.

Clearly, $L(1)=0$, and we assume that $L(n)=0<1.6708^n$ for $n \leq 0$. We will prove by induction that $L(n) \leq 1.6708^n$. Assume that this induction hypothesis is true on all chordal graphs with at most $n-1$ vertices. We will show that this leads to $L(n) \leq 1.6708^n$ for each of the inequalities above.

For inequality 1, we need to verify that $4 \cdot 1.6708^{n-3} \leq 1.6708^n$, which amounts to observing that $4 \leq 1.6708^3 \approx 4.6642$. For inequality 2, we need to verify that $5 \cdot 1.6708^{n-4} + 1.6708^{n-2} \leq 1.6708^n$, which amounts to observing that $5 \leq 1.6708^4 - 1.6708^2 \approx 5.0013$. As $4\, L(n-4) + L(n-2) \leq 5\, L(n-4) + L(n-2)$, we immediately obtain the bound for inequality 5. For inequality 3, we need to verify that $t \cdot 1.6708^{n-t} + 1.6708^{n-1} \leq 1.6708^n$, which amounts to observing that $t \leq 1.6708^t - 1.6708^{t-1}$, for all $t \geq 5$. Because $(t-1)\, L(n-t) + L(n-1) \leq t\, L(n-t) + L(n-1)$, we have the bound for inequality 6 as well. For inequalities 4 and

8, we observe that if $t \geq 2$ then $L(n) \leq L(n-t) + L(n-1) \leq L(n-2) + L(n-1)$, and we need to verify that $1.6708^{n-2} + 1.6708^{n-1} \leq 1.6708^n$, which amounts to observing that $1 + 1.6708 = 2.6708 \leq 1.6708^2 \approx 2.7916$. Finally, for inequalities 6 and 9, we have $L(n) \leq t\ L(n-t) \leq (t+1)\ L(n-t)$, and we verify that $(t+1) \cdot 1.6708^{n-t} \leq 1.6708^n$, by observing that $t+1 \leq 1.6708^t$, for all $t \geq 3$. □

4 Concluding remarks

As mentioned earlier, there are chordal graphs with $10^{n/5} \approx 1.5848^n$ minimal subset feedback vertex sets. We have shown that the maximum number of minimal subset feedback vertex sets in chordal graphs is upper bounded by 1.6708^n.

- Could it be that the lower bound is also an upper bound or are there chordal graphs with more than $10^{n/5}$ minimal subset feedback vertex sets?
- Is there an algorithm for Subset Feedback Vertex Set on chordal graphs with running time $O(c^n)$ such that $c < 1.6708^n$?

The lower bound on the maximum number of minimal subset feedback vertex sets of a split graph is $3^{n/3}$ [8], and it is obtained when the set S is equal to the independent set in the partition of the vertices into an independent set and a clique. Surprisingly, improving the upper bound that we gave in this paper even on split graphs seems to be a non-trivial task.

- Is there a better lower bound on split graphs?
- Is there a better upper bound on split graphs than on chordal graphs?
- Does Subset Feedback Vertex Set admit a faster algorithm on split graphs than on chordal graphs?

Finally, we conclude with the following question.

- Can all minimal subset feedback vertex sets in a graph be enumerated in output polynomial time, i.e., in time that is polynomial in the number of minimal subset feedback vertex sets?

Such an algorithm is known for enumerating minimal feedback vertex sets in general graphs [20]. It would be very interesting to have such an algorithm for subset feedback vertex sets, even on chordal graphs or split graphs.

References

1. D. G. Corneil and J. Fonlupt. The complexity of generalized clique covering. Disc. Appl. Math., 22:109–118 (1988/1989).
2. J.-F. Couturier, P. Heggernes, P. van 't Hof, Y. Villanger. Maximum number of minimal feedback vertex sets in chordal graphs and cographs. In Proceedings of COCOON 2012, LNCS 7434: 133-144 (2012).
3. M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. Subset feedback vertex set is fixed parameter tractable. In Proceedings of ICALP 2011, LNCS 6755:449–461 (2011).
4. G. A. Dirac. On rigid circuit graphs. Abh. Math. Sem. Univ. Hamburg, 25:71–76 (1961).
5. G. Even, J. Naor, and L. Zosin. An 8-approximation algorithm for the subset feedback vertex set problem. SIAM J. Comput., 30(4):1231–1252, 2000.
6. F. V. Fomin, S. Gaspers, A. V. Pyatkin, and I. Razgon. On the minimum feedback vertex set problem: Exact and enumeration algorithms. Algorithmica 52(2): 293–307 (2008).

7. F. V. Fomin, F. Grandoni, A. V. Pyatkin, and A. A. Stepanov. Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications. ACM Trans. Algorithms 5(1): (2008).
8. F. V. Fomin, P. Heggernes, D. Kratsch, C. Papadopoulos, and Y. Villanger. Enumerating minimal subset feedback vertex sets. In Proceedings of WADS, LNCS 6844: 399–410 (2011).
9. F. V. Fomin and D. Kratsch. Exact Exponential Algorithms. Springer, Texts in Theoretical Computer Science (2010).
10. F. V. Fomin and Y. Villanger. Finding induced subgraphs via minimal triangulations. In Proceedings of STACS 2010: 383–394 (2010).
11. D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. Pacific Journal of Math., 15:835–855 (1965).
12. J. A. George and J. W. H. Liu. Computer Solution of Large Sparse Positive Definite Systems. Prentice-Hall Inc. (1981)
13. M. R. Garey and D. S. Johnson. Computers and Intractability, A guide to the theory of NP-completeness. Freeman and Co. (1979).
14. P. A. Golovach, P. Heggernes, D. Kratsch, and R. Saei. An exact algorithm for Subset Feedback Vertex Set on chordal graphs. In Proceedings of IPEC 2012, LNCS 7537: 85–96 (2012).
15. M. C. Golumbic. Algorithmic Graph Theory and Perfect Graphs. Annals of Disc. Math. 57, Elsevier (2004).
16. S. Gaspers and M. Mnich. On feedback vertex sets in tournaments. In Proceedings ESA 2010, LNCS 6346:267–277 (2010).
17. M. M. Kanté, V. Limouzy, A. Mary, and L. Nourine. Enumeration of Minimal Dominating Sets and Variants. In Proceedings of FCT 2011, LNCS 6914: 298–309 (2011).
18. D. Kratsch, H. Müller, and I. Todinca. Feedback vertex set on AT-free graphs. Disc. Appl. Math., 156: 1936–1947 (2008).
19. J. W. Moon and L. Moser. On cliques in graphs. Israel J. Math. 3: 23–28 (1965).
20. B. Schwikowski and E. Speckenmeyer. On enumerating all minimal solutions of feedback problems. Disc. Appl. Math., 117:253–265 (2002).
21. C. Semple and M. Steel. Phylogenetics. Oxford lecture series in mathematics and its applications (2003).
22. J. P. Spinrad. Efficient graph representations. AMS, Fields Institute Monograph Series 19 (2003).

Chapter 11

Maximal induced matchings in triangle-free graphs

Manu Basavaraju, Pinar Heggernes, Pim van 't Hof, Reza Saei, and Yngve Villanger

In *Graph Theoretic Concepts in Computer Science,*, D. Kratsch and I. Todinca, Eds., vol. 8747 of *Lecture Notes in Computer Science*. Springer International publishing, 2014, pp. 93–104.
(Submitted for journal publication)

Maximal induced matchings in triangle-free graphs*

Manu Basavaraju, Pinar Heggernes, Pim van 't Hof, Reza Saei, and Yngve Villanger

Department of Informatics, University of Bergen, Norway,
{manu.basavaraju,pinar.heggernes,pim.vanthof,reza.saeidinvar,
yngve.villanger}@ii.uib.no

Abstract. An induced matching in a graph is a set of edges whose endpoints induce a 1-regular subgraph. It is known that every n-vertex graph has at most $10^{n/5} \approx 1.5849^n$ maximal induced matchings, and this bound is best possible. We prove that every n-vertex triangle-free graph has at most $3^{n/3} \approx 1.4423^n$ maximal induced matchings; this bound is attained by every disjoint union of copies of the complete bipartite graph $K_{3,3}$. Our result implies that all maximal induced matchings in an n-vertex triangle-free graph can be listed in time $O(1.4423^n)$, yielding the fastest known algorithm for finding a maximum induced matching in a triangle-free graph.

1 Introduction

A celebrated result due to Moon and Moser [9] states that every graph on n vertices has at most $3^{n/3} \approx 1.4423^n$ maximal independent sets. Moon and Moser also proved that this bound is best possible by characterizing the extremal graphs as follows: a graph on n vertices has exactly $3^{n/3}$ maximal independent sets if and only if it is the disjoint union of $n/3$ triangles. Given the structure of these extremal graphs, it is natural to investigate how many maximal independent sets a triangle-free graph can have. Hujter and Tuza [7] showed that a triangle-free graph on n vertices has at most $2^{n/2} \approx 1.4143^n$ maximal independent sets; this bound is attained by every 1-regular graph. Later, Byskov [2] gave an algorithmic proof of the same result, along with more general results.

More recently, Gupta, Raman, and Saurabh [5] showed that for any fixed non-negative integer r, there exists a constant $c < 2$ such that every graph on n vertices has at most c^n maximal r-regular induced subgraphs. The aforementioned result by Moon and Moser implies that if $r = 0$, then $c = 3^{1/3}$ is the best possible upper bound. Gupta et al. [5] complement this by proving tight upper bounds for the case where $r \in \{1, 2\}$. In particular, their result for $r = 1$ shows that every n-vertex graph has at most $10^{n/5} \approx 1.5849^n$ maximal induced matchings, and this upper bound is attained by every disjoint union of complete graphs on five vertices. The structure of these extremal graphs again raises the question how much the upper bound can be improved for triangle-free graphs. We answer this question by proving the following result.

Theorem 1. *Every triangle-free graph on n vertices contains at most $3^{n/3}$ maximal induced matchings, and this bound is attained by every disjoint union of copies of $K_{3,3}$.*

We would like to mention some implications of the above theorem. There exist algorithms that list the maximal independent sets of any graph with polynomial delay [8,10], which means that the time spent between the output of two successive maximal independent sets

*This work is supported by the Research Council of Norway (project SCOPE, 197548/F20), and by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreement n. 267959.

is polynomial in the size of the graph. Together with the aforementioned upper bounds on the number of maximal independent sets, this implies that the maximal independent sets of an n-vertex graph G can be listed in time $O^*(3^{n/3})$, or in time $O^*(2^{n/2})$ in case G is triangle-free.[1]

Cameron [3] observed that the maximal induced matchings of a graph G are exactly the maximal independent sets in the square of the line graph of G. Consequently, the maximal induced matchings of any graph can be listed with polynomial delay. Combining this with the aforementioned upper bound by Gupta et al. [5] yields an algorithm for listing all maximal induced matchings of an n-vertex graph in time $O^*(10^{n/5}) = O(1.5849^n)$. Gupta et al. [5] also obtained an algorithm for finding a maximum induced matching in an n-vertex graph in time $O(1.4786^n)$, which is the current fastest algorithm for solving this problem. Theorem 1 implies that we can do better on triangle-free graphs, as the following two results show. We point out that the problem of finding a maximum induced matching remains NP-hard on subcubic planar bipartite graphs [6], a small subclass of triangle-free graphs.

Corollary 1. *For every triangle-free graph on n vertices, all its maximal induced matchings can be listed in time $O^*(3^{n/3}) = O(1.4423^n)$ with polynomial delay.*

Corollary 2. *For every triangle-free graph G on n vertices, a maximum induced matching in G can be found in time $O^*(3^{n/3}) = O(1.4423^n)$.*

2 Definitions and Notations

All graphs we consider are finite, simple and undirected. We refer the reader to the monograph by Diestel [4] for graph terminology and notation not defined below.

Let G be a graph. For a vertex $v \in V(G)$, we write $N_G(v)$ and $N_G[v]$ to denote open and closed neighborhoods of v, respectively. Let $A \subseteq V(G)$. The closed neighborhood of A is defined as $N_G[A] = \bigcup_{v\in A} N_G[v]$, and the open neighborhood of A is $N_G(A) = N_G[A] \setminus A$. We write $G[A]$ to denote the subgraph of G induced by A, and we write $G-A$ to denote the graph $G[V(G)\setminus A]$. If $A = \{v\}$, then we simply write $G-v$ instead of $G-\{v\}$. For any non-negative integer r, we say that G is *r-regular* if the degree of every vertex in G is r. A 3-regular graph is called *cubic*. A cycle C with vertices $v_1, v_2 \dots, v_k$ and edges $v_1v_2, \dots, v_{k-1}v_k, v_kv_1$ is denoted by $C = v_1v_2 \cdots v_k$.

A *matching* in G is a subset $M \subseteq E(G)$ such that no two edges in M share an endpoint. For a matching M in G and a vertex $v \in V(G)$, we say that M *covers* v if v is an endpoint of an edge in M. A matching M is called *induced* if the subgraph induced by endpoints of the edges in M is 1-regular. An induced matching M in G is *maximal* if there exists no induced matching M' in G such that $M \subsetneq M'$. We write M_G to denote the set of all maximal induced matchings in G. Let X and Y be two disjoint subsets of $V(G)$. We define $M_G(X,Y)$ to be the set of all maximal induced matchings of G that cover no vertex of X and every vertex of Y. Clearly, $M_G = M_G(\emptyset,\emptyset)$. When there is no ambiguity we omit subscripts from the notations.

3 Twins and Maximal Induced Matchings

Let G be a graph. Two vertices $u, v \in V(G)$ are *(false) twins* if $N_G(u) = N_G(v)$. In this paper, whenever we write twin, we mean false twin. For every vertex $u \in V(G)$, the *twin set* of u is defined as $T_G(u) = \{v \in V(G) \mid N_G(u) = N_G(v)\}$, i.e., $T_G(u)$ consists of the vertex u and all

[1] We use the O^*-notation to suppress polynomial factors, i.e., we write $O^*(f(n))$ instead of $O(f(n)\cdot n^{O(1)})$ for any function f.

its twins. All the twin sets together form a partition of the vertex set of G, and we write $\tau(G)$ to denote the number of sets in this partition, i.e., $\tau(G)$ denotes the number of twin sets in G.

Definition 1. *Let G be a graph. For any two non-adjacent vertices $u, v \in V(G)$, we define $G_{u\to v}$ to be the graph obtained from G by making u into a twin of v by deleting the edge ux for every $x \in N_G(u) \setminus N_G(v)$ and adding the edge uy for every $y \in N_G(v) \setminus N_G(u)$.*

The following lemma identifies certain pairs of vertices u and v for which the operation in Definition 1 does not decrease the number of maximal induced matchings in the graph. This lemma will play a crucial role in the proof of our main result. Note that this lemma holds for general graphs G, and not only for triangle-free graphs.

Lemma 1. *Let G be a graph and let $u, v \in V(G)$. If no maximal induced matching in G covers both u and v, then $|M_{G_{u\to v}}| \geq |M_G|$ or $|M_{G_{v\to u}}| \geq |M_G|$.*

Proof. Without loss of generality, we assume that the number of matchings in M_G that cover u is greater than or equal to the number of matchings in M_G that cover v, i.e., $|M_G(\emptyset, \{u\})| \geq |M_G(\emptyset, \{v\})|$. Since every matching in M_G that covers u does not cover v due to the assumption that $M_G(\emptyset, \{u, v\}) = \emptyset$, it holds that $M_G(\emptyset, \{u\}) = M_G(\{v\}, \{u\})$. By symmetry, we also have that $M_G(\emptyset, \{v\}) = M_G(\{u\}, \{v\})$. This implies that $|M_G(\{v\}, \{u\})| \geq |M_G(\{u\}, \{v\})|$. We now use this fact to prove that $|M_{G_{v\to u}}| \geq |M_G|$.

For convenience, we write $G' = G_{v\to u}$. The set M_G of all maximal induced matchings in G can be partitioned as follows:

$$M_G = M_G(\{v\}, \{u\}) \uplus M_G(\{u\}, \{v\}) \uplus M_G(\emptyset, \{u, v\}) \uplus M_G(\{u, v\}, \emptyset).$$

We can partition $M_{G'}$ in the same way:

$$M_{G'} = M_{G'}(\{v\}, \{u\}) \uplus M_{G'}(\{u\}, \{v\}) \uplus M_{G'}(\emptyset, \{u, v\}) \uplus M_{G'}(\{u, v\}, \emptyset).$$

We claim that $M_G(\{v\}, \{u\}) = M_{G'}(\{v\}, \{u\})$. Let $M \in M_G(\{v\}, \{u\})$. We claim that $M \in M_{G'}(\{v\}, \{u\})$. It is easy to verify that M is a induced matching in G', as we only change edges incident with v when transforming G into G', and M does not cover v. For contradiction, suppose M is not a maximal induced matching in G'. Then there is an edge $xy \in E(G')$ such that $M \cup \{xy\}$ is an induced matching in G'. Since u and v are twins in G' and M covers u, we find that $v \notin \{x, y\}$. This implies that $xy \in E(G)$, so $M \cup \{xy\}$ is a matching in G that does not cover v. In fact, $M \cup \{xy\}$ is an induced matching in G, since every edge in $E(G) \setminus E(G')$ is incident with v. This contradicts the maximality of M in G. Hence we have that $M_G(\{v\}, \{u\}) \subseteq M_{G'}(\{v\}, \{u\})$. To show why $M_{G'}(\{v\}, \{u\}) \subseteq M_G(\{v\}, \{u\})$, let $M' \in M_{G'}(\{v\}, \{u\})$. For similar reasons as before, M' is an induced matching in G. To show that M' is maximal in G, suppose for contradiction that there is an edge $xy \in E(G)$ such that $M' \cup \{xy\}$ is an induced matching in G. Then $v \notin \{x, y\}$, this time due to the assumption that no maximal induced matching in G covers both u and v. Now we can use similar arguments as before to conclude that $M' \cup \{x, y\}$ is an induced matching in G', yielding the desired contradiction.

By assumption, we have $M_G(\emptyset, \{u, v\}) = \emptyset$. Since u and v are twins in G' by construction, we also know that $M_{G'}(\emptyset, \{u, v\}) = \emptyset$ and $M_{G'}(\{v\}, \{u\}) = M_{G'}(\{u\}, \{v\})$. Recall that $|M_G(\{v\}, \{u\})| \geq |M_G(\{u\}, \{v\})|$, which implies that $|M_{G'}(\{u\}, \{v\})| \geq |M_G(\{u\}, \{v\})|$. Hence, in order to show that $|M_{G'}| \geq |M_G|$, it suffices to show that $|M_{G'}(\{u, v\}, \emptyset)| \geq |M_G(\{u, v\}, \emptyset)|$.

Let $M \in M_G(\{u,v\},\emptyset)$. We claim that $M \in M_{G'}(\{u,v\},\emptyset)$. It is easy to see that M is an induced matching in G', as the only edges that are modified are incident with v and M does not cover v. Suppose, for contradiction, that M is not a maximal induced matching in G'. Then there exists an edge $xy \in E(G')$ such that $M \cup \{xy\}$ is an induced matching in G'. If $v \notin \{x,y\}$, then $M \cup \{xy\}$ is also an induced matching in G, contradicting the maximality of M. Thus we have $v \in \{x,y\}$. Without loss of generality, suppose $x = v$. Let $M' = M \cup \{vy\}$. Now consider $M'' = M \cup \{uy\}$. Since M' is induced matching and u and v are twins in G', we infer that M'' is also an induced matching in G'. Note that the edge uy is also present in G, so M'' is an induced matching in G. This contradicts the maximality of M, implying that $M \in M_{G'}(\{u,v\},\emptyset)$ and consequently $M_G(\{u,v\},\emptyset) \subseteq M_{G'}(\{u,v\},\emptyset)$. This completes the proof of Lemma 1. □

For our purposes, we need to extend Definition 1 as follows.

Definition 2. *Let G be a graph. For any two non-adjacent vertices $u,v \in V(G)$, the graph $G_{T_G(u)\to v}$ is the graph obtained from G by making each vertex of $T_G(u)$ into a twin of v as follows: for every $u' \in T_G(u)$, delete the edge $u'x$ for every $x \in N_G(u) \setminus N_G(v)$ and add the edge $u'y$ for every $y \in N_G(v) \setminus N_G(u)$.*

The following lemma is an immediate corollary of Lemma 1, since we can repeatedly apply the operation in Definition 1 on all the vertices in $T_G(u)$.

Lemma 2. *Let G be a graph and let $u,v \in V(G)$. If no maximal induced matching in G covers both u and v, then $|M_{G_{T_G(u)\to v}}| \geq |M_G|$ or $|M_{G_{T_G(v)\to u}}| \geq |M_G|$.*

We also need the following two lemmas in the proof of our main result.

Lemma 3. *Let G be a triangle-free graph. For any two non-adjacent vertices $u,v \in V(G)$, the graph $G_{T_G(u)\to v}$ is triangle-free.*

Proof. Let $u,v \in V(G)$. For contradiction, suppose that $G_{T_G(u)\to v}$ contains a triangle C. Observe that every edge that was added to G in order to create $G_{T_G(u)\to v}$ is incident with a vertex in $T_G(u)$ and a vertex in $N_G(v) \setminus N_G(u)$. Hence, C contains an edge $u'x$ such that $u' \in T_G(u)$ and $x \in N_G(v) \setminus N_G(u)$. Let y be the third vertex of C. Since G is triangle-free, $N_G(v)$ forms an independent set in both G and $G_{T_G(u)\to v}$. This implies in particular that y is not adjacent to v in $G_{T_G(u)\to v}$, and since we did not delete any edge incident with v when creating $G_{T_G(u)\to v}$, it holds that y is not adjacent to v in G either. Moreover, since both u' and y do not belong to $N_G(v) \setminus N_G(u)$, the edge $u'y$ is present in G. But then, by Definition 2, the edge $u'y$ should have been deleted when G was transformed into $G_{T_G(u)\to v}$. This yields the desired contradiction. □

Lemma 4. *Let G be a triangle-free graph and let $u,v \in V(G)$ be two non-adjacent vertices. If u and v are not twins, then $\tau(G_{T_G(u)\to v}) < \tau(G)$.*

Proof. Suppose u and v are not twins. Then $T_G(u)$ and $T_G(v)$ are two different twin sets in G. By Definition 2, the vertices of $T_G(u) \cup T_G(v)$ all belong to the same twin set in $G_{T_G(u)\to v}$, namely the twin set $T_{G_{T_G(u)\to v}}(u) = T_{G_{T_G(u)\to v}}(v)$. Let $x \in V(G) \setminus (T_G(u) \cup T_G(v))$. We prove that all the vertices in $T_G(x)$ belong to the same twin set in $G_{T_G(u)\to v}$, which implies that $\tau(G_{T_G(u)\to v}) < \tau(G)$.

Suppose there is a vertex $y \in T_G(x)$ such that x and y are not twins in $G_{T_G(u)\to v}$. Without loss of generality, suppose there is a vertex $z \in N_{G_{T_G(u)\to v}}(y) \setminus N_{G_{T_G(u)\to v}}(x)$. Since x and y are twins in G, we either have $xz, yz \in E(G)$ or $xz, yz \notin E(G)$. In the first case, the edge

xz is deleted from G when $G_{T_G(u)\to v}$ is created, which implies that $x \in N_G(u) \setminus N_G(v)$ by Definition 2. However, since x and y are twins in G, it holds that $y \in N_G(u) \setminus N_G(v)$ as well, implying that the edge yz should not exist in $G_{T_G(u)\to v}$. This contradicts the definition of z. If $xz, yz \notin E(G)$, then we can use similar argument to conclude that xz should be an edge in $G_{T_G(u)\to v}$, again yielding a contradiction. □

4 Proof of Theorem 1

This section is devoted to proving Theorem 1. We first prove that every triangle-free graph on n vertices has at most $3^{n/3}$ maximal induced matchings. At the end of the section, we show why the bound in Theorem 1 is best possible.

A triangle-free graph on n vertices that has more than $3^{n/3}$ maximal induced matchings is called a *counterexample*. For contradiction, let us assume that there exists a counterexample. Then there exists a counterexample G such that for every counterexample G', it holds that either $|V(G')| > |V(G)|$, or $|V(G')| = |V(G)|$ and $\tau(G') \geq \tau(G)$. Let $n = |V(G)|$. By definition of a counterexample, $|M_G| > 3^{n/3}$. We will prove a sequence of structural properties of G, and finally conclude that G does not exist, yielding the desired contradiction.

Lemma 5. *G is connected and has at least three vertices.*

Proof. First assume for contradiction that G is not connected. Let $G_1, G_2, \ldots, G_k$ denote the connected components of G. By the choice of G, none of the connected components of G is a counterexample. Hence $|M_{G_i}| \leq 3^{|V(G_i)|/3}$ for each $i \in \{1, \ldots, k\}$. But then $|M_G| = \prod_{i=1}^{k} |M_{G_i}| \leq 3^{n/3}$, contradicting the assumption that G is a counterexample. □

Lemma 6. *Let $u, v \in V(G)$. If there is no maximal induced matching in G that covers both u and v, then u and v are twins.*

Proof. Suppose there is no maximal induced matching in G that covers both u and v. In particular, this implies that u and v are not adjacent. Let $G' = G_{T_G(u)\to v}$ and $G'' = G_{T_G(v)\to u}$. By Lemma 2, we have that $|M_{G'}| \geq |M_G|$ or $|M_{G''}| \geq |M_G|$. Without loss of generality, suppose $|M_{G'}| \geq |M_G|$. The graph G' is triangle-free due to Lemma 3. This, together with the fact that $|M_{G'}| \geq |M_G| > 3^{n/3}$, implies that G' is a counterexample. But by Lemma 4, it holds that $\tau(G') < \tau(G)$, which contradicts the choice of G. □

Lemma 7. *For every edge $uv \in E(G)$ and every set $X \subseteq V(G) \setminus \{u, v\}$, it holds that $|M_G(X, \{u, v\})| \leq 3^{(n-|X \cup N[\{u,v\}]|)/3}$.*

Proof. Let $G' = G - (X \cup N_G[\{u, v\}])$. We show that for every matching M in $M_G(X, \{u, v\})$, it holds that $M \setminus \{uv\} \in M_{G'}$. Let $M \in M_G(X, \{u, v\})$. Since $uv \in E(G)$ and M covers both u and v, the edge uv belongs to M. Since M does not cover any vertex in X, it is clear that the set $M' = M \setminus \{uv\}$ is an induced matching in G'. We now show that M' is maximal. For contradiction, suppose there exists an edge $xy \in E(G')$ such that $M' \cup \{xy\}$ is an induced matching in G'. Since neither x nor y belongs to the set $X \cup N_G[\{u, v\}]$, we have in particular that there is no edge between the sets $\{x, y\}$ and $\{u, v\}$. Hence, adding the edge xy to M yields an induced matching in G, contradicting the assumption that M is a maximal induced matching in G.

We now know that for every matching $M \in M_G(X, \{u, v\})$, it holds that $M \setminus \{uv\} \in M_{G'}$. Note that, for any two matchings $M_1, M_2 \in M_G(X, \{u, v\})$ with $M_1 \neq M_2$, the sets $M_1 \setminus \{uv\}$ and $M_2 \setminus \{uv\}$ are not equal, as both M_1 and M_2 contain the edge uv. Hence we have that $|M_G(X, \{u, v\})| \leq |M_{G'}|$. Since G' has less vertices than G and is thus not a counterexample, we have that $|M_{G'}| \leq 3^{|V(G')|/3} = 3^{(n-|X \cup N[\{u,v\}]|)/3}$. We conclude that $|M_G(X, \{u, v\})| \leq |M_{G'}| \leq 3^{(n-|X \cup N[\{u,v\}]|)/3}$. □

Lemma 8. *G has no vertex of degree less than* 2.

Proof. By Lemma 5, the graph G is connected and $n \geq 3$. Hence, G has no vertices of degree 0. Assume for contradiction that G contains a vertex v with $d(v) = 1$. Let u be the unique neighbor of v. If G is a star, then $M_G = E(G)$, implying that $|M_G| = n - 1 \leq 3^{n/3}$. Since this contradicts the fact that G is a counterexample, we infer that G is not a star. Since G is connected and triangle-free, u has a neighbor w with $d(w) \geq 2$. Note that there is no maximal induced matching in G that covers both v and w. Then u and w must be twins due to Lemma 6. This is a contradiction, as $d(v) < d(w)$ implies that v and w cannot be twins. □

Lemma 9. *G has no* 5*-cycle containing two non-adjacent vertices of degree* 2.

Proof. For contradiction, suppose there is a 5-cycle containing two non-adjacent vertices u and v such that $d(u) = d(v) = 2$. Clearly, the vertices u and v are not twins, and there is no maximal induced matching in G that covers both u and v. This contradicts Lemma 6. □

Lemma 10. *G has no* 4*-cycle containing exactly one vertex of degree* 2.

Proof. Assume, for contradiction, that there exists a 4-cycle $C = uvwx$ such that $d(u) = 2$ and the other vertices of C have degree more than 2. Then u and w are not twins, and there is no maximal induced matching in G that covers both u and w. This contradicts Lemma 6. □

Lemma 11. *G has no two adjacent vertices of degree* 2.

Proof. For contradiction, suppose there are two vertices u and v such that $d(u) = d(v) = 2$ and $uv \in E(G)$. Let a and b denote the other neighbors of u and v, respectively. Since G is triangle-free, we have that $a \neq b$. We first show that $ab \notin E(G)$. For contradiction, assume that $ab \in E(G)$ and both a and b have degree 2. Then G is isomorphic to C_4, implying that $|M_G| = 4 \leq 3^{4/3}$. This contradicts the fact that G is a counterexample. Hence a or b has degree more than 2. Assume without loss of generality that $d(a) \geq 3$. Then a and v are not twins, and there is no matching in M_G covering both a and v. This contradiction to Lemma 6 implies that $ab \notin E$.

We now partition M_G into three sets $M(\emptyset, \{a\})$, $M(\{a\}, \{b\})$, and $M(\{a, b\}, \emptyset)$, and find an upper bound on the size of each of these sets.

We first consider $M(\emptyset, \{a\})$. Clearly, $|M(\emptyset, \{a\})| = \sum_{p \in N(a)} |M(\emptyset, \{a, p\})|$. Let $p = u$. Since $|N[\{a, u\}]| = d(a) + 2$, from Lemma 7 we have $|M(\emptyset, \{a, u\})| \leq 3^{(n-(d(a)+2))/3}$. Now consider the case that $p \neq u$. In this case, $|N[\{a, p\}]| = d(a) + d(p)$ and $d(p) \geq 2$ due to Lemma 8, and thus Lemma 7 implies $|M(\emptyset, \{a, p\})| \leq 3^{(n-(d(a)+2))/3}$. Consequently, we obtain

$$|M(\emptyset, \{a\})| = \sum_{p \in N(a)} |M(\emptyset, \{a, p\})| \leq d(a) \cdot 3^{\frac{n-(d(a)+2)}{3}} .$$

We now find an upper bound on $|M(\{a\}, \{b\})|$. Since no matching in the set $M(\{a\}, \{b\})$ covers u, it holds that $M(\{a\}, \{b\}) = M(\{a, u\}, \{b\})$. We use the fact that $|M(\{a, u\}, \{b\})| = \sum_{q \in N(b)} |M(\{a, u\}, \{b, q\})|$. If $q = v$, then $|M(\{a, u\}, \{b, v\})| \leq 3^{(n-(d(b)+3))/3}$ due to Lemma 7 and the fact that $d(v) = 2$ and $a \notin N[\{b, v\}]$. Let now $q \neq v$. First suppose q is adjacent to a. Then $qauvb$ is a 5-cycle, and hence Lemma 9 implies that $d(q) \geq 3$. Consequently, $|N[\{b, q\}]| = d(b) + d(q) \geq d(b) + 3$, and since $u \notin N[\{b, q\}]$, we find that $|M(\{a, u\}, \{b, q\})| \leq 3^{(n-(d(b)+4))/3}$ due to Lemma 7. Now suppose that q is not adjacent to a. Then $N[\{b, q\}]$ contains neither a nor u. Hence, Lemma 7 and the fact that $|N[\{b, q\}] \geq d(b) + 2$ imply that $|M(\{a, u\}, \{b, q\})| \leq 3^{(n-(d(b)+4))/3}$. We conclude that

$$|M(\{a\}, \{b\})| \leq 3^{\frac{n-(d(b)+3)}{3}} + (d(b) - 1) \cdot 3^{\frac{n-(d(b)+4)}{3}} .$$

Finally, we consider $M(\{a,b\},\emptyset)$. Every matching in $M(\{a,b\},\emptyset)$ is maximal and covers neither a nor b, so it must contain edge uv. Hence, it holds that $M(\{a,b\},\emptyset) = M(\{a,b\},\{u,v\})$. Since $|N[\{u,v\}]| = 4$, Lemma 7 gives

$$|M(\{a,b\},\emptyset)| = |M(\{a,b\},\{u,v\})| \leq 3^{\frac{n-4}{3}}.$$

Combining the obtained upper bounds, we find that

$$|M_G| \leq f(d(a),d(b)) \cdot 3^{\frac{n}{3}},$$

where the function f is defined as follows:

$$f(d(a),d(b)) = d(a) \cdot 3^{-\frac{d(a)+2}{3}} + 3^{-\frac{d(b)+3}{3}} + (d(b)-1) \cdot 3^{-\frac{d(b)+4}{3}} + 3^{-\frac{4}{3}}.$$

Recall that both a and b have degree at least 2 due to Lemma 8. We observe that $f(2,2) < 0.965$, yielding an upper bound of $0.965 \cdot 3^{n/3}$ on $|M_G|$ in case $d(a) = d(b) = 2$. Now consider the case where $d(a) = 2$ and $d(b) \geq 3$. Then the function f is decreasing with respect to $d(b)$. Since $f(2,3) < 0.959$, we find that $|M_G| < 0.959 \cdot 3^{n/3}$ in this case. By using similar arguments, we find that $|M_G| < 0.984 \cdot 3^{n/3}$ when $d(b) = 2$ and $d(a) \geq 3$. Finally, when both $d(a) \geq 3$ and $d(b) \geq 3$, then the function f is decreasing with respect to both variables $d(a)$ and $d(b)$ and is maximum when $d(a) = d(b) = 3$. Since $f(3,3) < 0.978$, we find that $|M_G| < 0.978 \cdot 3^{n/3}$ whenever $d(a) \geq 3$ and $d(b) \geq 3$. Summarizing, we obtain a contradiction to the assumption that $|M_G| > 3^{n/3}$ in each case, which completes the proof of this case. □

Lemma 12. *Let $u \in V(G)$. If u has degree 2, then both its neighbors have degree 3.*

Proof. Suppose u has degree 2, and let $N(u) = \{v,w\}$. Then $d(v) \geq 3$ and $d(w) \geq 3$ due to Lemmas 8 and 11. For contradiction, suppose one of the neighbors of u, say w, has degree more than 3. Note that $M_G = M(\{u\},\emptyset) \uplus M(\emptyset,\{u,v\}) \uplus M(\emptyset,\{u,w\})$. Since $M(\{u\},\emptyset) = M_{G-u}$ and $G-u$ is not a counterexample due to the choice of G, we have $|M(\{u\},\emptyset)| \leq 3^{(n-1)/3}$. Since $|N[\{u,v\}]| = d(v)+2$, we can use Lemma 7 to deduce that $|M(\emptyset,\{u,v\})| \leq 3^{(n-(d(v)+2))/3}$ and $|M(\emptyset,\{u,w\})| \leq 3^{(n-(d(w)+2))/3}$. Hence we find that

$$|M_G| \leq 3^{\frac{n-1}{3}} + 3^{\frac{n-(d(v)+2)}{3}} + 3^{\frac{n-(d(w)+2)}{3}}.$$

Since $d(v) \geq 3$ and $d(w) \geq 4$, we get

$$|M_G| \leq 3^{\frac{n-1}{3}} + 3^{\frac{n-5}{3}} + 3^{\frac{n-6}{3}} = \left(3^{-\frac{1}{3}} + 3^{-\frac{5}{3}} + 3^{-\frac{6}{3}}\right) \cdot 3^{\frac{n}{3}} < 3^{\frac{n}{3}},$$

yielding the desired contradiction. □

Lemma 13. *G has no vertex of degree more than 4.*

Proof. Suppose there is a vertex $u \in V(G)$ such that $d(u) \geq 5$. Due to Lemmas 8 and 12, every neighbor of u has degree at least 3. Clearly, $M_G = M(\emptyset,\{u\}) \uplus M(\{u\},\emptyset)$. To find an upper bound on $|M(\emptyset,\{u\})|$, observe that

$$|M(\emptyset,\{u\})| = \sum_{p \in N(u)} |M(\emptyset,\{u,p\})|.$$

For every $p \in N(u)$, it holds that $|N[\{u,p\}]| = d(u) + d(p)$ and $d(p) \geq 3$. Hence, using Lemma 7, we find that

$$|M(\emptyset,\{u\})| \leq d(u) \cdot 3^{\frac{n-(d(u)+3)}{3}}.$$

Since $M(\{u\},\emptyset) = M_{G-u}$ and $G-u$ is not a counterexample, we have that

$$|M(\{u\},\emptyset)| \leq 3^{\frac{n-1}{3}} .$$

Combining the two upper bounds yields

$$|M_G| \leq d(u) \cdot 3^{\frac{n-(d(u)+3)}{3}} + 3^{\frac{n-1}{3}} = \big(d(u) \cdot 3^{-\frac{d(u)+3}{3}} + 3^{-\frac{1}{3}}\big) \cdot 3^{\frac{n}{3}} .$$

Since $d(u) \geq 5$ by assumption and $d(u) \cdot 3^{-(d(u)+3)/3} + 3^{-1/3} < 1$ whenever $d(u) \geq 5$, we conclude that $|M_G| < 3^{n/3}$. This contradicts the fact that G is a counterexample. □

Lemma 14. *G has no 4-cycle containing a vertex of degree 2.*

Proof. Assume for contradiction that G has a 4-cycle C containing a vertex u of degree 2. Due to Lemmas 10 and Lemma 11, there is exactly one other vertex v in C that has degree 2, and u and v are not adjacent. Let w and y be the other two vertices of C. Since $d(u) = d(v) = 2$, Lemma 12 implies that $d(w) = d(y) = 3$. Let x and z be the neighbors of w and y, respectively, that do not belong to C. We claim that $x \neq z$. For contradiction, suppose $x = z$. Then there is no maximal induced matching in G that covers both u and x. Hence, due to Lemma 6, vertices u and x are twins. In particular $d(x) = 2$, which implies that $V(G) = \{u, v, w, y, x\}$ and $|M_G| = |E(G)| = 6 < 3^{5/3}$, contradicting the fact that G is a counterexample.

Observe that $d(z) \in \{2, 3, 4\}$ due to Lemmas 8 and 13, and $d(x) \geq 2$ due to Lemma 8. In order to find an upper bound on the number of maximal induced matchings in G, we partition M_G as follows:

$$M_G = M(\emptyset,\{w\}) \uplus M(\{w\},\{z\}) \uplus M(\{w,z\},\emptyset) . \tag{1}$$

Since $N(w) = \{u, v, x\}$, it holds that

$$M(\emptyset,\{w\}) = M(\emptyset,\{w,u\}) \uplus M(\emptyset,\{w,v\}) \uplus M(\emptyset,\{w,x\}).$$

Recall that $d(x) \geq 2$. For every $p \in \{u, v, x\}$, it holds that $|N[\{w,p\}]| \geq 5$ and consequently $|M(\emptyset,\{w,p\})| \leq 3^{(n-5)/3}$ due to Lemma 7. Therefore,

$$|M(\emptyset,\{w\})| \leq 3 \cdot 3^{\frac{n-5}{3}} .$$

We now consider $M(\{w,z\},\emptyset)$. Note that every maximal induced matching of G that covers neither w nor z must contain either uy or vy. Hence $|M(\{w,z\},\emptyset)| = |M(\{w,z\},\{u,y\})| + |M(\{w,z\},\{v,y\})|$. Since $|N[\{u,y\}]| = |N[\{v,y\}]| = 5$, we can use Lemma 7 to find that

$$|M(\{w,z\},\emptyset)| \leq 2 \cdot 3^{\frac{n-5}{3}} .$$

It remains to find an upper bound on $|M(\{w\},\{z\})|$. It is clear that $|M(\{w\},\{z\})| = \sum_{q \in N(z)} |M(\{w\},\{q,z\})|$. We first consider $M(\{w\},\{y,z\})$. Since $|N[\{y,z\}]| = d(y) + d(z) = 3 + d(z)$ and $w \notin N[\{y,z\}]$, we have $|M(\{w\},\{z,y\})| \leq 3^{(n-(d(z)+4))/3}$ due to Lemma 7. Now let $q \in N(z) \setminus \{y\}$. Observe that every matching in $M(\{w\},\{q,z\})$ contains edge qz and does not cover w by definition, and hence covers neither u nor v. This means that $M(\{w\},\{q,z\}) = M(\{u,v,w\},\{q,z\})$. Since $d(q) \geq 2$ due to Lemma 8, it holds that $|N[\{q,z\}]| \geq d(z) + 2$. If $q \neq x$, then $\{u,v,w\} \cap N[\{q,z\}] = \emptyset$ and hence $|\{u,v,w\} \cup N[\{q,z\}]| \geq d(z) + 5$. Suppose $q = x$. Then $wvyzx$ is a 5-cycle, and $d(q) \geq 3$ as a result of Lemma 9. Moreover, although now $w \in N[\{q,z\}]$, neither u nor v belongs to $N[\{q,z\}]$. Hence $|\{u,v,w\} \cup N[\{q,z\}]| \geq d(z) + 5$ also in this case. Using Lemma 7, we conclude that $|M(\{w\},\{q,z\})| = |M(\{u,v,w\},\{q,z\})| \leq 3^{n-(d(z)+5)/3}$. Since this holds for every $q \in N(z) \setminus \{y\}$, we find that

$$|M(\{w\},\{z\})| \leq 3^{\frac{n-(d(z)+4)}{3}} + (d(z)-1) \cdot 3^{\frac{n-(d(z)+5)}{3}} .$$

The obtained upper bounds on $|M(\emptyset,\{w\})|$, $|M(\{w\},\{z\})|$, and $|M(\{w,z\},\emptyset)|$, together with (1), yield the following inequality:

$$|M_G| \leq \left(5 \cdot 3^{-\frac{5}{3}} + 3^{-\frac{d(z)+4}{3}} + (d(z)-1) \cdot 3^{-\frac{d(z)+5}{3}}\right) \cdot 3^{\frac{n}{3}} .$$

Recall that $d(z) \in \{2,3,4\}$. Since it can readily be verified that for each value of $d(z) \in \{2,3,4\}$, the above inequality simplifies to $|M_G| < 3^{n/3}$, we obtain the desired contradiction. □

Lemma 15. *G has no vertex of degree* 2*.*

Proof. Due to Lemma 12, in order to prove Lemma 15, it suffices to prove that there is no vertex of degree 3 in G that is adjacent to a vertex of degree 2. For contradiction, suppose G has a vertex u such that $d(u) = 3$ and u is adjacent to at least one vertex of degree 2. Let $N(u) = \{v,w,x\}$. We distinguish two cases, depending on the number of vertices of degree 2 in the neighborhood of u.

Case 1. u has at least two neighbors of degree 2*.*

Without loss of generality, assume that $d(v) = d(x) = 2$. Let $N(x) = \{u,t\}$. Observe that $d(t) = 3$ due to Lemma 12, and $d(w) \in \{2,3,4\}$ due to Lemmas 8 and 13. It is easy to see that we can partition M_G as follows:

$$M_G = M(\emptyset,\{v\}) \uplus M(\{v\},\{w\}) \uplus M(\{v,w\},\{t\}) \uplus M(\{v,w,t\},\emptyset) .$$

First consider $M(\emptyset,\{v\})$. Since every matching in this set contains exactly one of the two edges incident with v, and both neighbors of v have degree exactly 3 due to Lemma 12, we can use Lemma 7 to find that

$$|M(\emptyset,\{v\})| \leq 2 \cdot 3^{\frac{n-5}{3}} .$$

Now consider $M(\{v\},\{w\})$. It is clear that

$$|M(\{v\},\{w\})| = |M(\{v\},\{u,w\})| + \sum_{q \in N(w)\setminus\{u\}} |M(\{v\},\{q,w\})| .$$

Lemma 7, together with the fact that $|N[\{u,w\}]| = d(u) + d(w) = 3 + d(w)$, implies that $|M(\{v\},\{u,w\})| \leq 3^{(n-(d(w)+3))/3}$. Let $q \in N(w) \setminus \{u\}$. Since v has degree 2 and is therefore not contained in a 4-cycle due to Lemma 14, vertex q is not adjacent to v. Hence $|\{v\} \cup N[\{q,w\}]| = 1 + d(q) + d(w) \geq 3 + d(w)$, where we use the fact that $d(q) \geq 2$ due to Lemma 8. This implies that $|M(\{v\},\{q,w\})| \leq 3^{(n-(d(w)+3))/3}$. Since this holds for any $q \in N(w) \setminus \{u\}$, we find that

$$|M(\{v\},\{w\})| \leq d(w) \cdot 3^{\frac{n-(d(w)+3)}{3}} .$$

To find an upper bound on $|M(\{v,w\},\{t\})|$, we first observe that $M(\{v,w\},\{t\}) = M(\{u,v,w\},\{t\})$, as any maximal induced matching that covers neither v nor w but covers t, cannot cover u. Note that $|M(\{u,v,w\},\{t\})| = |M(\{u,v,w\},\{t,x\})| + \sum_{q\in N(t)\setminus\{x\}} |M(\{u,v,w\},\{t,q\})|$. Recall that $d(x) = 2$ and $d(t) = 3$. Since G is triangle-free, x is adjacent to neither v nor w. The same holds for t due to Lemma 14 and the fact that x has degree 2. Hence $|\{u,v,w\} \cup N[\{t,x\}]| = 7$, so $|M(\{u,v,w\},\{t,x\})| \leq 3^{(n-7)/3}$ due to Lemma 7. Let $q \in N(t) \setminus \{x\}$. Then $q \notin \{u,v,w\}$ due to the triangle-freeness of G and Lemma 14. Moreover, neither u nor v is adjacent to q as a result of Lemmas 14 and 9, respectively. Recall that $d(q) \geq 2$ due to Lemma 8. Moreover, if w is adjacent to q, then q

has degree at least 3 by Lemma 9. Hence $|\{u,v,w\} \cup N[\{t,q\}]| \geq 8$, so Lemma 7 implies that $|M(\{u,v,w\},\{t,q\})| \leq 3^{(n-8)/3}$. Since $|N(t) \setminus \{x\}| = 2$, we conclude that

$$|M(\{v,w\},\{t\})| \leq 3^{\frac{n-7}{3}} + 2 \cdot 3^{\frac{n-8}{3}} .$$

Finally, we consider $M(\{v,w,t\},\emptyset)$. Since every matching in this set contains edge ux, we have that $M(\{v,w,t\},\emptyset) = M(\{v,w,t\},\{u,x\})$. Using Lemma 7 and the fact that $|N[\{u,x\}]| = 5$, we deduce that

$$|M(\{v,w,t\},\emptyset)| \leq 3^{\frac{n-5}{3}} .$$

Putting all this together, we obtain the following inequality:

$$|M_G| \leq 3 \cdot 3^{\frac{n-5}{3}} + d(w) \cdot 3^{\frac{n-(d(w)+3)}{3}} + 3^{\frac{n-7}{3}} + 2 \cdot 3^{\frac{n-8}{3}} .$$

It is easy to verify that the right-hand side of this inequality is less than $3^{n/3}$ for every fixed value of $d(w) \in \{2,3,4\}$. This contradicts the assumption that G is a counterexample and completes the proof of Case 1.

Case 2. u has exactly one neighbor of degree 2.

Without loss of generality, assume that $d(x) = 2$. Then $d(v) \geq 3$ and $d(w) \geq 3$ due to Lemma 8. Let $N(x) = \{u,t\}$. We partition M_G as follows:

$$M_G = M(\emptyset,\{t\}) \uplus M(\{t\},\{w\}) \uplus M(\{t,w\},\{v\}) \uplus M(\{t,w,v\},\emptyset) .$$

We first consider $M(\emptyset,\{t\})$. Due to Lemma 12, vertex t has degree 3. If t has at least 2 neighbors of degree 2, then we can apply Case 1 to vertex t to obtain a contradiction. Suppose t has at most one neighbor of degree 2. Since x has degree 2, both vertices in $N(t) \setminus \{x\}$ have degree at least 3. Hence $|N[\{t,x\}] = 5$ and $|N[\{t,q\}]| \geq 6$ for every $q \in N(t) \setminus \{x\}$, and we can apply Lemma 7 to find that

$$|M(\emptyset,\{t\})| \leq 3^{\frac{n-5}{3}} + 2 \cdot 3^{\frac{n-6}{3}} .$$

To find an upper bound on $|M(\{t\},\{w\})|$, we first observe that $M(\{t\},\{w\}) = M(\{t,x\},\{w\})$ due to the fact that no matching in $M(\{t\},\{w\})$ covers x. It is easy to see that $|M(\{t,x\},\{w\})| = |M(\{t,x\},\{w,u\})| + \sum_{q \in N(w)\setminus\{u\}} |M(\{t,x\},\{w,q\})|$. Since x has degree 2, it does not belong to any 4-cycle due to Lemma 14. This implies that $t \notin N[\{u,w\}]$, and hence $|\{t,x\} \cup N[\{w,u\}]| \geq d(w) + d(u) + 1 = d(w) + 4$. Applying Lemma 7 yields $|M(\{t,x\},\{u,w\})| \leq 3^{(n-(d(w)+4))/3}$. Let $q \in N(w)\setminus\{u\}$. By Lemma 8, vertex q has degree at least 2. Note that $q \neq t$ and $x \notin N(q)$ due to Lemma 14, and $d(q) \geq 3$ if $t \in N(q)$ due to Lemma 9. This implies that $|\{t,x\} \cup N[\{w,q\}]| \geq d(w)+4$. By Lemma 7, we have that $|M(\{t,x\},\{w,q\})| \leq 3^{(n-(d(w)+4))/3}$. Hence we conclude that

$$|M(\{t\},\{w\})| = |M(\{t,x\},\{w\})| \leq d(w) \cdot 3^{\frac{n-(d(w)+4)}{3}} .$$

Now consider $M(\{t,w\},\{v\})$. No matching in this set covers x and therefore we have $M(\{t,w\},\{v\}) = M(\{t,w,x\},\{v\})$. Clearly it holds that

$$|M(\{t,w,x\},\{v\})| = |M(\{t,w,x\},\{v,u\})| + \sum_{q \in N(v)\setminus\{u\}} |M(\{t,w,x\},\{v,q\})|.$$

Observe that $t \notin N[\{v,u\}]$ due to Lemma 14 and the fact that G is triangle-free. Hence $\{t,w,x\} \cup |N[\{v,u\}]| = d(v) + d(u) + 1 = d(v) + 4$, so we can apply Lemma 7 to find that

$|M(\{t,w,x\},\{v,u\})| \leq 3^{(n-(d(v)+4))/3}$. Let $q \in N(v) \setminus \{u\}$. Due to the triangle-freeness of G and Lemma 14, vertex x does not belong to $N[\{v,q\}]$, and neither t nor w belongs to $N[v]$. We claim that $|\{t,w,x\} \cup N[\{v,q\}]| \geq d(v)+5$. This is immediately clear if neither t nor w belongs to $N[q]$, as $d(q) \geq 2$. Suppose both t and w belong to $N[q]$. If $d(q) \leq 3$, then there is no maximal induced matching in G that covers both u and q. Since u and q are not twins, this contradicts Lemma 6. Hence $d(q) \geq 4$, implying that $|\{t,w,x\} \cup N[\{v,q\}]| \geq d(v)+5$. If exactly one of the vertices t and w belongs to $N[q]$, then $d(q) \geq 3$ as a result of Lemma 9 and Lemma 14, respectively. Hence we have that $|\{t,w,x\} \cup N[\{v,q\}]| \geq d(v)+5$ also in this case. We can now invoke Lemma 7 to find that $\sum_{q\in N(v)\setminus\{u\}} |M(\{t,w,x\},\{v,q\})| \leq (d(v)-1)\cdot 3^{(n-(d(v)+5))/3}$, and we can thus conclude that

$$|M(\{t,w\},\{v\})| = |M(\{t,w,x\},\{v\})| \leq 3^{\frac{n-(d(v)+4)}{3}} + (d(v)-1)\cdot 3^{\frac{n-(d(v)+5)}{3}}\,.$$

Finally, we consider $M(\{t,w,v\},\emptyset)$. Since every maximal induced matching in this set must contain edge ux, it holds that $M(\{t,w,v\},\emptyset) = M(\{t,w,v\},\{u,x\})$. The fact that $|N[\{u,x\}]| = 5$ together with Lemma 7 readily implies that

$$|M(\{t,w,v\},\emptyset)| = |M(\{t,w,v\},\{u,x\})| \leq 3^{(n-5)/3}\,.$$

Combining the obtained upper bounds yields the following inequality:

$$|M_G| \leq 2\cdot 3^{\frac{n-5}{3}} + 2\cdot 3^{\frac{n-6}{3}} + d(w)\cdot 3^{\frac{n-(d(w)+4)}{3}} + 3^{\frac{n-(d(v)+4)}{3}} + (d(v)-1)\cdot 3^{\frac{n-(d(v)+5)}{3}}\,.$$

Recall that $d(v) \geq 3$ and $d(w) \geq 3$. We also have that $d(v) \leq 4$ and $d(w) \leq 4$ as a result of Lemma 13. It is therefore easy to check that the right-hand side of this inequality is less than $3^{n/3}$. This contradicts the fact that G is a counterexample and completes the proof of the lemma. □

Lemma 16. *G is cubic.*

Proof. Due to Lemmas 8, 13, and 15, every vertex in G has degree 3 or 4. Hence, in order to prove Lemma 16, it suffices to prove that G has no vertex of degree 4. For contradiction, suppose there exists a vertex u such that $d(u) = 4$. Let v be a neighbor of u. To find an upper bound on $|M_G|$, we partition M_G into two sets $M(\emptyset,\{v\})$ and $M(\{v\},\emptyset)$ and find upper bounds on the sizes of these sets.

Observe that $|M(\emptyset,\{v\})| = \sum_{q\in N(v)} |M(\emptyset,\{v,q\})|$. If $q = u$, then $|N[\{v,q\}]| = d(v)+4$ and hence $|M(\emptyset,\{u,v\})| \leq 3^{(n-(d(v)+4))/3}$ by Lemma 7. For any vertex $q \in N(v)\setminus\{u\}$, the fact that $|N[\{q,v\}]| \geq d(v)+3$ together with Lemma 7 implies that $|M(\emptyset,\{q,v\})| \leq 3^{(n-(d(v)+3))/3}$. Hence we find that

$$|M(\emptyset,\{v\})| \leq 3^{\frac{n-(d(v)+4)}{3}} + (d(v)-1)\cdot 3^{\frac{n-(d(v)+3)}{3}}\,.$$

Since $M(\{v\},\emptyset) = M_{G-v}$ and $G-v$ is not a counterexample, we have that

$$|M(\{v\},\emptyset)| \leq 3^{\frac{n-1}{3}}\,.$$

Hence we conclude that

$$|M_G| \leq 3^{\frac{n-(d(v)+4)}{3}} + (d(v)-1)\cdot 3^{\frac{n-(d(v)+3)}{3}} + 3^{\frac{n-1}{3}}\,.$$

For every fixed value of $d(v) \in \{3,4\}$, it can easily be verified that $|M_G| \leq 3^{n/3}$, yielding the desired contradiction. □

Lemma 17. *Let $u, v \in V(G)$. If u and v are contained in a 5-cycle C, then u and v have no common neighbor in $V(G) \setminus V(C)$.*

Proof. Suppose there is a 5-cycle C containing both u and v. Then u and v are not twins. If u and v are adjacent, then they have no common neighbor due to the fact that G is triangle-free. Suppose u and v are non-adjacent, and, for contradiction, assume there is a vertex $x \in V(G) \setminus V(C)$ such that x is adjacent to both u and v. Since G is cubic due to Lemma 16, there is no maximal induced matching in G that covers both u and v. Hence u and v must be twins due to Lemma 6, yielding the desired contradiction. $\square$

Lemma 18. *G contains at least one 4-cycle.*

Proof. Assume for contradiction that G contains no 4-cycle. Let u be an arbitrary vertex in G. Recall that G is cubic due to Lemma 16. Let $N(u) = \{v, w, x\}$ and $N(x) = \{u, s, t\}$. We consider the following partition of M_G:

$$M(\emptyset, \{v\}) \uplus M(\{v\}, \{w\}) \uplus M(\{v, w\}, \{s\}) \uplus M(\{v, w, s\}, \{t\}) \uplus M(\{v, w, s, t\}, \emptyset)\,.$$

Since $d(v) = 3$ and the closed neighborhood of any edge incident with v consists of six vertices, Lemma 7 implies that

$$|M(\emptyset, \{v\})| \le 3 \cdot 3^{\frac{n-6}{3}}\,.$$

Let us consider $M(\{v\}, \{w\})$. Note that $|M(\{v\}, \{w\})| = \sum_{q \in N(w)} |M(\{v\}, \{w, q\})|$. If $q = u$, then $|M(\{v\}, \{w, q\})| \le 3^{(n-6)/3}$ due to Lemma 7 and that fact that $|N[\{w, q\}]| = 6$. Now suppose $q \in N(w) \setminus \{u\}$. The assumption that there is no 4-cycle in G implies that $v \notin N[\{w, q\}]$, and consequenly $|M(\{v\}, \{w, q\})| \le 3^{(n-7)/3}$ due to Lemma 7. We therefore have that

$$|M(\{v\}, \{w\})| \le 3^{\frac{n-6}{3}} + 2 \cdot 3^{\frac{n-7}{3}}\,.$$

We now find an upper bound on $|M(\{v, w\}, \{s\})|$. Let $M \in M(\{v, w\}, \{s\})$. Observe that M covers neither v nor w, but covers s. Since M is an induced matching, it cannot cover u. This implies that $M(\{v, w\}, \{s\}) = M(\{u, v, w\}, \{s\})$. Clearly, $|M(\{u, v, w\}, \{s\})| = \sum_{q \in N(s)} |M(\{u, v, w\}, \{s, q\})|$.

We claim that $|\{u, v, w\} \cup N[\{s, q\}]| \ge 8$ for any $q \in N(s)$. Suppose $q = x$. Since G is triangle-free and has no 4-cycles by assumption, neither v nor w belongs to $N[\{s, q\}]$, so $|\{u, v, w\} \cup N[\{s, q\}]| = 8$ in this case. Now suppose $q \in N(s) \setminus \{x\}$. Since G has no triangles and no 4-cycles, none of the vertices in $\{u, v, w\}$ belongs to $N[s]$. For the same reason, $u \notin N[q]$ and q is not adjacent to both v and w. Hence $|\{u, v, w\} \cup N[\{s, q\}]| \ge 8$ also in this case. Lemma 7 now implies that

$$|M(\{v, w\}, \{s\})| = |M(\{u, v, w\}, \{s\})| \le 3 \cdot 3^{\frac{n-8}{3}}\,.$$

Let us now consider $M(\{v, w, s\}, \{t\})$. Clearly, it holds that

$$|M(\{v, w, s\}, \{t\})| = \sum_{q \in N(t)} |M(\{v, w, s\}, \{t, q\})|.$$

Similar to the previous paragraph, we can use the assumption that G contains neither triangles nor 4-cycles to deduce that

$$|M(\{v, w, s\}, \{t\})| = |M(\{u, v, w, s\}, \{t\})| \le 3^{\frac{n-8}{3}} + 2 \cdot 3^{\frac{n-9}{3}}\,.$$

It remains to consider $M(\{v,w,s,t\},\emptyset)$. Since every maximal induced matching in this set contains edge ux, we have that $M(\{v,w,s,t\},\emptyset) = M(\{v,w,s,t\},\{u,x\})$. Since $|N[\{u,x\}]| = 6$, Lemma 7 implies that

$$|M(\{v,w,p,q\},\emptyset)| = |M(\{v,w,p,q\},\{u,x\})| \le 3^{\frac{n-6}{3}}.$$

We conclude that

$$|M_G| \le 5 \cdot 3^{\frac{n-6}{3}} + 2 \cdot 3^{\frac{n-7}{3}} + 4 \cdot 3^{\frac{n-8}{3}} + 2 \cdot 3^{\frac{n-9}{3}} < 3^{\frac{n}{3}},$$

yielding the desired contradiction. □

Lemma 19. *G is isomorphic to $K_{3,3}$.*

Proof. Let uv be an edge of G such that no edge in $E(G) \setminus \{uv\}$ is contained in more 4-cycles than uv is. Since u and v are adjacent and G is triangle-free, u and v have no common neighbor. Recall that G is cubic due to Lemma 16. Let $N(u) = \{a,d\}$ and $N(v) = \{b,c\}$. It is easy to see that edge uv is contained in at most four 4-cycles.

If uv is contained in exactly four 4-cycles, then G is isomorphic to $K_{3,3}$ and the lemma holds. Suppose uv is contained in at most three 4-cycles. Due to Lemma 18 and the choice of uv, edge uv belongs to at least one 4-cycle. Hence, there is at least one edge between sets $\{a,d\}$ and $\{b,c\}$. Note that $ad \notin E(G)$ and $bc \notin E(G)$, as G is triangle-free. We distinguish four cases, depending on the adjacencies between vertices in $\{a,d\}$ and $\{b,c\}$.

Case 1: $ab \in E(G)$ and $ac, db, dc \notin E(G)$.
Case 2: $ab, ac \in E(G)$ and $db, dc \notin E(G)$.
Case 3: $ab, cd \in E(G)$ and $ac, db \notin E(G)$.
Case 4: $ab, ac, bd \in E(G)$ and $dc \notin E(G)$.

Note that uv belongs to exactly one 4-cycle in Case 1, to exactly two 4-cycles in Cases 2 and 3, and to exactly three 4-cycles in Case 4.

We partiton M_G into five sets as follows:

$$M_G = M(\emptyset,\{a\}) \uplus M(\{a\},\{b\}) \uplus M(\{a,b\},\{c\}) \uplus M(\{a,b,c\},\{d\}) \uplus M(\{a,b,c,d\},\emptyset).$$

In Claims 1–5 below, we prove upper bounds on the sizes of the five sets in the right-hand side of the above expression. We then combine these five upper bounds in order to obtain an upper bound on $|M_G|$.

Claim 1. $|M(\emptyset,\{a\})| \le 3 \cdot 3^{(n-6)/3}$.

Since G is cubic due to Lemma 16, the closed neighborhood of any of the three edges incident with a consists of six vertices. Hence Lemma 7 ensures that $|M(\emptyset,\{a,q\})| \le 3^{(n-6)/3}$ for every $q \in N(a)$, implying the upper bound given in Claim 1.

Claim 2. $|M(\{a\},\{b\})| \le 2 \cdot 3^{(n-6)/3}$.

Note that in all four cases, ab belongs to $E(G)$. By definition, there is no matching in $M(\{a\},\{b\})$ that contains ab. For any of the other two edges incident with b, its closed neighborhood has size 6. Hence the correctness of the claimed upper bound again follows from Lemma 7.

Claim 3. $|M(\{a,b\},\{c\})| \le 3^{(n-6)/3} + 3^{(n-7)/3}$.

First we consider Case 1. In this case, the closed neighborhood of cv contains vertex b and it does not contain a. Therefore, $|\{a,b\} \cup N[\{c,v\}]| = 7$ and consequently, by Lemma 7 we have that $|M(\{a,b\},\{c,v\})| \leq 3^{(n-7)/3}$. Let cq be one of the other edges incident with c. Recall that uv belongs to exactly one 4-cycle in Case 1. Hence, vertex q is not adjacent to b, as otherwise bv belongs to two 4-cycles, contradicting the choice of uv. We claim that q is not adjacent to a. For contradiction, suppose q is adjacent to a. Then $qabvc$ is a 5-cycle containing a and v, so a and v cannot have a common neighbor in $V(G) \setminus \{q,a,b,v,c\}$ due to Lemma 17. The fact that both a and v are adjacent to u gives the desired contradiction. Hence, for any $q \in N(c) \setminus \{v\}$, we have that $|\{a,b\} \cup N[\{c,q\}]| = 8$, and thus Lemma 7 implies that $|M(\{a,b\},\{c,q\})| \leq 3^{(n-8)/3}$. We obtain that $|M(\{a,b\},\{c\})| \leq 3^{(n-7)/3} + 2 \cdot 3^{(n-8)/3}$, which is strictly smaller than $3^{(n-6)/3} + 3^{(n-7)/3}$.

Let us now consider Case 2. Observe that no matching in $M(\{a,b\},\{c\})$ contains edge ac. Hence $|M(\{a,b\},\{c\})| = |M(\{a,b\},\{c,v\})| + |M(\{a,b\},\{c,q\})|$, where q is the neighbor of c other than v and a. Both vertices a and b belong to $N[\{c,v\}]$ and hence $|\{a,b\} \cup N[\{c,v\}]| = 6$. Therefore, Lemma 7 guarantees that $|M(\{a,b\},\{c,v\})| \leq 3^{(n-6)/3}$. Note that $a \notin N[\{c,q\}]$. We claim that $b \notin N[\{c,q\}]$. For contradiction, suppose $b \in N[\{c,q\}]$. Then b is adjacent to q, and hence bv belongs to three 4-cycles, namely $bvua$, $bvcq$, and $bvca$. Since uv belongs to only two 4-cycles in Case 2, this contradicts the choice of uv. Hence, we have that $|\{a,b\} \cup N[\{c,q\}]| = 7$ and consequently $|M(\{a,b\},\{c,q\})| \leq 3^{(n-7)/3}$ by Lemma 7. We can now conclude that $|M(\{a,b\},\{c\})| \leq 3^{(n-6)/3} + 3^{(n-7)/3}$.

For Case 3, let q be the neighbor of c other than d and v. Since $b \in N[\{c,v\}]$ and $a \notin N[\{c,v\}]$ in Case 3, we have that $|\{a,b\} \cup N[\{c,v\}]| = 7$ and therefore $|M(\{a,b\},\{c,v\})| \leq 3^{(n-7)/3}$ due to Lemma 7. Moreover, since $N[\{c,d\}]$ contains neither a nor b, Lemma 7 implies that $|M(\{a,b\},\{c,d\})| \leq 3^{(n-8)/3}$. We now consider edge cq. Since no matching in $M(\{a,b\},\{c,q\})$ covers u, it holds that $M(\{a,b\},\{c,q\}) = M(\{a,b,u\},\{c,q\})$. Recall that $N(u) = \{v,a,d\}$, so $q \notin N[u]$. For contradiction, suppose $a \in N[q]$. Then $qauvc$ is a 5-cycle containing two vertices, namely u and c, that have a common neighbor, namely d, in the set $V(G) \setminus \{q,a,u,v,c\}$. This contradicts Lemma 17, so we conclude that $a \notin N[q]$. Consequently, we have that $|\{a,b,u\} \cup N[\{c,p\}]| = 8$, so Lemma 7 implies that $|M(\{a,b\},\{c,p\})| \leq 3^{(n-8)/3}$. We conclude that $|M(\{a,b\},\{c\})| \leq 3^{(n-7)/3} + 2 \cdot 3^{(n-8)/3} < 3^{(n-6)/3} + 3^{(n-7)/3}$.

Finally, we consider Case 4. Let $N(c = \{a,v,q\}$. Since no matching in $M(\{a,b\},\{c\})$ covers a, we have that $|M(\{a,b\},\{c\})| = |M(\{a,b\},\{c,v\})| + |M(\{a,b\},\{c,q\})|$. The fact that $|N[\{c,v\}]| = 6$ together with Lemma 7 implies that $|M(\{a,b\},\{c,v\})| \leq 3^{(n-6)/3}$. Since $N(a) = \{u,b,c\}$ and $N(b) = \{v,a,c\}$ in this case, neither a nor b belongs to $N[\{c,q\}]$. Hence, $|\{a,b\} \cup N[\{c,q\}]| = 8$ and using Lemma 7, we deduce that $|M(\{a,b\},\{c,q\})| \leq 3^{(n-8)/3}$. We can therefore conclude that $|M(\{a,b\},\{c\})| \leq 3^{(n-6)/3} + 3^{(n-8)/3} < 3^{(n-6)/3} + 3^{(n-7)/3}$.

Claim 4. $|M(\{a,b,c\},\{d\})| \leq 3^{(n-7)/3} + 3^{(n-8)/3}$.

First we consider Cases 1 and 2. In both of these cases, the closed neighborhood of du contains a, but neither b nor c belong to $N[\{d,u\}]$. Therefore, $|\{a,b,c\} \cup N[\{d,u\}]| = 8$ and consequently, by Lemma 7, we have that $|M(\{a,b,c\},\{d,u\})| \leq 3^{(n-8)/3}$. Let $q \in N(d) \setminus \{u\}$. Since no matching in the set $M(\{a,b,c\},\{d,q\})$ covers vertex v, we have that $M(\{a,b,c\},\{d,q\}) = M(\{a,b,c,v\},\{d,q\})$. Edge au belongs to all the 4-cycles to which uv belongs. By the choice of uv, edge au does not belong to any other 4-cycle. In particular, the vertices $\{a,q,d,u\}$ do not induce a C_4, which implies that $a \notin N[q]$. We claim that b is also not adjacent to q. For contradiction, suppose $b \in N[q]$. Then vertices b and u are contained in a 5-cycle, namely $bpduv$, so the fact that they are adjacent to a contradicts Lemma 17. Finally, we observe that $v \notin N[q]$, since $N(v) = \{u,b,c\}$. From this, we deduce that $|\{a,b,c,v\} \cup N[\{d,q\}]| = 9$, and hence $|M(\{a,b,c\},\{d,q\})| \leq 3^{(n-9)/3}$ due to Lemma 7. We conclude that $|M(\{a,b,c\},\{d\})| \leq 3^{(n-8)/3} + 2 \cdot 3^{(n-9)/3}$, which is less than $3^{(n-7)/3} + 3^{(n-8)/3}$.

Now we consider Case 3. Since no matching in $M(\{a,b,c\},\{d\})$ contains edge dc, we have that $|M(\{a,b,c\},\{d\})| = |M(\{a,b,c\},\{d,u\})| + |M(\{a,b,c\},\{d,q\})|$, where q is the neighbor of d other than c and u. In Case 3, both vertices a and c are in $N[\{d,u\}]$ and $b \notin N[\{d,u\}]$. Hence $|\{a,b,c\} \cup N[\{d,u\}]| = 7$, and therefore Lemma 7 implies that $|M(\{a,b,c\},\{d,u\})| \leq 3^{(n-7)/3}$. From the observation that no matching in $M(\{a,b,c\},\{d\})$ covers v, it follows that $M(\{a,b,c\},\{d,q\}) = M(\{a,b,c,v\},\{d,q\})$. We claim that neither v nor b belong to $N[\{d,q\}]$. The fact that neither v nor b belongs to $N[d]$ follows from the triangle-freeness of G and the fact that we are in Case 3. Moreover, since $N(v) = \{u,b,c\}$, we have that $v \notin N[q]$. For contradiction, suppose $b \in N[q]$. Then the vertices $\{q,b,v,u,d\}$ induce a 5-cycle. Vertices b and u lie on this 5-cycle and have a common neighbor, namely a, in $V(G) \setminus \{q,b,v,u,d\}$. This contradiction to Lemma 17 implies that $\{v,b\} \cap N[\{d,q\}] = \emptyset$. Hence $|\{a,b,c,v\} \cup N[\{d,q\}]| \geq 8$, and we can use Lemma 7 to find that $M(\{a,b,c\},\{d,q\})| \leq 3^{(n-8)/3}$. Consequently, we have that $|M(\{a,b,c\},\{d\})| \leq 3^{(n-7)/3} + 3^{(n-8)/3}$.

It remains to consider Case 4. Let q be the neighbor of d other than b and u. Since edge db is not contained in any of the matchings in $M(\{a,b,c\},\{d\})$, we have that $|M(\{a,b,c\},\{d\}| = |M(\{a,b,c\},\{d,u\})| + |M(\{a,b,c\},\{d,q\})|$. Since $c \notin N[\{d,u\}]$, we have that $|\{a,b,c\} \cup N[\{d,u\}]| = 7$ and hence $|M(\{a,b,c\},\{d,u\})| \leq 3^{(n-7)/3}$ due to Lemma 7. Observe that vertex v is not covered by any matching in $M(\{a,b,c\},\{d,q\})$, which implies that $M(\{a,b,c\},\{d,q\}) = M(\{a,b,c,v\},\{d,q\})$. Since $N(a) = N(v) = \{u,b,c\}$, we have that neither a nor v belongs to $N[\{d,q\}]$. We now show that c does not belong to $N[\{d,q\}]$ either. For contradiction, suppose otherwise. Since c is not adjacent to d in Case 4, vertex c must be adjacent to q. Hence the vertices $\{c,q,d,u,v\}$ induce a 5-cycle. By Lemma 17, no two vertices on this cycle have a common neighbor outside the cycle, contradicting the fact that both u and c are adjacent to a. This implies that $|\{a,b,c,v\} \cup N[\{d,q\}]| \geq 9$, so $|M(\{a,b,c\},\{d,q\})| \leq 3^{(n-9)/3}$ due to Lemma 7. We conclude that $|M(\{a,b,c\},\{d\})| \leq 3^{(n-7)/3} + 3^{(n-9)/3}$, which is clearly upper bounded by $3^{(n-7)/3} + 3^{(n-8)/3}$.

Claim 5. $|M(\{a,b,c,d\},\emptyset)| \leq 3^{(n-6)/3}$.

Every maximal induced matching in G that does not cover any vertex in $\{a,b,c,d\}$ contains edge uv. Therefore, $M(\{a,b,c,d\},\emptyset) = M(\{a,b,c,d\},\{u,v\})$. Since $|N[\{u,v\}]| = 6$ due to the fact that G is cubic by Lemma 16, it follows from Lemma 7 that $|M(\{a,b,c,d\},\{u,v\})| \leq 3^{(n-6)/3}$. This completes the proof of Claim 5.

Combining the upper bounds in Claims 1–5 yields the following:

$$|M_G| \leq 7 \cdot 3^{\frac{n-6}{3}} + 2 \cdot 3^{\frac{n-7}{3}} + 3^{\frac{n-8}{3}} < 3^{\frac{n}{3}} .$$

This contradicts the assumption that G is a counterexample. □

Lemma 19 states that G is isomorphic to $K_{3,3}$, so in particular $n = 6$. Since every maximal induced matching in $K_{3,3}$ consists of a single edge, we have that $|M_G| = |E(K_{3,3})| = 9 = 3^{n/3}$, contradicting the assumption that G is a counterexample. This contradiction implies that every triangle-free graph on n vertices has at most $3^{n/3}$ maximal induced matchings.

It remains to show that the bound in Theorem 1 is best possible. Let G be the disjoint union of p copies of $K_{3,3}$ for some positive integer p. Every maximal induced matching in G contains exactly one edge of each connected component of G, which implies that $|M_G| = 9^p = 9^{n/6} = 3^{n/3}$. This completes the proof of Theorem 1.

References

1. M. Basavaraju, P. Heggernes, P. van 't Hof, R. Saei, and Y. Villanger. Maximal induced matchings in triangle-free graphs. In: D. Kratsch, I. Todinca (eds.), Proc. WG 2014, LNCS, Springer, to appear.

2. J. M. Byskov. Enumerating maximal independent sets with applications to graph colouring. Operation Research Letters 32: 547–556 (2004).
3. K. Cameron. Induced matchings. Discr. Appl. Math. 24: 97–102 (1989).
4. R. Diestel. Graph Theory, Electronic Edition. Springer-Verlag, (2005).
5. S. Gupta, V. Raman, and S. Saurabh. Maximum r-regular induced subgraph problem: Fast exponential algorithms and combinatorial bounds. SIAM J. Disc. Math. 26: 1758–1780 (2012).
6. H. Hocquard, P. Ochem, and P. Valicov. Strong edge-colouring and induced matchings. Inf. Proc. Lett. 113: 836–843 (2013).
7. M. Hujter and Zs. Tuza. The number of maximal independent sets in triangle-free graphs. SIAM J. Disc. Math. 6: 284–288 (1993).
8. D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. On generating all maximal independent sets. Information Processing Letters, 27: 119–123 (1988).
9. J. W. Moon and L. Moser. On cliques in graphs. Israel J. Math. 3: 23–28 (1965).
10. S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. A new algorithm for generating all the maximal independent sets. SIAM Journal on Computing, 6: 505–517 (1977).

Chapter 12

Graph Classes and Ramsey numbers

Rémy Belmonte, Pinar Heggernes, Pim van't Hof, Arash Rafiey, and Reza Saei

Discrete Applied Mathematics 173 (2014), 16-27.
(A preliminary version appeared (without author Arash Rafiey) in COCCON 2012 (best paper award), LNCS 7434: 204 - 215, Springe.)

Graph Classes and Ramsey Numbers*

Rémy Belmonte, Pinar Heggernes, Pim van 't Hof, Arash Rafiey, and Reza Saei

Department of Informatics, University of Bergen, Norway†

Abstract. For a graph class $\mathcal{G}$ and any two positive integers i and j, the Ramsey number $R_{\mathcal{G}}(i,j)$ is the smallest positive integer such that every graph in $\mathcal{G}$ on at least $R_{\mathcal{G}}(i,j)$ vertices has a clique of size i or an independent set of size j. For the class of all graphs, Ramsey numbers are notoriously hard to determine, and they are known only for very small values of i and j. Even if we restrict $\mathcal{G}$ to be the class of claw-free graphs, it is highly unlikely that a formula for determining $R_{\mathcal{G}}(i,j)$ for all values of i and j will ever be found, as there are infinitely many nontrivial Ramsey numbers for claw-free graphs that are as difficult to determine as for arbitrary graphs. Motivated by this difficulty, we establish here exact formulas for all Ramsey numbers for three important subclasses of claw-free graphs: line graphs, long circular interval graphs, and fuzzy circular interval graphs. On the way to obtaining these results, we also establish all Ramsey numbers for the class of perfect graphs. Such positive results for graph classes are rare: a formula for determining $R_{\mathcal{G}}(i,j)$ for all values of i and j, when $\mathcal{G}$ is the class of planar graphs, was obtained by Steinberg and Tovey (J. Comb. Theory Ser. B 59 (1993), 288–296), and this seems to be the only previously known result of this kind. We complement our aforementioned results by giving exact formulas for determining all Ramsey numbers for several graph classes related to perfect graphs.

1 Introduction

Ramsey Theory is an important subfield of combinatorics that studies how large a system must be in order to ensure that it contains some particular structure. Since the start of the field in 1930 [22], there has been a tremendous interest in Ramsey Theory, leading to many results as well as several surveys and books (see, e.g., [17] and [21]). For every pair of positive integers i and j, the *Ramsey number* $R(i,j)$ is the smallest positive integer such that every graph on at least $R(i,j)$ vertices contains a clique of size i or an independent set of size j. Ramsey's Theorem [22], in its graph-theoretic version, states that the number $R(i,j)$ exists for every pair of positive integers i and j. As discussed by Diestel ([11], p. 252), this result might seem surprising at first glance. Even more surprising is how difficult it is to determine these values exactly; despite the vast amount of results that have been produced on Ramsey Theory during the past 80 years, we still do not know the exact value of, for example, $R(4,6)$ or $R(3,10)$ [21]. This difficulty is most adequately addressed by the following quote, attributed to Paul Erdős [23]: *"Imagine an alien force, vastly more powerful than us, landing on Earth and demanding the value of* $R(5,5)$ *or they will destroy our planet. In that case, we should marshal all our computers and all our mathematicians and attempt to find the value. But suppose, instead, that they ask for* $R(6,6)$*. In that case, we should attempt to destroy the aliens."* During the last two decades, with the use of computers, lower and upper bounds have been established for more and more Ramsey numbers. However, no more than 16 nontrivial Ramsey numbers have been determined exactly (see Table 1).

*This work is supported by the Research Council of Norway.

†Email addresses: remy.belmonte@ii.uib.no, pinar.heggernes@ii.uib.no, pim.vanthof@ii.uib.no, arashr@sfu.ca, reza.saeidinvar@ii.uib.no.

$i \backslash j$	3	4	5	6	7	8	9	10
3	6	9	14	18	23	28	36	40–42
4	9	18	25	36–41	49–61	56–84	73–115	92–149
5	14	25	43–49	58–87	80–143	101–216	126–316	144–442
6	18	36–41	58–87	102–165	113–298	132–495	169–780	179–1171
7	23	49–61	80–143	113–298	205–540	217–1031	241–1713	289–2826
8	28	56–84	101–216	132–495	217–1031	282–1870	317–3583	330–6090
9	36	73–115	126–316	169–780	241–1713	317–3583	565–6588	581–12677
10	40–42	92–149	144–442	179–1171	289–2826	330–6090	581–12677	798–23556

Table 1. Trivially, it holds that $R(1, j) = 1$ and $R(2, j) = j$ for all $j \geq 1$, and $R(i, j) = R(j, i)$ for all $i, j \geq 1$. The above table contains the currently best known upper and lower bounds on $R(i, j)$ for all $i, j \in \{3, \ldots, 10\}$ [5,12,15,21]. In particular, it contains all nontrivial Ramsey numbers whose exact values are known.

Confronted with such difficulty, it is natural to restrict the set of considered graphs. For any graph class $\mathcal{G}$ and any pair of positive integers i and j, we define $R_{\mathcal{G}}(i, j)$ to be the smallest positive integer such that every graph in $\mathcal{G}$ on at least $R_{\mathcal{G}}(i, j)$ vertices contains a clique of size i or an independent set of size j. To the best of our knowledge, Ramsey numbers of this type have been studied previously only for planar graphs, graphs with small maximum degree, and claw-free graphs. Let us briefly summarize the known results on these classes.

Planar graphs form the only graph class for which *all* Ramsey numbers have been determined exactly. Let $\mathcal{P}$ denote the class of planar graphs. Trivially, $R_{\mathcal{P}}(1, j) = R_{\mathcal{P}}(i, 1) = 1$ for all $i, j \geq 1$. The following theorem establishes all other Ramsey numbers for planar graphs. The theorem is due to Steinberg and Tovey [25], and its proof relies on the famous four color theorem.

Theorem 1 ([25]). *Let $\mathcal{P}$ be the class of planar graphs. Then*

- $R_{\mathcal{P}}(2, j) = j$ *for all* $j \geq 2$,
- $R_{\mathcal{P}}(3, j) = 3j - 3$ *for all* $j \geq 2$,
- $R_{\mathcal{P}}(i, j) = 4j - 3$ *for all* $i \geq 4$ *and* $j \geq 2$ *such that* $(i, j) \neq (4, 2)$, *and*
- $R_{\mathcal{P}}(4, 2) = 4$.

It is interesting to note that almost 25 years before Steinberg and Tovey published their result, Walker [26] established exact values and bounds on all Ramsey numbers for planar graphs, using Heawood's five color theorem. In fact, Walker proved the exact values of all Ramsey numbers for planar graphs, assuming the validity of the –then– four color conjecture.

For any positive integer k, let $\mathcal{G}_k$ be the class of graphs with maximum degree at most k. Staton [24] calculated the exact value of $R_{\mathcal{G}_3}(3, j)$ for all $j \geq 1$, while the Ramsey numbers $R_{\mathcal{G}_3}(4, j)$ for all $j \geq 1$ were obtained by Fraughnaugh and Locke [14]. In the same paper, Fraughnaugh and Locke [14] also determined the exact value of $R_{\mathcal{G}_4}(4, j)$ for all $j \geq 1$, while the numbers $R_{\mathcal{G}_4}(3, j)$ for all $j \geq 1$ had previously been obtained by Fraughnaugh Jones [13].

The only other graph class that has been studied in this context is the class $\mathcal{C}$ of claw-free graphs. Matthews [19] proved exact values as well as upper and lower bounds on some Ramsey numbers for claw-free graphs. In particular, he established the exact value of $R_{\mathcal{C}}(3, j)$ for all $j \geq 1$. Perhaps more interestingly, he observed that $R_{\mathcal{C}}(i, 3) = R(i, 3)$ for all $i \geq 1$. Since the exact value of $R(i, 3)$ is unknown for every $i \geq 10$ (see also Table 1), this implies that there is little hope of finding a formula for determining all Ramsey numbers for claw-free graphs.

Inspired by the contrast between the positive result on planar graphs and the negative result on claw-free graphs, we initiate a systematic study of Ramsey numbers for graph classes. We refer the reader to Figure 1 for an overview of the inclusion relationships between the graph classes mentioned below and an overview of our results. First, in Section 3, we show that the mentioned negative result for claw-free graphs also holds for several other graph classes, such as triangle-free graphs, AT-free graphs, and P_5-free graphs, to name but a few. On the positive side, in the same section, we determine all Ramsey numbers for three important[1] subclasses of claw-free graphs: line graphs, long circular interval graphs, and fuzzy circular interval graphs. To prove the latter results, we fist prove that fuzzy linear interval graphs, which form a subclass of fuzzy circular interval graphs, are perfect, and we establish a formula for all Ramsey numbers for this graph class and for perfect graphs.

In Section 4, we continue to give positive results that complement the negative results mentioned above. In particular, we are able to determine all Ramsey numbers for bipartite graphs, an important subclass of triangle-free graphs; for co-comparability graphs, a large subclass of AT-free graphs; and for split graphs and cographs, two famous subclasses of P_5-free graphs. In other words, our results narrow the gap between known graph classes for which all Ramsey numbers can be determined by exact formulas, and known graph classes for which it is highly unlikely that such a formula will ever be found. Note that the graph classes that are mentioned so far in this paragraph are all perfect. We complete Section 4 by showing that all Ramsey numbers can be determined for several other subclasses of perfect graphs, and two graph classes close to perfect graphs: cactus graphs and circular-arc graphs. The methods we use in Section 3 and in Section 4 are similar: in graphs that belong to a non-perfect graph class $\mathcal{G}$, we identify subgraphs that are perfect, and we use the formula for the corresponding perfect subclass to obtain a formula for $\mathcal{G}$.

2 Preliminaries

All graphs we consider are undirected, finite and simple. A subset S of vertices of a graph is a *clique* if all the vertices in S are pairwise adjacent, and S is an *independent set* if no two vertices of S are adjacent. For any graph class $\mathcal{G}$ and any two positive integers i and j, we define the Ramsey number $R_{\mathcal{G}}(i,j)$ to be the smallest positive integer such that every graph in $\mathcal{G}$ on at least $R_{\mathcal{G}}(i,j)$ vertices contains a clique of size i or an independent set of size j. When $\mathcal{G}$ is the class of all graphs, we write $R(i,j)$ instead of $R_{\mathcal{G}}(i,j)$. It is well-known that Ramsey numbers for general graphs are symmetric, i.e., that $R(i,j) = R(j,i)$ for all $i,j \geq 1$. More generally, $R_{\mathcal{G}}(i,j) = R_{\mathcal{G}}(j,i)$ for all $i,j \geq 1$ for every class $\mathcal{G}$ that is closed under taking complements, i.e., if for every graph G in $\mathcal{G}$, its complement $\overline{G}$ also belongs to $\mathcal{G}$. If $\mathcal{G}$ is not closed under taking complements, then the Ramsey numbers for $\mathcal{G}$ are typically not symmetric. For any two graph classes $\mathcal{G}$ and $\mathcal{G}'$ such that $\mathcal{G} \subseteq \mathcal{G}'$, we clearly have that $R_{\mathcal{G}}(i,j) \leq R_{\mathcal{G}'}(i,j)$ for all $i,j \geq 1$. In particular, it holds that $R_{\mathcal{G}}(i,j) \leq R(i,j)$ for any graph class $\mathcal{G}$ and all $i,j \geq 1$, which implies that all such numbers $R_{\mathcal{G}}(i,j)$ exist.

The following observation holds for all the graph classes studied in this paper, and for the class of all graphs in particular.

Observation 1 *For any graph class $\mathcal{G}$, $R_{\mathcal{G}}(1,j) = 1$ for all $j \geq 1$. Moreover, if $\mathcal{G}$ contains all edgeless graphs, then $R_{\mathcal{G}}(2,j) = j$ for all $j \geq 1$.*

[1] Recently, Chudnovsky and Seymour [8] proved that every claw-free graph can be composed from graphs belonging to some basic classes. In [9], they identified line graphs and long circular interval graphs as the two principal basic classes of claw-free graphs. Fuzzy circular interval graphs form a superclass of long circular interval graphs.

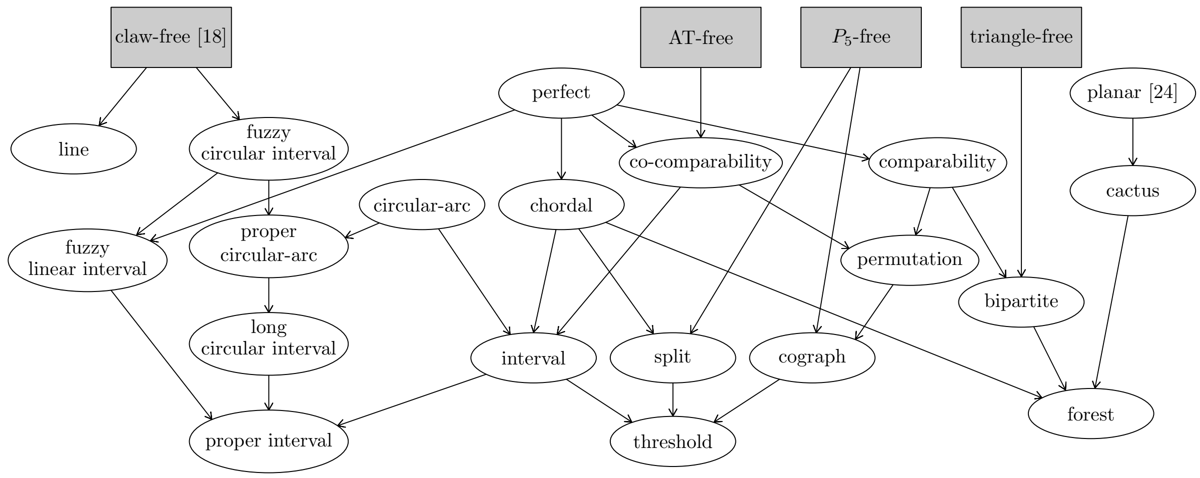


Fig. 1. An overview of the graph classes mentioned in this paper. An arrow from a class $\mathcal{G}_1$ to a class $\mathcal{G}_2$ indicates that $\mathcal{G}_2$ is a proper subclass of $\mathcal{G}_1$. All the depicted inclusion relations were previously known, apart from one: we prove in Lemma 3 that every fuzzy linear interval graph is perfect. For each of the graph classes in an elliptic frame, there exists a formula for determining all Ramsey numbers. Prior to our work, such a formula was only known for planar graphs [25]. For each of the graph classes in a shaded rectangular box, such a formula is unlikely to be found, as there are infinitely many nontrivial Ramsey numbers that are as hard to determine as for general graphs. This was previously known only for claw-free graphs [19].

For some graph classes, we will also make use of the following observation.

Observation 2 *For any graph class $\mathcal{G}$, $R_{\mathcal{G}}(i,1) = 1$ for all $i \geq 1$. Moreover, if $\mathcal{G}$ contains all complete graphs, then $R_{\mathcal{G}}(i,2) = i$ for all $i \geq 1$.*

We refer to the monograph by Diestel [11] for any standard graph terminology not defined below. Let $G = (V, E)$ be a graph, let $v \in V$ and $S \subseteq V$. The complement of G is denoted by $\overline{G}$. We write $G[S]$ to denote the subgraph of G induced by S. For notational convenience, we sometimes write $G - S$ instead of $G[V \setminus S]$ and $G - v$ instead of $G[V \setminus \{v\}]$. The maximum degree of G is denoted by $\Delta(G)$. The clique number $\omega(G)$ of G is the size of a largest clique in G, and the independence number $\alpha(G)$ of G is the size of a largest independent set in G. We write $\nu(G)$ to denote the size of a maximum matching in G, and $\chi(G)$ to denote the chromatic number of G. Given graphs $G_1 = (V_1, E_1), \ldots, G_k = (V_k, E_k)$ such that $V_i \cap V_j = \emptyset$ and $E_i \cap E_j = \emptyset$ for every $i, j \in \{1, \ldots, k\}$ with $i \neq j$, the *disjoint union* of $G_1, \ldots, G_k$ is the graph $(V_1 \cup \ldots \cup V_k, E_1 \cup \ldots \cup E_k)$. The complete graph on ℓ vertices is denoted by K_ℓ. We use P_ℓ and C_ℓ to denote the graphs that are isomorphic to the path and the cycle on ℓ vertices, respectively, i.e., P_ℓ is the graph with vertex set $\{v_1, v_2, v_3, \ldots, v_\ell\}$ and edge set $\{v_1v_2, v_2v_3, \ldots, v_{\ell-1}v_\ell\}$, and C_ℓ is obtained from P_ℓ by adding the edge $v_\ell v_1$.

We now give a brief definition of most of the graph classes studied in this paper. Some graph classes whose definitions require additional terminology will be defined in the next section. For many of the classes mentioned here, several equivalent definitions and characterizations are known; we only mention those that best fit our purposes. Figure 1 shows the inclusion relationships between the classes mentioned in this paper and summarizes our results. More information on these classes, including a wealth of information on applications, can be found in the excellent monographs by Brandstädt et al. [4] and by Golumbic [16].

For every fixed graph H, the class of *H-free* graphs is the class of graphs that do not contain an induced subgraph isomorphic to H. The *claw* is the graph isomorphic to $K_{1,3}$ and the *triangle* is the graph isomorphic to K_3. An *asteroidal triple (AT)* is a set of three pairwise non-adjacent vertices such that between every two of them, there is a path that does not contain a neighbor of the third. A graph is *AT-free* if it does not contain an AT. A graph $G = (V, E)$ is a *circular-arc graph* if there exists a family $\mathcal{I}$ of arcs of a circle $\mathcal{C}$ such that one can associate with each vertex $v \in V$ an arc in $\mathcal{I}$ and such that two vertices of G are adjacent if and only if their corresponding arcs intersect. The pair $(G, \mathcal{I})$ is called a *circular-arc model* of G. A *proper circular-arc graph* is a circular-arc graph G that has an circular-arc model $(G, \mathcal{I})$ in which no arc of $\mathcal{I}$ properly contains another.

A graph is *perfect* if $\omega(G') = \chi(G')$ for every induced subgraph G' of G. The strong perfect graph theorem, proved by Chudnovsky et al. [7] after being conjectured by Berge more than 40 years earlier, states that a graph is perfect if and only if it does not contain a chordless cycle of odd length at least 5 or the complement of such a cycle as an induced subgraph. The graph classes we define next are all perfect. A graph is *chordal* if it does not contain a chordless cycle of length greater than 3 as an induced subgraph. A graph $G = (V, E)$ is an *interval graph* if it is the intersection graph of a family $\mathcal{I}$ of intervals of the real line, i.e., if there exists a family $\mathcal{I}$ of intervals of the real line such that one can associate with each vertex $v \in V$ an interval in $\mathcal{I}$ and such that two vertices of G are adjacent if and only if their corresponding intervals intersect; the pair $(G, \mathcal{I})$ is called an *interval model* of G. If a graph G admits an interval model $(G, \mathcal{I})$ such that no interval of $\mathcal{I}$ properly contains another, then G is a *proper interval graph.* A *comparability* graph is a graph that is transitively orientable, i.e., its edges can be directed such that whenever (a, b) and (b, c) are directed edges, then (a, c) is a directed edge. A graph is a *co-comparability* graph if it is the complement of a comparability graph. A *permutation graph* is the intersection graph of a family of line segments connecting two parallel lines; the class of permutation graphs is exactly the intersection between the classes of comparability and co-comparability graphs. A graph is a *cograph* if and only if it does not contain an induced subgraph isomorphic to P_4. A *split graph* is a graph whose vertex set can be partitioned into a clique and an independent set. A graph G is a *threshold* graph if and only if there is an ordering $v_1, \ldots, v_n$ of its vertices such that $N_G[v_1] \subseteq N_G[v_2] \subseteq \cdots \subseteq N_G[v_n]$; it is well-known that every threshold graph is a split graph. A graph is *bipartite* if its vertex set can be partitioned into two independent sets.

3 Ramsey numbers for subclasses of claw-free graphs and for perfect graphs

Matthews [19] showed that when $\mathcal{G}$ is the class of claw-free graphs, $R_{\mathcal{G}}(i, 3) = R(i, 3)$ for every positive integer i, which implies that there are infinitely many (nontrivial) Ramsey numbers for claw-free graphs that are as hard to determine as for arbitrary graphs. The next theorem implies that this is the case for many other graph classes as well.

Theorem 2. *Let $\mathcal{G}$ be a class of graphs. If $\mathcal{G}$ contains the class of K_i-free graphs as a subclass for some i, then $R_{\mathcal{G}}(i, j) = R(i, j)$ for all $j \geq 1$. Moreover, if $\mathcal{G}$ contains the class of $\overline{K_j}$-free graphs as a subclass for some j, then $R_{\mathcal{G}}(i, j) = R(i, j)$ for all $i \geq 1$.*

Proof. Let i be an integer, and suppose that $\mathcal{G}$ contains the class of K_i-free graphs as a subclass. Clearly, $R_{\mathcal{G}}(i, j) \leq R(i, j)$ for all $j \geq 1$. We now show that $R_{\mathcal{G}}(i, j) \geq R(i, j)$ for all $j \geq 1$. For every integer $j \geq 1$, there exists, by the definition $R(i, j)$, a graph G on $R(i, j) - 1$ vertices that contains neither K_i nor $\overline{K_j}$ as an induced subgraph. Since G is K_i-free, we have that $G \in \mathcal{G}$. This implies that $R_{\mathcal{G}}(i, j) \geq |V(G)| + 1 = R(i, j)$, and hence $R_{\mathcal{G}}(i, j) = R(i, j)$, for all $j \geq 1$. The proof of the second statement in the theorem is identical up to symmetry. $\square$

Note that setting $j = 3$ in Theorem 2 implies the aforementioned result by Matthews on claw-free graphs, and also shows that there are infinitely many nontrivial Ramsey numbers $R_{\mathcal{G}}(i, j)$ that are as hard to determine as $R(i, j)$ when $\mathcal{G}$ is the class of AT-free graphs or the class of P_ℓ-free graphs with $\ell \geq 5$. Setting $i = 3$ shows that the same holds for the class of triangle-free graphs.

In this section, we show that all Ramsey numbers can be determined for some important subclasses of claw-free graphs. In particular, in Section 3.1 we determine all Ramsey numbers for line graphs, arguably the most famous subclass of claw-free graphs. We also obtain such a formula for long circular interval graphs and for fuzzy circular interval graphs in Section 3.3. The results of Section 3.3 rely on the fact that fuzzy linear interval graphs, which form a subclass of fuzzy circular interval graphs, are perfect; this fact is proved in Section 3.2, where we also establish all Ramsey numbers for perfect graphs and for fuzzy linear interval graphs.

3.1 Line graphs

For every graph G, the *line graph* of G, denoted $L(G)$, is the graph with vertex set $E(G)$, where there is an edge between two vertices $e, e' \in E(G)$ if and only if the edges e and e' are incident in G. Graph G is called the *preimage graph* of $L(G)$. A graph is a *line graph* if it is the line graph of some graph. Let $\mathcal{L}$ denote the class of all line graphs. In this subsection, we determine all Ramsey numbers for line graphs.

Recall that the value of $R_{\mathcal{L}}(i, j)$ for $i \in \{1, 2\}$ and every $j \geq 1$ follows from Observation 1. The case $i = 3$ is the first nontrivial case for the class of line graphs. In his study of the Ramsey numbers $R_{\mathcal{C}}(3, j)$ for all $j \geq 1$, where $\mathcal{C}$ is the class of claw-free graphs, Matthews [19] used arguments that yield the following theorem, which also holds for claw-free graphs. We present its proof for the sake of completeness.

Theorem 3. *For every integer $j \geq 1$, $R_{\mathcal{L}}(3, j) = \lfloor (5j - 3)/2 \rfloor$.*

Proof. Suppose G is a line graph that contains neither a clique of size 3 nor an independent set of size j. Since G is both triangle-free and claw-free, we must have $\Delta(G) \leq 2$. Hence G is the disjoint union of a collection of paths and cycles. Let $\mathcal{S}$ be the set of connected components of G, and let $\mathcal{S}_0 \subseteq \mathcal{S}$ be the set of connected components of G that are odd cycles. Then $\alpha(S) = (|V(S)| - 1)/2$ for every $S \in \mathcal{S}_0$ and $\alpha(S) = \lceil |V(S)|/2 \rceil$ for every $S \in \mathcal{S} \setminus \mathcal{S}_0$, which implies that

$$\alpha(G) = \sum_{S \in \mathcal{S}_0} (|V(S)| - 1)/2 + \sum_{S \in \mathcal{S} \setminus \mathcal{S}_0} \lceil |V(S)|/2 \rceil \geq \sum_{S \in \mathcal{S}} |V(S)|/2 - \sum_{S \in \mathcal{S}_0} 1/2\,,$$

and hence $\alpha(G) \geq (|V(G)| - |\mathcal{S}_0|)/2$.

Since G is triangle-free, every connected component in $\mathcal{S}_0$ contains at least 5 vertices. Hence $|\mathcal{S}_0| \leq |V(G)|/5$, which together with the inequality $\alpha(G) \geq (|V(G)| - |\mathcal{S}_0|)/2$ implies that $\alpha(G) \geq 2|V(G)|/5$. On the other hand, we have that $\alpha(G) \leq j-1$, since we assumed that G has no independent set of size j. Combining these two bounds on $\alpha(G)$ yields the inequality $2|V(G)|/5 \leq j - 1$, or equivalently $|V(G)| \leq 5(j-1)/2 \leq \lfloor (5j-3)/2 \rfloor - 1$, where the last inequality holds due to the fact that $|V(G)|$ is an integer. This shows that any line graph that has neither a clique of size 3 nor an independent set of size j has at most $\lfloor (5j-3)/2 \rfloor - 1$ vertices, which implies that $R_{\mathcal{L}}(3, j) \leq \lfloor (5j-3)/2 \rfloor$.

It remains to show that $R_{\mathcal{L}}(3, j) \geq \lfloor (5j-3)/2 \rfloor$ for every $j \geq 1$. In order to do this, it suffices to show that for every $j \geq 1$, there exists a line graph G_j on $\lfloor (5j-3)/2 \rfloor - 1$ vertices satisfying $\omega(G_j) < 3$ and $\alpha(G) < j$. We construct such a graph G_j for every $j \geq 1$ as follows. If $j = 2k$, then G_j is the disjoint union of $k - 1$ copies of C_5 and one copy of K_2, while we define G_j to be the disjoint union of k copies of C_5 if $j = 2k+1$. Note that G_j is a line graph for every $j \geq 1$. It is easy to verify that for every $j \geq 1$, it holds that $\omega(G_j) = 2 < 3$ and $\alpha(G_j) = j-1 < j$. If $j = 2k$, then $|V(G_j)| = 5(k-1)+2 = (10k-6)/2 = (5j-4)/2 - 1 = \lfloor (5j-3)/2 \rfloor - 1$, where the last equality holds since j is even. In a similar way, it is easy to check that $|V(G)| = \lfloor (5j-3)/2 \rfloor - 1$ if $j = 2k+1$. We conclude that $R_{\mathcal{L}}(3,j) \geq \lfloor (5j-3)/2 \rfloor$ and consequently $R_{\mathcal{L}}(3,j) = \lfloor (5j-3)/2 \rfloor$ for every $j \geq 1$. □

In order to determine the Ramsey numbers $R_{\mathcal{L}}(i,j)$ for $i \geq 4$, we make use of the following two results, the first of which is an easy observation.

Lemma 1. *Let H be a graph, let $G = L(H)$ be the line graph of H, and let $i \geq 4$ and $j \geq 1$ be two integers. Then H has a vertex of degree at least i if and only if G has a clique of size i. Moreover, H has a matching of size j if and only if G has an independent set of size j.*

Proof. It is clear from the definition of line graphs that if H contains a vertex of degree at least i or a matching of size j, then G contains a clique of size i or an independent set of size j, respectively. For the reverse direction, suppose G has a clique X of size i. Let $e_1, \ldots, e_i$ be the edges in H corresponding to the vertices in X. Since X is a clique and $i \geq 4$, all edges in $\{e_1, \ldots, e_i\}$ must share a common vertex v. Hence H contains a vertex of degree at least i. If G contains a independent set of size j, then the corresponding edges in H form a matching of size j. □

Theorem 4 ([1,2,10]). *Let $i \geq 4$ and $j \geq 1$ be two integers, and let H be an arbitrary graph such that $\Delta(H) < i$ and $\nu(H) < j$. Then*

$$|E(H)| \leq \begin{cases} i(j-1) - (t+r) + 1 & \text{if } i = 2k \\ i(j-1) - r + 1 & \text{if } i = 2k+1, \end{cases}$$

where $j = tk + r$, $t \geq 0$ and $1 \leq r \leq k$, and this bound is tight.

In a preliminary version [2] of our current paper, we presented Theorem 4 with a full proof. Very recently, we have been made aware of the fact that Theorem 4 had already been proved by Balachandran and Khare [1], and that the upper bound in Theorem 4 also appeared in a paper by Chvátal and Hanson (Lemma 2 in [10]).

Lemma 1 and Theorem 4 yield the following formula for all the Ramsey numbers for line graphs that were not covered by Observation 1 and Theorem 3.

Theorem 5. *For every pair of integers $i \geq 4$ and $j \geq 1$, it holds that*

$$R_{\mathcal{L}}(i,j) = \begin{cases} i(j-1) - (t+r) + 2 & \text{if } i = 2k \\ i(j-1) - r + 2 & \text{if } i = 2k+1, \end{cases}$$

where $j = tk + r$, $t \geq 0$ and $1 \leq r \leq k$.

Proof. For notational convenience, we define a function ρ as follows: for every pair of integer $i \geq 4$ and $j \geq 1$, let

$$\rho(i,j) = \begin{cases} i(j-1) - (t+r) + 1 & \text{if } i = 2k \\ i(j-1) - r + 1 & \text{if } i = 2k+1, \end{cases}$$

where $j = tk + r$, $t \geq 0$ and $1 \leq r \leq k$. Then an equivalent way of stating Theorem 5 is to say that $R_{\mathcal{L}}(i,j) = \rho(i,j) + 1$ for every $i \geq 4$ and $j \geq 1$.

Let G be a line graph that contains neither a clique of size i nor an independent set of size j, and let H be the preimage graph of G. Then $\Delta(H) < i$ and $\nu(H) < j$ as a result of Lemma 1.

Hence, by Theorem 4, we have that $|E(H)| \leq \rho(i,j)$. Since G is the line graph of H, we have $|V(G)| = |E(H)| \leq \rho(i,j)$, implying the upper bound $R_{\mathcal{L}}(i,j) \leq \rho(i,j)+1$. To prove the matching lower bound, note that Theorem 4 ensures, for every $i \geq 4$ and $j \geq 1$, the existence of a graph H with exactly $\rho(i,j)$ edges that satisfies $\Delta(H) < i$ and $\nu(H) < j$. The line graph $G = L(H)$ of such a graph has exactly $\rho(i,j)$ vertices, and contains neither a clique of size i nor an independent set of size j due to Lemma 1. This implies that $R_{\mathcal{L}}(i,j) \geq \rho(i,j)+1$, and consequently $R_{\mathcal{L}}(i,j) = \rho(i,j)+1$, for every $i \geq 4$ an $j \geq 1$. □

3.2 Perfect graphs and fuzzy linear interval graphs

Although perfect graphs are not related to claw-free graphs, we start this subsection by determining all Ramsey numbers for perfect graphs. We then show that fuzzy linear interval graphs, which are known to be claw-free [8,18], are also perfect. Combining these two results yields a formula for all Ramsey numbers for fuzzy linear interval graphs.

A graph class $\mathcal{G}$ is *χ-bounded* if there exists a non-decreasing function $f : \mathbb{N} \to \mathbb{N}$ such that for every $G \in \mathcal{G}$, we have $\chi(G') \leq f(\omega(G'))$ for every induced subgraph G' of G. Such a function f is called a *χ-bounding function* for $\mathcal{G}$, and we say that $\mathcal{G}$ is *χ-bounded* if there exists a χ-bounding function for $\mathcal{G}$. Both Walker [26] and Steinberg and Tovey [25] observed the close relationship between the chromatic number and Ramsey number of a graph when they studied Ramsey numbers for planar graphs. Their key observation can be applied to any χ-bounded graph class as follows.

Lemma 2. *Let $\mathcal{G}$ be a χ-bounded graph class with χ-bounding function f. Then $R_{\mathcal{G}}(i,j) \leq f(i-1)(j-1)+1$ for all $i,j \geq 1$.*

Proof. Let G be a graph in $\mathcal{G}$ with at least $f(i-1)(j-1)+1$ vertices. Suppose that G contains no K_i. Since G has no K_i, we have $\omega(G) \leq i-1$. By the definition of a χ-bounding function, $\chi(G) \leq f(\omega(G)) \leq f(i-1)$. Let ϕ be any proper vertex coloring of G. Since ϕ uses at most $f(i-1)$ colors and G has at least $f(i-1)(j-1)+1$ vertices, there must be a color class that contains at least j vertices. Consequently, G contains an independent set of size j. □

Theorem 6. *Let $\mathcal{G}$ be the class of perfect graphs or a subclass of it containing all disjoint unions of complete graphs. Then $R_{\mathcal{G}}(i,j) = (i-1)(j-1)+1$ for all $i,j \geq 1$.*

Proof. Observe that the identity function is a χ-bounding function for the class of perfect graphs and each of its subclasses. Consequently, $R_{\mathcal{G}}(i,j) \leq (i-1)(j-1)+1$ for all $i,j \geq 1$ due to Lemma 2. The matching lower bound follows from the observation that the disjoint union of $j-1$ copies of K_{i-1} is a graph on $(i-1)(j-1)$ vertices that belongs to $\mathcal{G}$ and that has neither a clique of size i nor an independent set of size j. □

Theorem 6 immediately implies a formula for all Ramsey numbers for another subclass of claw-free graphs, namely proper interval graphs, which are perfect, claw-free, and contain disjoint unions of complete graphs. We will now show that fuzzy linear interval graphs, which form a superclass of proper interval graphs and a subclass of claw-free graphs, are perfect, and the same formula holds for all their Ramsey numbers as well.

Let us define fuzzy linear interval graphs, along with the remaining graph classes of Section 3. Given a circle $\mathcal{C}$, a *closed interval* of $\mathcal{C}$ is a proper subset of $\mathcal{C}$ homeomorphic to the closed unit interval $[0,1]$; in particular, every closed interval of $\mathcal{C}$ has two distinct endpoints. Linear interval graphs and circular interval graphs, identified by Chudnovsky and Seymour [8] as two basic classes of claw-free graphs, can be defined as follows.

Definition 1. *A graph $G = (V, E)$ is a* circular interval graph *if the following conditions hold:*

- *there is an injective mapping φ from V to a circle $\mathcal{C}$;*
- *there is a set $\mathcal{I}$ of closed intervals of $\mathcal{C}$, none including another, such that two vertices $u, v \in V$ are adjacent if and only if $\varphi(u)$ and $\varphi(v)$ belong to a common interval of $\mathcal{I}$.*

A graph G is a linear interval graph *if it satisfies the above conditions when we substitute "circle" by "line".*

We call the triple $(V, \varphi, \mathcal{I})$ in Definition 1 a *circular interval model* of G. A graph $G = (V, E)$ is a *long circular interval graph* if it has a circular interval model $(V, \varphi, \mathcal{I})$ such that no three intervals in $\mathcal{I}$ cover the entire circle $\mathcal{C}$.

It is known that linear interval graphs and circular interval graphs are equivalent to proper interval graphs and proper circular-arc graphs, respectively [8] (see also [18]). It is immediate from the above definitions that circular interval graphs form a superclass of both long circular interval graphs and linear interval graphs. We now define a superclass of circular interval graphs that was introduced by Chudnovsky and Seymour [8] as yet another important class of claw-free graphs.

Definition 2. *A graph $G = (V, E)$ is a* fuzzy circular interval graph *if the following conditions hold:*

- *there is a (not necessarily injective) mapping φ from V to a circle $\mathcal{C}$;*
- *there is a set $\mathcal{I}$ of closed intervals of $\mathcal{C}$, none including another, such that no point of $\mathcal{C}$ is an endpoint of more than one interval in $\mathcal{I}$, and*
  - *if two vertices $u, v \in V$ are adjacent, then $\varphi(u)$ and $\varphi(v)$ belong to a common interval of $\mathcal{I}$;*
  - *if two vertices $u, v \in V$ are not adjacent, then either there is no interval in $\mathcal{I}$ that contains both $\varphi(u)$ and $\varphi(v)$, or there is exactly one interval in $\mathcal{I}$ whose endpoints are $\varphi(u)$ and $\varphi(v)$.*

A graph G is a fuzzy linear interval graph *if it satisfies the above conditions when we substitute "circle" by "line".*

We call the triple $(V, \varphi, \mathcal{I})$ in Definition 2 a *fuzzy circular interval model* (or simply *model*) of G, and call $(V, \varphi, \mathcal{I})$ a *fuzzy linear interval model* if $\mathcal{I}$ is a set of intervals of a line. Clearly, the class of fuzzy linear interval graphs is a subclass of fuzzy circular interval graphs. This also holds for the class of circular interval graphs (and hence for the class of proper circular-arc graphs), as they are exactly those fuzzy circular interval graphs $G = (V, E)$ that have a model $(V, \varphi, \mathcal{I})$ such that φ is injective [8]. Similarly, a graph $G = (V, E)$ is a linear interval graph if and only if it is a fuzzy linear interval graph that has a model $(V, \varphi, \mathcal{C})$ such that φ is injective.

Let us remark that *circular-arc graphs* form neither a subclass nor a superclass of fuzzy circular interval graphs: the claw is an example of a circular-arc graph that is not a fuzzy circular interval graph, whereas the complement of C_6 is known not to be a circular-arc graph (see, e.g., [3]), but is a fuzzy circular interval graph (see Figure 2 for a fuzzy circular interval model of this graph).

Very recently, Chudnovsky and Plumettaz [6] proved that every linear interval trigraph is perfect, where linear interval trigraph is a notion closely related to linear interval graphs. Their argument can be adapted to prove the following lemma.

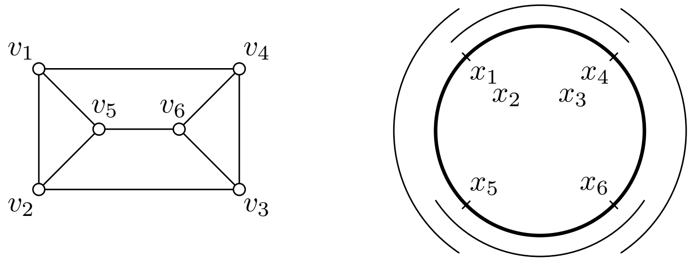


Fig. 2. The graph $\overline{C_6}$ and a fuzzy circular interval model of this graph, where $x_i = \varphi(v_i)$ for every $i \in \{1, \ldots, 6\}$.

Lemma 3. *Every fuzzy linear interval graph is perfect.*

Proof. We prove the lemma by induction on the number of vertices. Note that the graph on one vertex is perfect. Suppose that every fuzzy linear interval graph on at most $n-1$ vertices is perfect, and let $G = (V, E)$ be a fuzzy linear interval graph on n vertices with model $(V, \varphi, \mathcal{I})$, where $\mathcal{I}$ is a set of intervals of a line $\mathcal{L}$. For any two vertices $u, v \in V$, we write $\varphi(u) \leq \varphi(v)$ if $\varphi(u) = \varphi(v)$ or if $\varphi(u)$ lies to the left of $\varphi(v)$ on the line $\mathcal{L}$. Let $v_n \in V$ be such that $\varphi(v) \leq \varphi(v_n)$ for all $v \in V \setminus \{v_n\}$, i.e., no vertex of G is mapped to the right of v_n. Let $w \in N_G(v_n)$ be such that $\varphi(w) \leq \varphi(v)$ for all $v \in N_G(v_n)$, i.e., no neighbor of v_n is mapped to the left of w. Since w is adjacent to v_n, there is an interval in $I \in \mathcal{I}$ that contains both w and v_n. By the definition of w, every vertex of $N_G(v_n)$ belongs to I, which implies that $N_G(v_n)$ is a clique in G.

Due to the strong perfect graph theorem [7], in order to complete the proof of the lemma, it suffices to prove that G contains neither a chordless odd cycle nor the complement of such a cycle. For contradiction, suppose there exists a set $X \subseteq V$ such that X induces a chordless odd cycle or the complement of such a cycle. Note that for every $x \in X$, the set $N_G(x) \cap X$ is not a clique in G. Since we proved that $N_G(v_n)$ is a clique, we deduce that $v_n \notin X$. Consequently, X is a subset of the vertices of the graph $G - v_n$, which means that $G - v_n$ contains a chordless odd cycle or the complement of such a cycle. However, by the induction hypothesis and the fact that $G - v_n$ is a fuzzy linear interval graph, the graph $G - v_n$ is perfect. This yields the desired contradiction. $\square$

It is easy to see that any disjoint union of complete graphs is a fuzzy linear interval graph. Hence Theorem 6 readily implies the next result.

Theorem 7. *Let $\mathcal{G}$ be the class of fuzzy linear interval graphs. Then $R_{\mathcal{G}}(i, j) = (i-1)(j-1)+1$ for all $i, j \geq 1$.*

3.3 Long circular interval graphs and fuzzy circular interval graphs

Before we proceed with the results of this section, we need some additional terminology and settle notation. Let $G = (V, E)$ be a fuzzy circular interval graph with model $(V, \varphi, \mathcal{I})$. For every point p on the circle $\mathcal{C}$, we define $\varphi^{-1}(p) = \{v \in V \mid \varphi(v) = p\}$. Moreover, for every interval I of $\mathcal{C}$ (possibly $I \notin \mathcal{I}$), we define $\varphi^{-1}(I) = \bigcup_{p \in I} \varphi^{-1}(p)$. Throughout this section, we assume that for every $I \in \mathcal{I}$, the set $\varphi^{-1}(I)$ is non-empty, as otherwise we can simply delete I from $\mathcal{I}$. For any two points p_1 and p_2 on the circle $\mathcal{C}$, we write $[p_1, p_2]$ to denote the closed interval of $\mathcal{C}$ that we span when we traverse $\mathcal{C}$ clockwise from p_1 to p_2. We write $\langle p_1, p_2\rangle = [p_1, p_2] \setminus \{p_1, p_2\}$ and $\langle p_1, p_2] = [p_1, p_2] \setminus \{p_1\}$ and $[p_1, p_2\rangle = [p_1, p_2] \setminus \{p_2\}$.

We would like to point out that every fuzzy circular interval graph $G = (V, E)$ has a model $(V, \varphi, \mathcal{I})$ such that for every point p on $\mathcal{C}$ for which the set $X_p = \varphi^{-1}(p)$ is non-empty, the vertices of X_p form a clique in G. To see this, note that if there is an edge between any two vertices in X_p, then there is an interval of $\mathcal{I}$ that covers p, and hence X_p is a clique. If the vertices in X_p form an independent set in G, then there is an interval $[p^-, p^+]$ of $\mathcal{C}$ that contains p and does not intersect with any interval in $\mathcal{I}$. Hence we can simply define a new model $(V, \varphi', \mathcal{I})$ such that φ' maps the vertices of X_p to distinct points in the interval $[p^-, p^+]$.

Let I be a closed interval of a circle $\mathcal{C}$. We write I^ℓ and I^r to denote the two points on $\mathcal{C}$ such that $I = [I^\ell, I^r]$. By slight abuse of terminology, we refer to I^ℓ and I^r as the *left endpoint* and the *right endpoint* of the interval I, respectively. A point p is an *interior point* of I if p is not an endpoint of I, i.e., if $p \in I \setminus \{I^\ell, I^r\}$. For any two points p_1 and p_2 that belong to I, we write $p_1 < p_2$ if p_2 does not belong to the subinterval $[I^\ell, p_1]$, i.e., if p_1 is closer to the left endpoint of I than p_2 is.

In order to determine all Ramsey numbers for long circular interval graphs and fuzzy circular interval graphs, we will use the following two lemmas.

Lemma 4. *Every fuzzy circular interval graph G contains a clique X of size at most $\omega(G)-1$ such that $G - X$ is a fuzzy linear interval graph.*

Proof. Let $G = (V, E)$ be a fuzzy circular interval graph with model $(V, \varphi, \mathcal{I})$, where $\mathcal{I}$ is a set of intervals of a circle $\mathcal{C}$. The lemma trivially holds if G is a fuzzy linear interval graph. Suppose G is not a fuzzy linear interval graph. Let P be the set of $2|\mathcal{I}|$ points on the circle $\mathcal{C}$ that are endpoints of intervals in $\mathcal{I}$. We partition P into two sets by defining $P_\ell = \{p \in P \mid p = I^\ell \text{ for some } I \in \mathcal{I}\}$ and $P_r = \{p \in P \mid p = I^r \text{ for some } I \in \mathcal{I}\}$. We also define $Q = \{q \in \mathcal{C} \mid \varphi^{-1}(q) \neq \emptyset\}$, i.e., Q consists of the points q on $\mathcal{C}$ such that φ maps at least one vertex of V to q.

Now let $a \in Q$ be an arbitrary point on the circle. We claim that a is an interior point of at least one interval of $\mathcal{I}$. Since G is not a fuzzy linear interval graph, a is covered by an interval $J \in \mathcal{I}$. Suppose a is not an interior point of J, and without loss of generality assume that $a = J^\ell$. Let $p \in P \setminus \{a\}$ be the first point of P that we encounter when we traverse $\mathcal{C}$ counterclockwise from a, i.e., p is the unique point in $P \setminus \{a\}$ such that the interval $\langle p, a\rangle$ contains no vertex of P. Let $q \in \langle p, a\rangle$. Since G is not a fuzzy linear interval graph, q is covered by an interval $J' \in \mathcal{I}$. Since $q \notin P$ and $a = J^\ell$, and no two intervals in $\mathcal{I}$ have a common endpoint by definition, it holds that $J' \neq J$ and hence a is an interior point of J'.

Let $\mathcal{I}_a \subseteq \mathcal{I}$ consist of all the intervals in $\mathcal{I}$ that contain a as an interior point, where $\mathcal{I}_a = \{A_1, A_2, \ldots, A_s\}$ such that $a < A_1^r < A_2^r < \cdots < A_s^r$. As we argued above, the set $\mathcal{I}_a$ is non-empty. If a happens to be the endpoint of some interval $J \in \mathcal{I} \setminus \mathcal{I}_a$, then we assume, without loss of generality, that $a = J^\ell$. We define $X = \varphi^{-1}([A_1^\ell, a\rangle)$, i.e., X consists of those vertices of G that are mapped by φ to some point on $\mathcal{C}$ in the interval $[A_1^\ell, a\rangle$. Since the interval $[A_1^\ell, a\rangle$ is contained in the interval A_1, this set X is a clique. Moreover, since $[A_1^\ell, a]$ is also a subinterval of A_1, the set $X \cup \varphi^{-1}(a)$ is also a clique in G. Since $a \in Q$, the set $\varphi^{-1}(a)$ is non-empty, so X has size at most $\omega(G) - 1$.

It remains to prove that the graph $G - X$ is a fuzzy linear interval graph. We do this by constructing a fuzzy linear interval model $(V \setminus X, \varphi', \mathcal{I}')$ of $G - X$ from the model $(V, \varphi, \mathcal{I})$ of G as follows (see Figure 3 for a helpful illustration). First, we define φ' to be the restriction of φ to the vertices of $V \setminus X$, i.e., φ' is the mapping from $V \setminus X$ to $\mathcal{C}$ such that $\varphi'(v) = \varphi(v)$ for all $v \in V \setminus X$. Clearly, $(V \setminus X, \varphi', \mathcal{I})$ is a fuzzy circular interval model of $G - X$. Let $x \in P_r \cup Q$ be such that the interval $\langle x, a\rangle$ does not contain any element of $P_r \cup Q$. Note that it is possible that the interval $\langle x, a\rangle$ contains an element of P_ℓ; any such element is the left endpoint of some interval in $\mathcal{I}_a$ (for example, the left endpoint of interval A_s in Figure 3

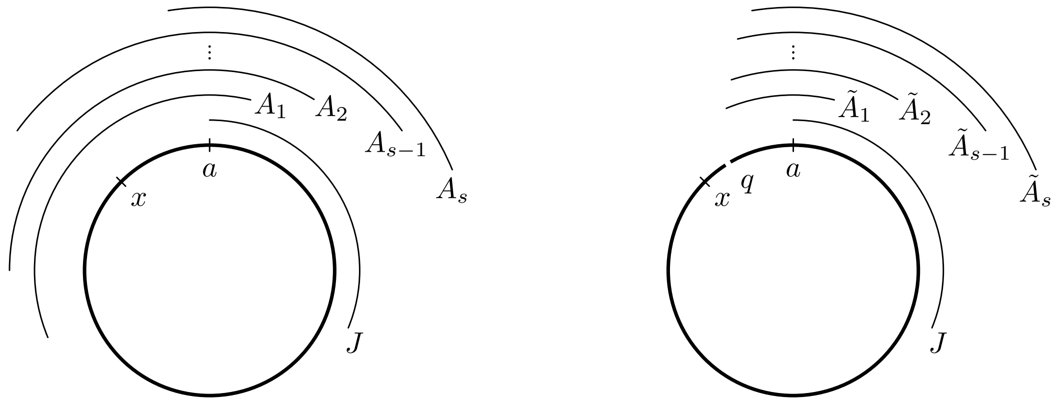


Fig. 3. The intervals $\tilde{A}_1, \ldots, \tilde{A}_s$ are obtained from the intervals $A_1, \ldots, A_s$ by moving the left endpoints of $A_1, \ldots, A_s$ to within the interval $\langle x, a\rangle$. The resulting fuzzy circular interval model can be modified into a fuzzy linear interval model by cutting the circle at any point q in the interval $\langle x, \tilde{A}_1^\ell\rangle$.

lies in the interval $\langle x, a\rangle$). Informally speaking, we will now "shrink" the intervals in $\mathcal{I}_a$ by moving their left endpoints in such a way that all these left endpoints end up in the interval $\langle x, a\rangle$ and the obtained model is still a model of $G - X$, i.e., the new intervals force the same adjacencies and non-adjacencies in the corresponding graph.

Formally, we define, for every $p \in \{1, \ldots, s\}$, a new closed interval $\tilde{A}_p$ of $\mathcal{C}$ such that $\tilde{A}_p^r = A_p^r$ and $\tilde{A}_p^\ell$ is chosen arbitrarily in such a way that $x < \tilde{A}_1^\ell < \tilde{A}_2^\ell < \cdots < \tilde{A}_s^\ell < a$; see Figure 3. Let $\tilde{\mathcal{I}}_a = \{\tilde{A}_1, \ldots, \tilde{A}_s\}$, and let $\mathcal{I}' = (\mathcal{I} \setminus \mathcal{I}_a) \cup \tilde{\mathcal{I}}_a$. Let us first show that $(V \setminus X, \varphi', \mathcal{I}')$ is a fuzzy circular interval model of $G - X$. First note that we chose the left endpoints of the intervals in $\tilde{\mathcal{I}}_a$ in such a way that no interval of $\tilde{\mathcal{I}}_a$ contains another. Moreover, since the interval $\langle x, a\rangle$ contains no vertex of P_r, no point of $\mathcal{C}$ is an endpoint of more than one interval in $\mathcal{I}'$. From the definition of X it follows that, for every $p \in \{1, \ldots, s\}$, φ does not map any vertex of $G - X$ to a point in the interval $[A_p^\ell, a\rangle$. In other words, for every vertex v of $G - X$, interval A_p contains the point $\varphi'(v)$ if and only if interval $\tilde{A}_p$ does, for every $p \in \{1, \ldots, s\}$. This guarantees that the triple $(V \setminus X, \varphi', \mathcal{I}')$ indeed is a fuzzy circular interval model of $G - X$. To see why $(V \setminus X, \varphi', \mathcal{I}')$ is a fuzzy *linear* interval model of $G - X$, it suffices to observe that we can cut the circle $\mathcal{C}$ at any point q in the interval $\langle x, \tilde{A}_1^\ell\rangle$, as any such point q is not covered by any interval in $\mathcal{I}'$ (again, see Figure 3). This completes the proof of Lemma 4. □

As the next observation will be used also in the next section, we state it as a separate lemma.

Lemma 5. *Let G be a graph such that $\omega(G) < i$ and $\alpha(G) < j$ for two integers $i, j \geq 3$. If G contains a clique X of size at most $\omega(G) - 1$ such that $G - X$ is a perfect graph, then G has at most $(i-1)j - 1$ vertices.*

Proof. Suppose G contains a clique X such that $|X| \leq \omega(G) - 1 \leq i - 2$ and $G - X$ is perfect. Since $G - X$ is an induced subgraph of G, it contains neither a clique of size i nor an independent set of size j. Hence, due to Theorem 6, we have that $|V(G-X)| \leq (i-1)(j-1)$. Then $|V| = |V(G - X)| + |X| \leq (i-1)(j-1) + (i-2) = (i-1)j - 1$. □

We are now ready to determine all Ramsey numbers for long circular interval graphs and fuzzy circular interval graphs. Since the class of long circular interval graphs (and hence its superclass of fuzzy circular interval graphs) contains all edgeless graphs and all complete graphs, Observations 1 and 2 yield the Ramsey numbers for both classes for all $i, j \in \{1, 2\}$. All other Ramsey numbers for these two classes are given by the following formula.

Theorem 8. *Let $\mathcal{G}$ be the class of long circular interval graphs or the class of fuzzy circular interval graphs. Then $R_{\mathcal{G}}(i, j) = (i-1)j$ for all $i, j \geq 3$.*

Proof. Let G be a fuzzy circular interval graph, and let i and j be two integers such that $i, j \geq 3$. Suppose G contains neither a clique of size i nor an independent set of size j. By Lemma 4, G contains a clique X of size at most $i-2$ such that $G - X$ is a fuzzy linear interval graph. Since the graph $G - X$ is perfect due to Lemma 3, we know that G has at most $(i-1)j - 1$ vertices as a result of Lemma 5. Hence $R_{\mathcal{G}}(i, j) \leq (i-1)j$ for all $i, j \geq 3$ if $\mathcal{G}$ is the class of fuzzy circular interval graphs, and the same holds if $\mathcal{G}$ is the class of long circular interval graphs, as they form a subclass of fuzzy circular interval graphs.

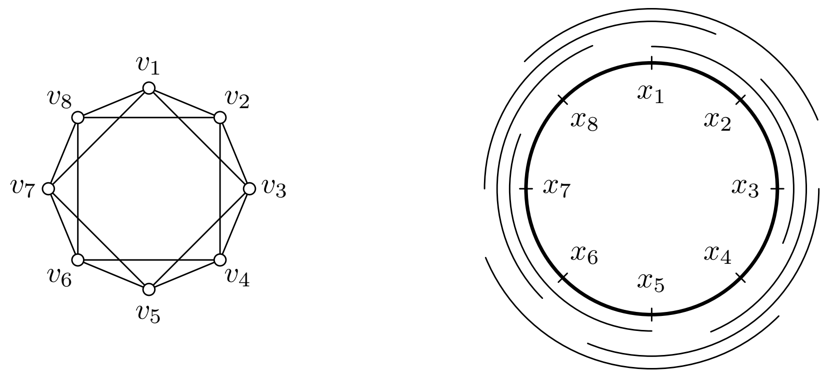


Fig. 4. The graph $G^*_{i,j}$ when $i = 4$ and $j = 3$, together with a circular interval model in which no three intervals cover the circle.

It remains to prove that $R_{\mathcal{G}}(i, j) \geq (i-1)j$ for all $i, j \geq 3$. Note that it suffices to construct a long circular interval graph on $n = (i-1)j - 1$ vertices that has no clique of size i and no independent set of size j. For every $i, j \geq 3$, let $G^*_{i,j}$ be the $(i-2)$th power of $C_{(i-1)j-1}$, i.e., let $G^*_{i,j} = (V, E)$ be the graph obtained from a cycle C on $n = (i-1)j - 1$ vertices by making any $i-1$ consecutive vertices of cycle into a clique. For any subset S of vertices in $G^*_{i,j}$, we say that the vertices of S are consecutive if they appear consecutively on the cycle C. To show that $G^*_{i,j}$ is a long circular interval graph, we construct a long circular interval model $(V, \varphi, \mathcal{I})$ of $G^*_{i,j}$ as follows (see Figure 4 for an illustration of the case where $i = 4$ and $j = 3$). Let $V = \{v_1, v_2, \ldots, v_n\}$ and let $\varphi : V \to \mathcal{C}$ be a mapping that injectively maps the vertices of V to the circle $\mathcal{C}$ in such a way that $\varphi(v_1), \ldots,$ $varphi(v_n)$ appear consecutively on the circle in clockwise order. Let $x_i = \varphi(v_i)$ for every $i \in \{1, \ldots, n\}$, and let $X = \{x_1, \ldots, x_n\}$. For every $p \in \{1, \ldots, n\}$, we define an interval I_p such that $I^\ell_p = x_p$ and I^r_p is chosen arbitrarily such that $x_{p+i-1} < I^r_p < x_{p+i}$, where the indices are taken modulo n (See Figure 4). Let $\mathcal{I} = \{I_1, \ldots, I_n\}$.

Since every interval in $\mathcal{I}$ contains the image of exactly $i-1$ consecutive vertices of $G^*_{i,j}$, forcing them to be in a clique, the triple $(V, \varphi, \mathcal{I})$ clearly is a circular interval model of $G^*_{i,j}$. To prove that $G^*_{i,j}$ is a *long* circular interval graph, it suffices to argue that no three intervals of $\mathcal{I}$ cover the entire circle. By construction, any two intervals $I_p, I_q \in \mathcal{I}$ overlap if and only if there exists a point $x \in X$ that is contained in both I_p and I_q. As a result, any three intervals of $\mathcal{I}$ cover at most $3(i-1)-2$ points of X. Recall that $j \geq 3$, so $|X| \geq 3(i-1)-1$. This implies that for any three intervals in $\mathcal{I}$, at least one point of X is not covered by these three intervals.

It is clear that $G^*_{i,j}$ contains no clique of size i. To show that $\alpha(G^*_{i,j}) < j$, suppose, for contradiction, that $G^*_{i,j}$ contains an independent set S of size j. Since every $i-1$ consecutive vertices in $G^*_{i,j}$ form a clique, we have at least $i-2$ consecutive vertices of $V(G^*_{i,j}) \setminus S$ between any two vertices of S. Since $|S| \geq j$, this implies that $G^*_{i,j}$ contains at least $(i-2)j + |S| \geq (i-1)j$ vertices. This contradiction to the fact that $|V(G^*_{i,j})| = (i-1)j - 1$ completes the proof. □

4 Ramsey numbers for subclasses of perfect graphs and other related classes

In the previous section, we proved fuzzy linear interval graphs to be perfect and we used this result to determine all Ramsey numbers for them. Furthermore, we used this to determine the Ramsey numbers for fuzzy circular interval graphs, by identifying subgraphs that are fuzzy linear interval. In this section, we will see that similar methods can be applied to other other graph classes in which we can identify perfect subgraphs: circular-arc graphs and proper circular-arc graphs. These results are given in Section 4.2. They rely on Ramsey numbers for some subclasses of perfect graphs which we determine first in Section 4.1. We conclude with a formula for all Ramsey numbers for cactus graphs in Section 4.3.

4.1 Subclasses of perfect graphs

Recall that Ramsey numbers for AT-free graphs, for triangle-free graphs, and for P_5-free graphs are as hard to determine as for arbitrary graphs, by Theorem 2. In this section we will see that for several subclasses of these graph classes, we can determine all Ramsey numbers. In particular this is true for split graphs and cographs, which are subclasses of P_5-free graphs; for co-comparability graphs and interval graphs, which are subclasses of AT-free graphs; and for bipartite graphs, which is a subclass of triangle-free graphs.

The following corollary of Theorem 6 follows by observing that all mentioned graph classes contain all disjoint unions of complete graphs.

Corollary 1. *Let $\mathcal{G}$ be the class of chordal graphs, interval graphs, proper interval graphs, comparability graphs, co-comparability graphs, permutation graphs, or cographs. Then $R_{\mathcal{G}}(i,j) = (i-1)(j-1)+1$ for all $i, j \geq 1$,*

Next we consider perfect graph classes that do not contain all disjoint unions of complete graphs. Recall that, for $i \in \{1,2\}$ and every $j \geq 1$, the Ramsey numbers $R_{\mathcal{G}}(i,j)$ for any graph class $\mathcal{G}$ considered in this paper immediately follow from Observation 1. Furthermore, Observation 2 yields all Ramsey numbers $R_{\mathcal{G}}(i,j)$ with $j \in \{1,2\}$ and $i \geq 1$ when $\mathcal{G}$ is the class of split graphs or threshold graphs, since both classes contain all complete graphs. The following two theorems establish all other Ramsey numbers for these two graphs classes.

Theorem 9. *Let $\mathcal{G}$ be the class of split graphs. Then $R_{\mathcal{G}}(i,j) = i+j-1$ for all $i, j \geq 3$.*

Proof. Let G be a split graph on at least $i+j-1$ vertices whose vertices are partitioned into a clique C and an independent set I. Since $|V(G)| \geq i+j-1$, it is not possible that $|C| < i$ and $|I| < j$, which implies that G contains a clique of size i or an independent set of size j. Hence $R_{\mathcal{G}}(i,j) \leq i+j-1$. For the lower bound, consider a split graph G whose vertices are partitioned into a clique C of size $i-1$ and an independent set I of size $j-1$, such that C is a maximal clique in G, and every vertex $v \in C$ has at least one neighbor in I. Note that such a graph G exists due to the assumption that $i, j \geq 3$. This graph G has $i+j-2$ vertices, and G contains neither a clique of size i nor an independent set of size j. $\square$

Theorem 10. *Let $\mathcal{G}$ be the class of threshold graphs. Then $R_{\mathcal{G}}(i,j) = i+j-2$ for all $i, j \geq 3$.*

Proof. Let G be a threshold graph on at least $i+j-2$ vertices whose vertices are partitioned into a clique C and an independent set I, such that I is a maximal independent set. Then there is a vertex $v \in I$ that is adjacent to all the vertices in C. We claim that G contains a clique of size i or an independent set of size j. This is clearly the case if $|C| \geq i-1$, since $C \cup \{v\}$ is a clique. Suppose $|C| \leq i-2$. Then, since $|C|+|I| = |V(G)| \geq i+j-2$, we know that $|I| \geq j$. This implies that $R_{\mathcal{G}}(i,j) \leq i+j-2$ for all $i, j \geq 3$.

To prove the matching lower bound, let G be the threshold graph obtained from a clique of size $i-2$ by adding $j-1$ independent vertices, and making every vertex of the clique adjacent to each of these independent vertices. It is easy to verify that G contains neither a clique of size i nor an independent set of size j. Since $|V(G)| = i+j-3$, we conclude that $R_{\mathcal{G}}(i,j) \geq i+j-2$ and hence $R_{\mathcal{G}}(i,j) = i+j-2$ for all $i, j \geq 3$. $\square$

We conclude this section by considering two subclasses of perfect graphs to which Obervation 2 does not apply.

Theorem 11. *Let $\mathcal{G}$ be the class of bipartite graphs or the class of forests. Then $R_{\mathcal{G}}(i,j) = 2j-1$ for all $i \geq 3$ and $j \geq 1$.*

Proof. We first prove the theorem when $\mathcal{G}$ is the class of bipartite graphs. Suppose that G is a bipartite graph on at least $2j-1$ vertices with partition (A, B) of its vertices into two independent sets. Then A or B contains at least j vertices. Hence $R_{\mathcal{G}}(i,j) \leq 2j-1$ for all $i \geq 3$ and $j \geq 1$ when $\mathcal{G}$ is the class of bipartite graphs. Since forests are bipartite, this upper bound applies also when $\mathcal{G}$ is the class of forests. For the lower bound, consider a path on $2j-2$ vertices, which is a forest and hence a bipartite graph on $2j-2$ vertices containing neither a clique of size i nor an independent set of size j. $\square$

4.2 Circular-arc graphs

In this subsection, we determine all Ramsey numbers for circular-arc graphs and proper circular-arc graphs. We will use the same approach as for fuzzy circular interval graphs. In particular, we will use the following lemma, which strongly resembles Lemma 4, in combination with Lemma 5.

Lemma 6. *Every circular-arc graph G contains a clique X of size at most $\omega(G)-1$ such that $G-X$ is an interval graph.*

Proof. Let $G = (V, E)$ be a circular-arc graph with circular-arc model $(G, \mathcal{I})$, where $\mathcal{I}$ is a set of closed intervals (arcs) of a circle $\mathcal{C}$. Without loss of generality, we assume that no two intervals in $\mathcal{I}$ share an endpoint. Recall that the vertices of G correspond to the intervals in $\mathcal{I}$, and not to points of the circle $\mathcal{C}$ as is the case in the definition of (fuzzy) circular interval graphs. We first show that there is a point q on $\mathcal{C}$ such that at most $\omega(G)-1$ intervals of $\mathcal{I}$ contain q. Let $I \in \mathcal{I}$, and let a be an endpoint of I. If at most $\omega(G)-1$ intervals of $\mathcal{I}$ contain a, then we can simply take $q = a$. Suppose exactly $\omega(G)$ intervals of $\mathcal{I}$ contain a. Let P be the set of $2|V|$ points on $\mathcal{C}$ that are endpoints of intervals in $\mathcal{I}$. Let $p \in P \setminus \{a\}$ be such that p does not belong to I, and there is no point of $P \setminus \{a, p\}$ that lies between a and p. Then we can choose q to be any point on the circle between a and p; any such point q is contained in exactly $\omega(G)-1$ intervals of $\mathcal{I}$, namely all those that contain a, apart from interval I. Let $\mathcal{I}_q \subseteq \mathcal{I}$ denote the subset of intervals containing q.

Now let X denote the subset of vertices of G corresponding to the intervals containing q. It is clear that X is a clique of size at most $\omega(G)-1$. Moreover, the graph $G-X$ is an interval graph, as we can obtain an interval model of $G-X$ from the circular-arc model $(G, \mathcal{I} \setminus \mathcal{I}_q)$ by cutting the circle $\mathcal{C}$ at point q to obtain a line. □

We now determine all Ramsey numbers for circular-arc graphs and proper circular-arc graphs that were not covered by Observations 1 and 2.

Theorem 12. *Let $\mathcal{G}$ be the class of circular-arc graphs or the class of proper circular-arc graphs. Then $R_{\mathcal{G}} = (i-1)j$ for all $i, j \geq 3$.*

Proof. Let $G = (V, E)$ be a circular-arc graph, and let i and j be two integers such that $i, j \geq 3$. Suppose $\omega(G) < i$ and $\alpha(G) < j$. Let $X \subseteq V$ be a clique of size at most $i-2$ such that $G-X$ is an interval graph; the existence of such a clique is guaranteed by Lemma 6. Since all interval graphs are perfect, Lemma 5 implies that G has at most $(i-1)j-1$ vertices. Consequently, we have that $R_{\mathcal{G}}(i,j) \leq (i-1)j$ if $\mathcal{G}$ is the class of circular-arc graphs. Since proper circular-arc graphs form a subclass of circular-arc graphs, the same trivially holds if $\mathcal{G}$ is the class of proper circular-arc graphs. The matching lower bound follows immediately from the fact that the graph $G^*_{i,j}$, constructed in the proof of Theorem 8, is a circular interval graph, and hence a proper circular-arc graph, on $(i-1)j-1$ vertices with $\omega(G^*_{i,j}) < i$ and $\alpha(G^*_{i,j}) < j$. □

4.3 Cactus graphs

Finally we determine all Ramsey numbers for cactus graphs. There exist several equivalent definitions of cactus graphs in the literature. Before we give the definition that will be used in the proof of Theorem 13 below, we first recall some terminology.

Let $G = (V, E)$ be a graph. A *cut vertex* of G is a vertex whose deletion strictly increases the number of connected components. A maximal connected subgraph without a cut vertex is called a *block*. Let $G = (V, E)$ be a connected graph, let $A \subseteq V$ be the set of cut vertices in G, and let $\mathcal{B}$ be the set of blocks of G. The *block graph* of G is the bipartite graph with vertex set $A \cup \mathcal{B}$ such that there is an edge between two vertices $a \in A$ and $B \in \mathcal{B}$ if and only if $a \in V(B)$. It is well-known that the block graph of a connected graph is a tree [11]. We define the *block forest* of a graph to be the disjoint union of the block graphs of its connected components.

A graph G is a *cactus graph* if every block of G with more than two vertices is a cycle.[2] Equivalently, a graph G is a cactus graph if every edge of G is contained in at most one cycle. The class of cactus graphs forms a subclass of planar graphs that contains all forests.

Theorem 13. *Let $\mathcal{G}$ be the class of cactus graphs. Then*

$$R_{\mathcal{G}}(i,j) = \begin{cases} \lfloor \frac{5}{2}(j-1) \rfloor + 1 & \textit{if } i = 3 \\ 3(j-1)+1 & \textit{if } i \geq 4 \end{cases}$$

for every pair of integers $i \geq 3$ and $j \geq 1$.

[2] In some definitions, a cactus graph is required to be connected. We do not impose this restriction on cactus graphs, but point out that Theorem 13 would still hold if we did; to see this, it suffices to observe that the graph G_j, constructed in the proof of Theorem 13, can easily be turned into a *connected* cactus graph by making an arbitrarily chosen vertex in one connected component adjacent to exactly one vertex in each of the other connected components.

Proof. It is well-known that every cactus graph is outerplanar, and that every outerplanar graph G is 3-colorable [20]. Hence the function f, defined by $f(x) = 3$ for every $x \in \mathbb{N}$, is a χ-bounding function for the class of cactus graphs, and Lemma 2 implies that $R_{\mathcal{G}}(i,j) \leq 3(j-1)+1$ for all $i, j \geq 1$. For every $i \geq 4$ and $j \geq 1$, the disjoint union of $j-1$ triangles is a cactus graph on $3(j-1)$ vertices that has neither a clique of size i nor an independent set of size j. Consequently, $R_{\mathcal{G}}(i,j) \geq 3(j-1)+1$, and hence $R_{\mathcal{G}}(i,j) = 3(j-1)+1$, for every $i \geq 4$ and $j \geq 1$.

Now suppose $i = 3$. We show, by induction on j, that for every $j \geq 1$, every cactus graph that contains neither a clique of size 3 nor an independent set of size j has at most $\lfloor \frac{5}{2}(j-1) \rfloor$ vertices. The statement trivially holds when $j = 1$. Suppose G is a cactus graph that contains neither a clique of size 3 nor an independent set of size j for some $j \geq 2$. If G has no edges, then $|V(G)| \leq j-1 \leq \lfloor \frac{5}{2}(j-1) \rfloor$, so we assume that G has at least one edge. Then G has a block B on at least two vertices such that B has degree at most 1 in the block forest of G. This block B is either a connected component of G, or B contains exactly one cut vertex b that has neighbors in the graph $G' = G - V(B)$. In either case, since G is triangle-free, B has an independent set S of size $\lfloor |V(B)|/2 \rfloor$ such that S does not contain b. Then G' does not contain an independent set of size $j - \lfloor |V(B)|/2 \rfloor$, as otherwise the union of this set and S would be an independent set of size j in G. Since $\lfloor |V(B)/2| \rfloor \geq 1$, the induction hypothesis guarantees that G' has at most $\lfloor \frac{5}{2}(j-1-\lfloor |V(B)|/2 \rfloor) \rfloor$ vertices. Consequently, $|V(G)| = |V(G')| + |V(B)| \leq \lfloor \frac{5}{2}(j-1-\lfloor |V(B)|/2 \rfloor) \rfloor + |V(B)| = \lfloor \frac{5}{2}(j-1) - \frac{5}{2}\lfloor |V(B)|/2 \rfloor + |V(B)| \rfloor$. Recall that G is triangle-free, so $|V(B)| \neq 3$. Since $\frac{5}{2}\lfloor |V(B)|/2 \rfloor + |V(B)| \leq 0$ for every $|V(B)| \geq 2$ with $|V(B)| \neq 3$, we find that $|V(G)| \leq \lfloor \frac{5}{2}(j-1) \rfloor$. This implies that $R_{\mathcal{G}}(3,j) \leq \lfloor \frac{5}{2}(j-1) \rfloor + 1$ for every $j \geq 1$.

It remains to prove the matching lower bound. In the proof of Theorem 3, we constructed, for every integer $j \geq 1$, a line graph G_j on $\lfloor (5j-3)/2 \rfloor - 1 = \lfloor \frac{5}{2}(j-1) \rfloor$ vertices satisfying $\omega(G_j) < 3$ and $\alpha(G_j) < j$. Since G_j is a disjoint union of copies of C_5 and possibly one copy of K_2, it is clear that G_j is a cactus graph for every $j \geq 1$. This implies that $R_{\mathcal{G}}(3,j) \geq \lfloor \frac{5}{2}(j-1) \rfloor + 1$ for every $j \geq 1$. □

5 Conclusion

Given the difficulty of determining all Ramsey numbers for claw-free graphs [19] and our positive results on several important subclasses of claw-free graphs, another interesting class to consider is the class of quasi-line graphs. A graph is a *quasi-line graph* if the neighborhood of every vertex can be covered by two cliques. Quasi-line graphs form a subclass of claw-free graphs [8] and a superclass of both line graphs and fuzzy circular interval graphs. Is it possible to determine all Ramsey numbers for quasi-line graphs?

Acknowledgment

We are grateful to Daniël Paulusma for bringing references [1] and [10] to our attention.

References

1. N. Balachandran, and N. Khare. Graphs with restricted valency and matching number. Discrete Math. 309:4176–4180, 2009.
2. R. Belmonte, P. Heggernes, P. van 't Hof, and R. Saei. Ramsey numbers for line graphs and perfect graphs. COCOON 2012, LNCS 7434: 204–215, Springer.

3. F. Bonomo, G. Durán, L. N. Grippo, and M. D. Safe. Partial characterizations of circular-arc graphs. J. Graph Theory 61(4):289–306, 2009.
4. A. Brandstädt, V. B. Le, and J. P. Spinrad. Graph classes: a survey. SIAM, 1999.
5. S. A. Burr, P. Erdős, R. J. Faudree, R. H. Schelp. On the difference between consecutive Ramsey numbers. Utilitas Mathematica 35:115–118, 1989.
6. M. Chudnovsky and M. Plumettaz. The structure of claw-free perfect graphs. Submitted for publication. Preprint available at `http://www.columbia.edu/~mc2775/clawsperfect.pdf`.
7. M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. Ann. Math. 164:51–229, 2006.
8. M. Chudnovsky and P. Seymour. The structure of claw-free graphs. Surveys in Combinatorics 2005, London Math. Soc. Lecture Note Ser., 327.
9. M. Chudnovsky and P. Seymour. Claw-free graphs III: Circular interval graphs. J. Comb. Theory Ser. B 98:812–834, 2008.
10. V. Chvátal and D. Hanson. Degrees and matchings. J. Comb. Theory Ser. B 20:128–138, 1976.
11. R. Diestel. Graph Theory. Springer-Verlag, Electronic Edition, 2005.
12. G. Exoo. On the Ramsey number $R(4,6)$. Electr. J. Comb. 19, #P66, 2012.
13. K. Fraughnaugh Jones. Independence in graphs with maximum degree four. J. Comb. Theory Ser. B 37:254–269, 1984.
14. K. L. Fraughnaugh and S. C. Locke. Lower bounds on size and independence in K_4-free graphs. J. Graph Theory 26:61–71, 1997.
15. J. Goedgebeur and S. P. Radziszowski. New computational upper bounds for Ramsey numbers $R(3,k)$. Manuscript available at `arxiv.org/abs/1210.5826`.
16. M. C. Golumbic. Algorithmic Graph Theory and Perfect Graphs. Ann. Discrete Math. 57, Elsevier, 2004.
17. R. L. Graham, B. L. Rothschild, and J. H. Spencer. Ramsey Theory. Second Edition, Wiley, 1990.
18. G. Oriolo, U. Pietropaoli, G. Stauffer. On the recognition of fuzzy circular interval graphs. Discrete Math. 312:1426–1435, 2012.
19. M. M. Matthews. Longest paths and cycles in $K_{1,3}$-free graphs. J. Graph Theory, 9:269–277, 1985.
20. A. Proskurowski and M. M. Sysło. Efficient vertex- and edge-coloring of outerplanar graphs. SIAM J. Alg. Disc. Meth. 7:131–136, 1986.
21. S. P. Radziszowski. Small Ramsey numbers. Electr. J. Comb., Dynamic Surveys, 2011.
22. F. P. Ramsey. On a problem of formal logic. Proc. London Math. Soc. Series 30:264–286, 1930.
23. J. H. Spencer. *Ten Lectures on the Probabilistic Method*, SIAM, 1994.
24. W. Staton. Some Ramsey-type numbers and the independence ratio. Trans. Amer. Math. Soc. 256:353–370, 1979.
25. R. Steinberg and C. A. Tovey. Planar Ramsey numbers. J. Comb. Theory Ser. B 59:288–296, 1993.
26. K. Walker. The analog of Ramsey numbers for planar graphs. Bull. London Math. Soc. 1:187–190, 1969.