Graph Transversals for Hereditary Graph Classes: a Complexity Perspective

GIACOMO PAESANI

DURHAM UNIVERSITY

DOCTORAL THESIS

Graph Transversals for Hereditary Graph Classes:
a Complexity Perspective

Author:
Giacomo PAESANI

Supervisors:
Daniël PAULUSMA
Matthew JOHNSON

A thesis submitted for the degree of Doctor of Philosophy

Algorithms and Complexity in Durham
Department of Computer Science

August 2021

Abstract

Within the broad field of Discrete Mathematics and Theoretical Computer Science, the theory of graphs has been of fundamental importance in solving a large number of optimization problems and in modelling real world situations. In this thesis we study a topic that covers many aspects of Graph Theory: transversal sets. A transversal set in a graph G is a vertex set that intersects every subgraph of G that belongs to a certain class of graphs. The focus is on *vertex cover, feedback vertex set* and *odd cycle transversal*.

The decision problems Vertex Cover, Feedback Vertex Set and Odd Cycle Transversal ask, for a given graph G and an integer $k \geq 0$, whether there is a corresponding transversal of G of size at most k. These problems are NP-complete in general and our focus is to determine the complexity of the problems when various restrictions are placed on the input, both for the purpose of finding tractable cases and to increase our understanding of the point at which a problem becomes NP-complete. We consider graph classes that are closed under vertex deletion and in particular H-free graphs, i.e. graphs that do not contain a graph H as induced subgraph.

The first chapter is an introduction to the thesis. There we illustrate the motivation of our work and introduce most of the terminology we have used for our research. In the second chapter, we develop a number of structural results for some classes of H-free graphs.

The third chapter looks at the Subset Transversal problems: there we prove that Feedback Vertex Set and Odd Cycle Tranversal and their *subset* variants can be solved in polynomial time for both P_4-free and $(sP_1 + P_3)$-free graphs, while for Subset Vertex Cover we show that it can be solved in polynomial time for $(sP_1 + P_4)$-free graphs.

The fourth chapter is entirely dedicated to the Connected Vertex Cover problem. The connectivity constraint requires additional proof techniques. We prove this problem can be solved in polynomial time for $(sP_1 + P_5)$-free graphs, even when *weights* are given to the vertices of the graph.

We continue the research on *connected* transversals in the fifth chapter: we show that Connected Feedback Vertex Set, Connected Odd Cycle Transversal and their *extension* variants can be solved in polynomial time for both P_4-free and $(sP_1 + P_3)$-free graphs.

In the sixth chapter we study the price of independence: can the size of a smallest *independent* transversal be bounded in terms of the minimum size of a transversal? We establish complete and almost-complete dichotomies which determine for which graph classes such a bound exists and for which cases such a bound is the identity.

Contents

Tables

Figures

Declaration of Authorship

No part of this thesis has been previously been submitted for any degree at any institution. Most of the results contained in this thesis have appeared, often in preliminary form, in the papers [9,17,35,36,37,64], all of which have been subject to peer review. Each section is inspired by the results from one or more of these papers. At the beginning of each chapter, we mention where the result in every section of this chapter has been published. Although the results in this thesis are obtained by joint research, I have actively participated in the discussions leading to these results and my contribution to them has been significant.

This work was supported by a Durham Doctoral Studentship.

Acknowledgements

The list of people I should cite to thank is so long that I will avoid to do that. Simply put, any of you has given so much motivation and support: reaching this goal without you would have been way harder and your help has been fundamental to hang tough. Nevertheless I wand to spend a few words for some individuals.

First I thank my parents, Costanza and Roberto, and my brother, Stefano, for emphasising the importance of education in life and inspiring profound passion towards Science and Mathematics.

I would also like to thank my supervisor Prof. Daniël Paulusma. I came to Durham with a mentality that a student had to be in total subjection towards his/her professors not only on the professional aspect but also on human relationship point of view. During these years Daniël and his constructive attitude helped me to break this stereotype: we usually spend part of our meetings talking regarding family health situations, TV series and updates on Covid-19. Of course I could praise him for a very long time on technical aspects and for very fruitful discussions regarding our topics but it is not the point of these lines. Too bad this virus kept us apart for the last year of my studies in Durham.

Notable mention has to be done to my second supervisor Prof. Matthew Johnson. Thank you for your support and presence when Daniël could not be available. Without your assistance in the background I would have been lost in the middle of my time in Durham.

Next I would love to thank all the co-authors that, together with my supervisors, contributed to publish our research: Hans L. Bodlaender, Nick Brettell, Konrad K. Dabrowski, Carl Feghali, Jake Horsfield, Andrea Munaro, Paweł Rzążewski, Viktor Zamaraev and Erik Jan van Leeuwen. Hope I have not slowed your research too much and our collaborations have been enjoyable. I wish we could meet some time in the future for additional fruitful discussions, in case my work stays in the research field (as I hope). Working with you has widened my horizons and enriched me as a student and as a person: many thanks.

Finally I want to spend some lines on Durham, the Department, the University and its community as a whole. With many ups and downs this experience served me to grow so much as an individual that I could never expect. Despite I did not join the college and social life in Durham, I have appreciated the structure and the very positive aspects of it. The setting of the town was extraordinary and it will have always a special place in my heart.

From a passionate and enthusiastic myself, cheers!

1 Introduction

A graph is a model that represents, mostly binary, relationships between objects; it is composed of a set of items, which we call vertices, and a set of relationships among them, which we call edges. For example, consider a number of individuals in a social network: each person can be considered as a vertex and an edge is present between two individuals if they are friends. A common question is to determine if there is a group of people large enough in a social network such that every person has a friendship with everybody else in the group; the term *clique* is associated to such groups. This concept has appeared in the mathematical literature with a paper by Erdős and Szekeres [40] in 1935, linked to the famous Ramsey Theory. Later Luce and Perry [77] introduced cliques into the social sciences. Finally in 1957, Harary and Ross [58] began to study the algorithmic aspects of finding cliques. Maximum Clique and other related problems have a central role in Graph Theory and Theoretical Computer Science.

Fig. 1: A City Map with Streets and Crossings

Another common application of graphs is in the study of physical maps: for example, in Figure 1 we see a small part of a city map where streets are the edges and their crossings form the vertex set of a graph. One might be interested to compute the shortest route among two vertices of a map: this problem is formally known as Shortest Path. Another interesting problem on maps inspects if it is possible to place a bounded number of cameras on crossings in order to cover every street: in a more formal setting the

VERTEX COVER problem requires to select a bounded set of vertices that transverse every edge of the graph. In this regard both these problems are central to the study of algorithms on graphs.

The SHORTEST PATH problem can be seen as the discrete version of finding a geodesic, i.e. a curve between two points of minimum length, in a metric space and has became a classical problem in Graph Theory and has been generalized in a large variety of ways to answer different questions. A large body of literature can be found in [27] where different aspects of the problem are considered. Frequently we use the following variant of this problem: we ask if it is possible to connect a given set of vertices with a limited number of edges; this variant is known as STEINER TREE.

The *vertex cover* concept is very important in this thesis and is part of a group of definitions that are closely related. Pairs of vertices that do not belong to a vertex cover can not be linked by an edge, or the vertex cover definition is contradicted; these vertex sets are also know as *independent sets* and the problem asking if there exists an independent set of bounded size is called INDEPENDENT SET. The operation of graph complementation, i.e. the replacement of every edge with a non-edge, and vice versa, transforms independent sets into cliques, and vice versa. In this sense vertex cover, independent set and clique are strongly related concepts for general graphs.

VERTEX COVER is an NP-*complete* problem: we do not know any efficient algorithm for solving VERTEX COVER and and many Computer Scientists believe there are not any. In practice however, input graphs usually have a certain structure and such structure helps in the design of efficient algorithms. In the study of NP-complete problems, it is common among Computer Scientists to determine graph classes such that the problem is still NP-complete when restricted to such classes or their properties can be exploited to design efficient algorithms.

Topics in Graph Theory are usually motivated by interesting questions regarding the physical world and other field of science but they commonly have relevance and interest on their own. To properly study these combinatorial objects we need to give precise definitions, statements and algorithms; moreover, we have to adopt a mathematical attitude to manage all of them together.

The vertex cover definition is part of a large group of concepts that are known as *graph transversals*. A graph transversal is a vertex subset of some given graph that intersects all the subgraphs belonging to a predefined set of graphs. When a transversal exists, usually the aim is to minimize its size. For very specific and more common predefined sets, many variants and generalizations have been analysed over the years. We survey both *computational complexity* and *structural* results around the most common graph transversals and their variants, obtained adding extra constrains, when the input is *restricted to some special graph class*. Before presenting these results we first state the necessary definitions and terminology.

1.1 Basic Graph Terminology

While self-loops, directed and multiple edges can be relevant and generalizing factors in the study of transversals, we will assume such structures are not present in the graphs dealt in this thesis to exclude extra complications. Hence, we only consider finite undirected graphs with no multiple edges or self loops, that is, a *graph* G is an ordered pair (V, E), where V is a finite set of elements called *vertices* and E is a set of unordered pairs uv, with $u, v \in V$ and $u \neq v$, called *edges*. The sets V and E are called the *vertex set* and *edge set* of G, respectively. In some situations we write $V(G)$ and $E(G)$ instead of V and E for a graph G whenever the topic includes more than one graph and there is a risk of ambiguity. Moreover for a graph $G = (V, E)$, let n be the number of vertices $|V|$ and m be the number of edges $|E|$ in the graph G.

A graph H is a *subgraph* of a graph G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ having only edges contained in $V(H)$. A graph H is an *induced* subgraph of a graph G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ having all the edges contained in $V(H)$. We write $H \subseteq G$ and $H \subseteq_i G$ to denote that H is a subgraph or an induced subgraph of G, respectively. For a subset $S \subseteq V$, we let $G[S]$ and $G - S$ denote the induced subgraph of G with $V(G[S]) = S$ and $G[V \setminus S]$.

For an integer $r \geq 1$, the graph P_r denotes the *path* on r vertices, i.e., $V(P_r) = \{v_1, ..., v_r\}$ and $E(P_r) = \{v_i v_{i+1} \mid 1 \leq i \leq r - 1\}$. For an integer $r \geq 3$, the graph C_r denotes the *cycle* on r vertices, i.e., $V(C_r) = \{v_1, ..., v_r\}$ and $E(C_r) = \{v_i v_{i+1} \mid 1 \leq i \leq r-1\} \cup \{v_1 v_r\}$. The *length* of a path or cycle is the number of its edges. A cycle or path is even or odd depending on the parity of its length. The graph K_r denotes the *complete* graph on r vertices, i.e., $V(K_r) = \{v_1, ..., v_r\}$ and $E(K_r) = \{v_i v_j \mid 1 \leq i < j \leq r\}$. The vertex set of a complete graph is called a *clique*. The graph rP_1 denotes the graph on r vertices with empty edge set; such vertex sets are called *independent*. We say that a graph G is *connected* if for every pair of distinct vertices u and v, there is a path connecting u and v.

Let $G = (V, E)$ be a graph. A *vertex-weighting* of G is a function $w_V : V \to \mathbb{Q}^+$ (where $\mathbb{Q}^+$ denotes the set of strictly positive rational numbers), that is, each vertex v has an associated positive rational weight $w_V(v)$. Moreover an *edge-weighting* of G is a function $w_E : E \to \mathbb{Q}^+$, that is, each edge e has an associated positive rational weight $w_E(e)$. The weight of a set of vertices, or edges, is the sum of the weights of its elements.

For two graphs G and H, a vertex mapping $f : V(G) \to V(H)$ is called a *graph isomorphism* when $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$. In that case we say that G and H are *isomorphic*. In some situations, with an abuse of notation, we write that two graphs are equal or are the same when instead we mean those two graphs are isomorphic.

1.2 Basic Complexity Theory

Computational complexity studies how much time and space are involved and necessary to solve problems. Algorithms plays a central role in this investigation, since they are often the tools we use to solve such problems. Moreover, as the complexity of an algorithm is always an upper bound on the complexity of the problem solved by that algorithm, it is common to focus on its performance.

We put particular emphasis on the time resource: the number of required elementary operations, which are assumed to take a constant amount of time, on a given input express the time complexity.

In these settings it is common to use the so called big O notation: let $f : \mathbb{R}^+ \longrightarrow \mathbb{R}$ and $g : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ two functions, we say $f(n) = O(g(n))$ if there exists a real number $M > 0$ and and integer n_0 such that $|f(n)| \leq Mg(n)$ for all $n \geq n_0$. Informally we say that an algorithm runs in $O(g(n))$-time on an input of size n if its running time does not grow faster than $g(n)$.

The class P contains all problems that are solvable in polynomial time, and the class NP contains all problems for which a candidate solution can be verified in polynomial time. Trivially $\mathsf{P} \subseteq \mathsf{NP}$ holds. On the other side it is widely believed that $\mathsf{P} \neq \mathsf{NP}$, but this is a major unsolved problem in Theoretical Computer Science. A polynomial time reduction, or simply a reduction, for a problem π to another problem π' is an algorithm that runs in polynomial time, for transforming any input of π into an equivalent input of π'.

A problem is called NP-complete if it is in NP and every problem in NP can be reduced into this problem in polynomial time; informally the class of NP-complete problems contains the hardest problems in NP. A problem is called NP-hard if every other problem in NP can be reduced in polynomial time to it; informally the class of NP-hard problems contains all the problems that are at least as hard as the hardest problems in NP. Figure 2 describes the relationships between the described complexity classes.

A parametrized problem consists of a tuple (π, k) where π is the problem instance and k is the parameter. A parametrized problem is said to be *fixed parameter tractable* or FPT if there exists an algorithm for the problem with time complexity $O(f(k) \cdot |\pi|^{O(1)})$, where f is a function of k alone and $|\pi|$ represents the size of the input instance.

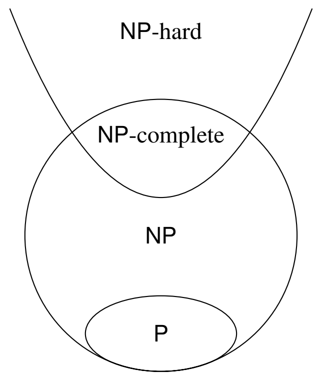


Fig. 2: Euler diagram for P, NP, NP-complete and NP-hard set of problems under the assumption P≠NP.

Given a graph G with n vertices and m edges, the computational complexity of an algorithm on G is expressed as a function of n and m. For example, the breadth-first search algorithm [81] runs in $O(n + m)$ time.

1.3 Graph Transversal Terminology

Let $\mathcal{H}$ be a possibly infinite set of graphs and $G = (V, E)$ be a graph; a subset $S \subseteq V$ is an *$\mathcal{H}$-transversal* of G if $G - S$ contains no subgraphs isomorphic to an element of $\mathcal{H}$. Let $\mathcal{H}$ be a set of graphs, we can define the following decision problem:

$\mathcal{H}$-Transversal
Instance: a graph $G = (V, E)$ and a positive integer k.
Question: does G have an $\mathcal{H}$-transversal S with $\lvert S \rvert \leq k$?

The following definitions introduce notable graph transversals that are obtained by choosing $\mathcal{H}$ in a very natural way. A set S is a *vertex cover* if it is a $\{P_2\}$-transversal. A set S is a *feedback vertex set* if it is a $\{C_3, C_4, C_5, ...\}$-transversal. A set S is an *odd cycle transversal* if it is a $\{C_3, C_5, C_7, ...\}$-transversal.

Every vertex cover is also a feedback vertex set,
which is, in turn, an odd cycle transversal.

Now we can define a decision problem for each transversal as follows:

Vertex Cover
Instance: a graph $G = (V, E)$ and a positive integer k.
Question: does G have a vertex cover S with $\|S\| \leq k$?

Feedback Vertex Set
Instance: a graph $G = (V, E)$ and a positive integer k.
Question: does G have a feedback vertex set S with $\|S\| \leq k$?

Odd Cycle Transversal
Instance: a graph $G = (V, E)$ and a positive integer k.
Question: does G have an odd cycle transversal S with $\|S\| \leq k$?

By [67], Vertex Cover and Feedback Vertex Set are NP-complete and Cygan et al. [33] proved the same for Odd Cycle Transversal. Due to these fundamental results, the aim of this thesis is to consider input graphs that belong to some special graph class in order to understand the graph properties that force intractability or allow polynomial-time solvability for these problems.

1.4 Variants of Graph Transversals

One can be interested to slightly modify definitions and questions of a research topic to get a wider understanding of it. To what extent can we use or adapt proof techniques, tools and results for the modified setting? How far is it reasonable to push the changes such that the results provide useful indications for the original topic? For the sake of comparison, it is important to highlight differences and common points between the modified and original versions of the subject: in this way we do not only obtain information on how hard it is to deal with such topic but we also show the strengths and weaknesses of those changes.

The goal of this thesis is to contribute to the process of relating graph transversals and their variants. Given the fact there is already a large body of literature on the topic, which we survey in Sections 3.1,4.1,5.1 and 6.1, our aim is to obtain complexity dichotomies and increase our understanding of the structural properties.

We study four variants of graph transversal, two of which result in generalizations and the other two are specializations. Let $G = (V, E)$ be a graph, $W \subseteq V$ and $\mathcal{H}$ be a set of graphs; a subset $S_W \subseteq V$ is an $\mathcal{H}$-transversal *extension* of W if S is an $\mathcal{H}$-transversal that contains W. Note that if $W = \emptyset$ this definition coincides with the original one. Let

$G = (V, E)$ be a graph, $T \subseteq V$ and $\mathcal{H}$ be a set of graphs; a subset $S_T \subseteq V$ is a *subset* $\mathcal{H}$-transversal of T if $G - S_T$ contains no subgraphs isomorphic to an element of $\mathcal{H}$ that contains a vertex of T. Note that if $T = V$ this definition coincides with the original one. Let $G = (V, E)$ be a graph and a set of graphs $\mathcal{H}$; a subset $S \subseteq V$ is a *connected* $\mathcal{H}$-transversal if it is an $\mathcal{H}$-transversal that induces a connected subgraph. Let $G = (V, E)$ be a graph and a set of graphs $\mathcal{H}$; a subset $S \subseteq V$ is an *independent* $\mathcal{H}$-transversal if it is an $\mathcal{H}$-transversal that induces an independent set. We define the corresponding decision problems as follows:

$\mathcal{H}$-TRANSVERSAL EXTENSION
Instance: a graph $G = (V, E)$, $W \subseteq U$, a set of graphs $\mathcal{H}$ and a positive integer k.
Question: does G have an $\mathcal{H}$-transversal S_W containing W with $|S_W| \leq k$?

SUBSET $\mathcal{H}$-TRANSVERSAL
Instance: a graph $G = (V, E)$, $T \subseteq U$, a set of graphs $\mathcal{H}$ and a positive integer k.
Question: does G have an $\mathcal{H}$-transversal S_T of T with $|S_T| \leq k$?

CONNECTED $\mathcal{H}$-TRANSVERSAL
Instance: a graph $G = (V, E)$, a set of graphs $\mathcal{H}$ and a positive integer k.
Question: does G have a connected $\mathcal{H}$-transversal S with $|S| \leq k$?

INDEPENDENT $\mathcal{H}$-TRANSVERSAL
Instance: a graph $G = (V, E)$, a set of graphs $\mathcal{H}$ and a positive integer k.
Question: does G have a independent $\mathcal{H}$-transversal S with $|S| \leq k$?

Moreover, we can define another generalization of a transversal decision problem as follows:

WEIGHTED $\mathcal{H}$-TRANSVERSAL
Instance: a graph $G = (V, E)$, a vertex-weighting function w_V, a set of graphs $\mathcal{H}$ and a positive rational k.
Question: does G have an $\mathcal{H}$-transversal S with $w_V(S) \leq k$?

For example, we can define VERTEX COVER EXTENSION, SUBSET VERTEX COVER, CONNECTED FEEDBACK VERTEX SET, INDEPENDENT FEEDBACK VERTEX SET or WEIGHTED ODD CYCLE TRANSVERSAL; moreover, it is possible to mix two or more variants together in order to obtain different case studies, like SUBSET CONNECTED FEEDBACK VERTEX SET EXTENSION.

We often consider the optimization version of these transversal problems, in which case we are asked to find an *optimal* solution. Note that while problems like Vertex Cover, Feedback Vertex Set and Odd Cycle Transversal are all NP-complete problems by [33,67], the corresponding optimization problems are NP-hard. There is a linear time reduction of a decision problem to the corresponding optimization version: let (I, k) be an input of such decision problem, the optimal solution S for input I can be accepted or not depending on its value when compared to k.

1.5 More Graph Terminology

For a subset $F \subseteq E(G)$, $G - F$ denotes the graph obtained from G by removing the edge set F. We say that a subgraph H *spans* a graph G if $H = G - F$, for some edge set $F \subseteq E(G)$. The *union* of two graphs G and H is the graph with $V(G) \cup V(H)$ as vertex set and $E(G) \cup E(H)$ as edge set. If no vertex is in common, that is, $V(G) \cap V(H) = \emptyset$, then we call the union of G and H the *disjoint* union of G and H, denoted $G + H$. The disjoint union of r copies of G is denoted by rG. The *join* of two graphs G and H is the graph with $V(G) \cup V(H)$ as vertex set and the edge set is obtained from $E(G) \cup E(H)$ by adding all possible edges between $V(G)$ and $V(H)$. If no vertex is in common, then the join of G and H is denoted by $G \times H$. Let S and T be two disjoint vertex sets, then we say S is *complete* to T if every vertex of S is adjacent to every vertex of T, i.e. $G[S \cup T] = G[S] \times G[T]$, and S is *anti-complete* to T if there are no edges between S and T, i.e. $G[S \cup T] = G[S] + G[T]$.

Let $G = (V, E)$ be a graph. The *degree* $deg_G(u)$ of a vertex $u \in V$ is the number of edges incident with it, or equivalently the size of its *neighbourhood* $N_G(u) = \{v \in V \mid uv \in E\}$; the *closed neighbourhood* $N_G[u]$ is defined to be $N_G(u) \cup \{u\}$. For a vertex set $U \subseteq V$ we can equivalently define its neighbourhood and closed neighbourhood as the sets $N_G(U) = (\bigcup_{u \in U} N_G(u)) \setminus U$ and $N_G[U] = N_G(U) \cup U$, respectively: the neighbourhood of a vertex set contains all the vertices not in the set that are adjacent to it.

A vertex of degree 0 is an *isolated* vertex. If a graph is not connected, then it is called *disconnected* and can be seen as the disjoint union of its maximal connected induced subgraphs, called *connected components*. The *girth* of G is the length of a shortest cycle in G; if G has no cycle, then the girth of G is equal to $+\infty$. The *complement* of G, denoted by $\overline{G}$, has V as vertex set and an edge between two distinct vertices if and only if these vertices are not adjacent in G. A set $D \subseteq V$ is a *dominating set* of G if every vertex of the set $V \setminus D$ is adjacent to at least one vertex of D, that is, $N(D) = V \setminus D$. An edge uv dominates G if $\{u, v\}$ is dominating. A *matching* in a graph is a set of pairwise disjoint edges. A matching is *perfect* if every vertex of the graph is contained in one edge of the matching.

Let k be a natural number, a graph $G = (V, E)$ is called *k-connected* if $|V| > k$ and the graph $G - U$ is connected for every set $U \subseteq V$ with $|U| < k$. Note that every (non-empty) graph is 0-connected, and the 1-connected graphs are precisely the non-trivial connected graphs. We say a graph is *biconnected* if it is 2-connected or K_2. A *block* of a graph is a maximal biconnected subgraph and is *non-trivial* if it contains a cycle, or, equivalently, is on at least three vertices. A *block decomposition* of a graph is a partition of its vertex set into blocks and it is well known that this can be found in $O(n + m)$ time (see e.g. [62]).

We say that we *identify* two vertices u and v in a graph G if from $G - \{u, v\}$ we add a new vertex that is adjacent to $N_G(\{u, v\})$. If $uv \in E(G)$, then this operation is also called an *edge contraction*. For a subset $F \subseteq E(G)$, G/F denotes the graph obtained from G by contracting the edge set F. A graph G contains a graph H as a *minor* if a subgraph of G can be modified into H by a sequence of edge contractions. We write $H \subseteq_m G$ to denote that H is a minor of G. For any integer $k \geq 1$, we say that we *subdivide* an edge $e = uv$ k-times or apply a *k-subdivision* on e, if we replace e with a path having the vertices u and v as endpoints having exactly k new internal vertices.

A *colouring* of a graph $G = (V, E)$ is a mapping from the vertex set V to a finite set of positive integers, i.e., $\phi : V \rightarrow \{1, 2, ..., t\}$ for some $t \geq 1$, such that $\phi(u) \neq \phi(v)$ whenever $uv \in E$. A *k-colouring* of G is a colouring ϕ of G with $1 \leq \phi(v) \leq k$ for all $v \in V$. In that case we say G is *k-colourable*. Equivalently, a graph is k-colourable if we can partition its vertex set into k (possibly empty) independent sets (called *colour classes* or *partition classes*). The smallest integer k for which a graph G is k-colourable is called the *chromatic number* of G, denoted by $\chi(G)$.

1.6 Special Graph Classes

In this section we give the definitions of a number of graph classes known in the literature.

Let G be a graph and $\{H_1, ..., H_p\}$ be a set of graphs. We say that G is *$(H_1, ..., H_p)$-free* if G has no induced subgraph isomorphic to a graph in $\{H_1, ..., H_p\}$; we may write H-free instead of (H)-free. In a similar fashion we can define that G is *$(H_1, ..., H_p)$-subgraph-free* or *$(H_1, ..., H_p)$-minor-free* if G has no subgraph or no minor isomorphic to a graph in $\{H_1, ..., H_p\}$, respectively. Note that if H' is an induced subgraph of H, every H'-free graph is also H-free.

A graph class $\mathcal{G}$ is called *hereditary* if it is closed when taking induced subgraphs, that is, if $G \in \mathcal{G}$ then $G' \in \mathcal{G}$, for every $G' \subseteq_i G$. It is well-known that a graph class $\mathcal{G}$

is hereditary if and only if $\mathcal{G}$ is the class of $\mathcal{F}_{\mathcal{G}}$-free graphs, for a possibly infinite set of graphs $\mathcal{F}_{\mathcal{G}}$ (see e.g. [42]). Most of the graph classes that we have introduced and are going to define are hereditary, when possible we specify the minimal set of forbidden subgraphs $\mathcal{F}_{\mathcal{G}}$. As we have seen already, the classes of complete graphs and of independent sets coincide with $2P_1$-free and K_2-free graphs, respectively.

For an integer $r \geq 1$, a graph is *r-partite* if its vertex set can be partitioned into r non-empty sets $A_1, ..., A_r$ such that no edge is contained in A_i, for $1 \leq i \leq r$. A 2-partite graph is also called *bipartite*. The class of bipartite graphs coincides with $(C_3, C_5, C_7, ...)$-free graphs. For integers $r \geq 1$ and $s \geq 1$, the graph $K_{r,s}$ denotes the *complete bipartite* graph with partition classes of size r and s, respectively, i.e. $V(K_{r,s}) = \{u_1, ..., u_r\} \cup \{v_1, ..., v_s\}$ and $E(K_{s,r}) = \{u_i v_j \mid 1 \leq i \leq r \text{ and } 1 \leq j \leq s\}$, alternatively $K_{r,s} = rP_1 \times sP_1$. For an integer $r \geq 1$, the graph $K_{1,r}$ is also called a *star*; in particular the graph $K_{1,3}$ is called the *claw*. For an integer $r \geq 1$, let $K^+_{1,r}$ denote the graph obtained from $K_{1,r}$ by subdividing one edge. See Figure 3 for drawings of these graphs. For integers $p \geq 1$ and $q \geq 1$, the *double star* $D_{p,q}$ is a graph with $V(D_{p,q}) = \{x, y\} \cup \{u_1, \ldots u_p\} \cup \{v_1, \ldots, v_q\}$ and $E(D_{p,q}) = \{xy\} \cup \{xu_i \mid 1 \leq i \leq p\} \cup \{yv_j \mid 1 \leq j \leq q\}$ (see also Figure 21 for an example).

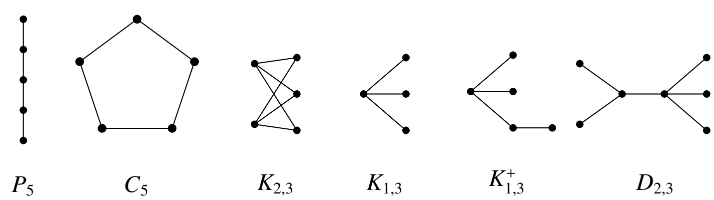


Fig. 3: Notable graphs.

A graph is a *tree* if it is connected and without cycles. In a tree a vertex of degree one is called *leaf*, while all the vertices of degree at least two are known as *internal vertices*. A graph is a *forest*, if each connected component is a tree. A graph is a *linear forest*, if each connected component is a path. While the class of trees is not hereditary, the classes of forests and of linear forests coincide with $(C_3, C_4, C_5, ...)$-free and $(K_{1,3}, C_3, C_4, C_5, ...)$-free graphs, respectively.

A graph is *perfect* if the chromatic number of every induced subgraph equals the size of a largest clique in that subgraph. By the Strong Perfect Graph Theorem [30], a graph is perfect if and only if it is $(C_5, \overline{C_5}, C_7, \overline{C_7}, C_9, \overline{C_9}, ...)$-free. A graph is a *permutation* graph if line segments connecting two parallel lines can be associated to its vertices in such a way that two vertices are adjacent if and only if their corresponding line segments intersects. A graph is an *interval* graph if intervals of the real line can be associated to its vertices in such a way that two vertices are adjacent if and only if their corresponding intervals overlap.

A *chord* of a cycle C is an edge between two vertices $u, v \in V(C)$ with $uv \notin E(C)$. A graph is *chordal* if every cycle on four or more vertices has a chord. Equivalently, a graph is chordal if and only if it is $(C_4, C_5, C_6, ...)$-free. A graph G is a *split* graph if its vertex set can be partitioned into a clique and an independent set. Split graphs coincide with $(2P_2, C_4, C_5)$-free graphs [44]. The *line graph* of a graph $G = (V, E)$ is the graph $L(G)$ with E as the vertex set and $e, e' \in E$ are adjacent in $L(G)$ if and only if e and e' share an end-vertex in G. By a classical result by Beineke [6], the class of line graphs is characterized by a set of nine forbidden induced subgraphs, which contains the claw $K_{1,3}$.

A graph class is called *minor-hereditary* if it is closed under minors. In a long series of papers (see from [93] to [94]) Robertson and Seymour proved that any minor-hereditary graph class can be defined by a finite set of forbidden minors. A graph is *planar* if it can be drawn in the plane so that its edges can intersect only at their end-vertices. By Wagner's Theorem [98], a graph is planar if and only if it is $(K_5, K_{3,3})$-minor-free.

The graph complementation operation gives a way to define a number of graph classes. Given a graph class $\mathcal{G}$, we let *co-*$\mathcal{G}$ be the *complementary* class of $\mathcal{G}$ that is obtained from $\mathcal{G}$ by complementing every graph in the class. For example, cobipartite graphs are all the graphs which complements are bipartite. Note that if a graph class $\mathcal{G}$ is hereditary then also co-$\mathcal{G}$ is so.

1.7 Why Hereditary Graph Classes?

Recall the definition of a hereditary graph class: it is a class of graphs that is closed when taking induced subgraphs. Independently from the containment relation chosen for our research, we want to emphasize the importance of considering graph classes closed under such relation. Loosely speaking, containment relations of graphs allow one to modify a graph into another graph by the use of a given set of rules, sometimes called operations. It is a very common use in structural and algorithmic Graph Theory, and more in general in the whole field of science, to be interested in the effects of modifying a given object following specific steps. To what extent the properties of such object can be extended to the product of its modification? Which properties are preserved at any point of those steps? While a general and complete answer is impossible to give, considering graph classes closed under a given containment relation has important consequences on our research.

Deleting a vertex, along with all adjacent edges, from a graph is the only operation we are allowed to apply when considering the induced subgraph relation. This allows to preserve the (non-)adjacency of pairs of vertices that have not been deleted. This operation set of taking induced subgraphs is limited, compared to the subgraph and minor relations (those are allowed to delete edges and, for the minor relation, also to contract edges), but has notable strengths. For any set of graphs $\mathcal{H}$, the class of $\mathcal{H}$-free graphs contains the class of $\mathcal{H}$-subgraph-free graphs which contains the class of $\mathcal{H}$-minor-free graphs: these containments hold but equalities hold only for very specific cases of $\mathcal{H}$. Moreover hereditary graph classes capture a very relevant part of the Graph Theory literature and research. Most of the graph classes we define, as already noted, are hereditary.

Selecting an $\mathcal{H}$-transversal S of a graph $G = (V, E)$ corresponds to a partition $\{S, V \setminus S\}$ of V such that $G - S$ contains no subgraphs isomorphic to an element of $\mathcal{H}$. Loosely speaking we can think of V as a starting pool and we study the process of placing its elements into the pools S and $V \setminus S$ as efficiently and quickly as possible while satisfying said condition on $G - S$. The operation of vertex deletion serves not only to stay inside an hereditary graph class but also to craft such partition. Edge deletion and edge contraction do not provide the same benefits in our settings: a profitable use of edge deletion would require to change the definition of $\mathcal{H}$-transversal from being a vertex set to an edge set.

Finally we want to discuss our choice regarding the definition of an $\mathcal{H}$-transversal S for a graph G: we require $G - S$ to contain no subgraph isomorphic to an element of the set $\mathcal{H}$, that is, $G - S$ is $\mathcal{H}$-subgraph-free. We compare this definition with the one requiring $G - S$ to be $\mathcal{H}$-free. For vertex cover, feedback vertex set and odd cycle transversal, these definitions are the same, since for every edge, cycle and odd cycle there is an induced one "contained" in it. The situation changes when we study, for example, subset odd cycle transversal.

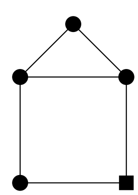


Fig. 4: The square vertex of the House forms a set T.

For an example of subset odd cycle transversal, see Figure 4: there is a unique odd T-cycle but it is not induced.

1.8 Overview of the Thesis

In the rest of the chapters of this work we analyse different aspects of transversals, both recalling known results from literature and including original work. Recall that every hereditary graph class $\mathcal{G}$ can be characterized by a possibly infinite set $\mathcal{F}_{\mathcal{G}}$ of forbidden induced subgraphs. This enables us to initiate a *systematic* study, starting from the case where $|\mathcal{F}_{\mathcal{G}}| = 1$.

In Chapter 2, we list and prove a sequence of structural lemmas from [17,35,64] regarding some notable subclasses of H-free graphs, especially when H is a linear forest. We use these structural results in later chapters to create efficient algorithms that solve the transversal problems.

Chapter 3 deals with algorithmic aspects regarding all the original transversal problems and their subset variant. The original results spring from *On cycle transversals and their connected variants in the absence of a small linear forest* [35] while most of them have been generalized in *Computing subset transversals in H-free graphs* [17] after considering the subset version of the problems. In particular we prove Subset Vertex Cover is polynomial time solvable on $(sP_1 + P_4)$-free graphs, while the same result holds for Subset Feedback Vertex Set and Subset Odd Cycle Transversal on both P_4-free and $(sP_1 + P_3)$-free graphs. On the other hand Odd Cycle Transversal and Subset Odd Cycle Transversal are proved to be NP-complete on $(P_2 + P_5, P_6)$-free and split graphs, respectively.

Chapter 4 is completely dedicated to Connected Vertex Cover and *Connected vertex cover for $(sP_1 + P_5)$-free graphs* [64] proves it is polynomial time solvable on $(sP_1 + P_5)$-free graphs, even for the weighted version. In [35] we note that this result holds also for the extension variant of the problem.

In Chapter 5 we research the case where the connected and the extension variants are combined together for all the other transversal problems on H-free graphs. In particular with our work [35] we prove that Connected Feedback Vertex Set Extension and Connected Odd Cycle Transversal Extension are polynomial time solvable for both P_4-free and $(sP_1 + P_3)$-free graphs. A brief introduction to the Steiner Tree problem obtained from *Steiner trees for hereditary graph classes: a treewidth perspective* [9] allows us to develop a strategy to solve Weighted Connected $\mathcal{H}$-Transversal Extension, for any graph set $\mathcal{H}$, at the cost of strong limitations on the input.

In Chapter 6 we resume our research on structural properties: we analyse for which graph H, the size of a minimum independent transversal of an H-free graph can be lower bounded by a function of the size of a minimum transversal of the same graph. In *On the price of independence for vertex cover, feedback vertex set and odd cycle transversal* [36] we study when such bounding function exists or not, while in *Independent transversals versus transversals* [37] we check when the bounding function is the identity. Finally we merge the results to express explicit values of the bounding functions and prove some of them are tight.

At the end of each chapter, we discuss open questions and explore different directions for future research regarding each topic.

2 Interesting Classes of H-free Graphs

In this thesis we put great emphasis on the fact we are working with hereditary graph classes. For this reason it is important to highlight relevant structural properties of these classes for the most frequent cases. In Section 2.1 we analyse the case of P_4-free graphs, including a decomposition and efficient recognition result. In Section 2.2, we study the class of $(sP_1 + P_3)$-free graphs, for any integer $s \geq 0$; there we prove a number of stuctural results which provide great support for later chapters.

Finally in Section 2.3 we write more in general regarding P_r-free graphs. First we showcase literature examples where there is a complexity jump from P_r-free graphs to P_{r+1}-free graphs on different problems. Then we provide more structural results on these graph classes that serve as auxiliary tools for later proofs.

Before we examine in depth the structure of graphs in these hereditary graph classes, we want to explain how we use the properties proved in this chapter to create efficient algorithms that solve the decision problems dealt in this thesis. We consider different cases that correspond to different structures present in a transversal of a graph and describe polynomial-time subroutines that find a minimum transversal for each case. We obtain an optimal solution by running each of these subroutines in turn: for each case we obtain a potential solution and we output the one with minimum size overall.

2.1 Case: $H = P_4$

The class of P_3-free graphs can be easily described as exactly those graphs that are disjoint unions of cliques: every connected component of a P_3-free graph has diameter at most one, and so it is a clique. The class of P_4-free graphs, that is, graphs that do not contain an induced path on four vertices, have a more complex structure and have a relevant role on our research. A graph is a *cograph* if it can be generated from K_1 by a sequence of join and disjoint union operations. A graph is a cograph if and only if it is P_4-free (see e.g. [14]). The following lemma is well known, but we include a short proof for completeness.

Lemma 1. *Every connected P_4-free graph on at least two vertices has a spanning complete bipartite subgraph which can be found in polynomial time.*

Proof. Let $G = (V, E)$ be a connected P_4-free graph on at least two vertices. Then there is a partition $V = X \cup Y$ such that G is the join of $G[X]$ and $G[Y]$. Hence, G has a spanning complete bipartite subgraph with partition classes X and Y. Note that this implies that $\overline{G}$ is disconnected. In order to find a (not necessarily unique) spanning complete bipartite subgraph of G with partition classes X and Y in polynomial time, we put the vertices of one connected component of $\overline{G}$ in X and all the other vertices of $\overline{G}$ in Y. □

It is also well known (see e.g. [31]) that a graph G is a cograph if and only if G allows a unique cotree decomposition called the *cotree* T_G of G, which has the following properties:

1. The root r of T_G corresponds to the graph $G_r = G$.
2. Each leaf x of T_G corresponds to exactly one vertex of G, and vice versa, hence x corresponds to a unique single-vertex graph G_x.
3. Each internal node x of T_G has at least two children, is labelled + or ×, and corresponds to an induced subgraph G_x of G defined as follows:
   - if x is a +-node, then G_x is the disjoint union of all graphs G_y where y is a child of x;
   - if x is a ×-node, then G_x is the join of all graphs G_y where y is a child of x.
4. Labels of internal nodes on the (unique) path from any leaf to r alternate between + and ×.

Note that T_G has $O(n)$ vertices. We modify T_G into a *modified cotree* T'_G in which each internal node has exactly two children by applying a well-known procedure (see e.g. [10]). If an internal node x of T_G has more than two children y_1 and y_2, remove the edges xy_1 and xy_2 and add a new vertex x' with edges xx', $x'y_1$ and $x'y_2$. If x is a +-node, then x' is a +-node. If x is a ×-node, then x' is a ×-node. Applying this rule exhaustively yields T'_G. As T_G has $O(n)$ vertices, constructing T'_G from T_G takes linear time.

The following result, due to Corneil, Perl and Stewart, proves cographs can be recognized efficiently.

Lemma 2 **([32]).** *Let $G = (V, E)$ be a graph with n vertices and m edges. Then deciding whether or not G is a cograph, and constructing a modified cotree T'_G (if it exists), takes $O(n + m)$ time.*

2.2 Case: $H = sP_1 + P_3$

The class of H-free graphs, when $H = sP_1 + P_3$ for some $s \geq 1$, plays a central role in this research: we developed many polynomial-time results for this class of graphs and those results are often the best possible for the moment, in the sense that for no graph $H \supset_i sP_1 + P_3$ such polynomial-time result is known.

Let G be an $(sP_1 + P_3)$-free graph; if we remove from G an induced P_3 and its neighbours, we are left with a graph having at most $s - 1$ independent vertices. In the same way, if we remove from G a set of s independent vertices and their neighbours, we are left with a P_3-free graph (that is, a disjoint union of complete graphs). In this sense, the class of $(sP_1 + P_3)$-free graphs can be seen as the generalization of the class of sP_1-free and of P_3-free graphs.

In this section we gather a number of preliminary lemmas that assist to exploit properties and structure of graphs in this class.

Let us define a function a on non-negative integers by $a(s) := \max\{7, 4s - 2\}$.

Lemma 3. *Let s be a non-negative integer, and let R be an $(sP_1 + P_3)$-free tree. Then either*

(i) *$|V(R)| \leq a(s)$, or*

(ii) *R has precisely one vertex r of degree more than 2 and at most $s - 1$ vertices of degree 2, each adjacent to r. Moreover, r has at least $3s - 1$ neighbours.*

Proof. If R has no vertices of degree more than 2, then R is a path and has at most $2s + 2 \leq a(s)$ vertices, otherwise R has an induced $sP_1 + P_3$ subgraph. Now let r be a vertex of degree more than 2, and let x, y and z be distinct neighbours of r. We view r as the root of the tree, and for $v \in V(R)$ we use R_v to denote the subtree rooted at v.

Suppose that R_x has a vertex of degree at least 2. Then R_x has an induced P_3 subgraph, so $R - (V(R_x) \cup \{r\})$ is sP_1-free, and hence, by [87, Observation 1], this subtree consists of at most $2(s - 1)$ vertices. Likewise, $R[\{y, r, z\}] = P_3$, so $R_x - x$ is sP_1-free, and hence consists of at most $2(s - 1)$ vertices. Thus $|V(R)| \leq 2(s - 1) + 2(s - 1) + 2 = 4s - 2$.

We may now assume that for each $v \in N(r)$, the subtree R_v has no vertices of degree at least 2; that is, either $R_v = P_1$ or $R_v = P_2$. It remains to show that when (i) does not hold, at most $s - 1$ of the R_v subgraphs are isomorphic to P_2. Towards a contradiction, suppose that R has s vertices at distance 2 from r, and $|V(R)| > a(s)$. Since $|V(R)| > 2(s + 1) + 1$ for any non-negative integer s, the vertex r has at least $s + 2$ neighbours. Without loss of generality, label the neighbours of r as $v_1, v_2, \ldots, v_{deg(r)}$ such that $R_{v_i} = P_2$ for each $i \in \{1, \ldots, s\}$. Then $R[v_{s+1}, r, v_{s+2}] = P_3$, and $R_{v_i} - \{v_i\} = P_1$ for each $i \in \{1, \ldots, s\}$; a contradiction.

Finally, $|N_R(r)| + (s - 1) + 1 \geq |V(R)| \geq 4s - 1$, so $|N_R(r)| \geq 3s - 1$. □

Lemma 4. *Let $s \geq 0$ be an integer. Let $R = (V, E)$ be an $(sP_1 + P_3)$-free tree. Then R has at most $4s$ internal vertices.*

Proof. Let U be the set of internal vertices of R. Suppose that $|U| \geq 4s + 1 \geq 1$. We will show that this leads to a contradiction. As a path with at least $4s + 1$ internal vertices

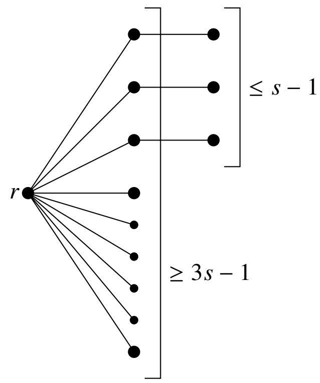


Fig. 5: The structure of an $(sP_1 + P_3)$-free tree, as given by Lemma 3, when (i) does not hold.

contains an induced $sP_1 + P_3$, we may assume that R is not a path and so has at least three leaves. Hence $|V| \geq 4s + 4$.

Let X and Y be the two bipartition sets of R, and assume without loss of generality that $|X| \geq 2s+2$. For $Z \in \{X, Y\}$, let L_Z and U_Z be the leaves and internal vertices of R that belong to Z, respectively. If there is a vertex in Y of degree at least 2 that is anti-complete to a set of s vertices of X, then R contains an induced sP_1+P_3, a contradiction. Therefore we may assume that every vertex of Y either has degree at least $|X| - s + 1$ or is in L_Y. Then

$$\begin{aligned}
|X| + |U_Y| + |L_Y| - 1 &= |X| + |Y| - 1 \\
&= |V| - 1 \\
&= |E| \\
&= \sum_{v \in Y} \deg(v) \\
&\geq \sum_{v \in U_Y} (|X| - s + 1) + |L_Y| \\
&= (|X| - s + 1)|U_Y| + |L_Y| \\
&= |X| \cdot |U_Y| - s|U_Y| + |U_Y| + |L_Y|.
\end{aligned}$$

Thus we have $|X| - 1 \geq |X||U_Y| - s|U_Y|$ and we rearrange to see that

$$|U_Y| \leq \frac{|X| - 1}{|X| - s} = 1 + \frac{s - 1}{|X| - s}.$$

Since $|X| \geq 2s + 2$, we have that $|U_Y| < 2$. First suppose $|U_Y| = 0$. Then $|U_X| \leq 1$ and $|L_X| = 0$, or $|U_X| = 0$ and $|L_X| \leq 1$. Both cases contradict the assumption that X has at least $2s + 2$ vertices. Now suppose $|U_Y| = 1$. Then, by our assumption that $|U| \geq 4s + 1$, we have that $|U_X| \geq 4s$ and so $|L_Y| \geq |U_X| \geq 4s$. Now it is easy to find an induced $sP_1 + P_3$ (see Figure 6), and this contradiction completes the proof. □

The bound of $4s$ in Lemma 4 is not tight but, as we shall see later, it suffices for our purposes.

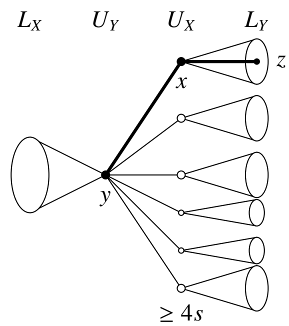


Fig. 6: The structure of the tree R in the proof of Lemma 4 in the case when $|U_Y| = 1$. The set L_X is an independent set of vertices and each of them is adjacent to the unique vertex $y \in U_Y$. The set L_Y is partitioned into independent sets of vertices that have the same neighbour in U_X. The vertices y, x, z, together with s vertices of L_y not adjacent to x, induces an $sP_1 + P_3$ in R (which leads to the desired contradiction in the proof).

Let us define a function b on non-negative integers by $b(s) := \max\{3, 2s - 1\}$.

Lemma 5. *Let $s \geq 0$ be an integer. Let B be a bipartite $(sP_1 + P_3)$-free graph. If B has a connected component on at least $b(s)$ vertices, then there are at most $s - 1$ other connected components of B and each of them is on at most two vertices.*

Proof. First note that the $s = 0$ case of the lemma is trivially true, as every connected component of a bipartite P_3-free graph has at most two vertices.

Suppose, for contradiction, that B has a connected component C_1 on at least $b(s)$ vertices and a connected component C_2 on at least three vertices. As C_1 is bipartite and contains at least $2s - 1$ vertices, C_1 contains an independent set of s vertices that induce sP_1. As C_2 is bipartite and contains at least three vertices, C_2 has a vertex v of degree at least 2, and so v and two of its neighbours induce a P_3. Thus G is not $(sP_1 + P_3)$-free, a contradiction.

Similarly, if B contains a connected component C_1 on at least $b(s) \geq 3$ vertices, then this connected component contains an induced P_3. Since B is $(sP_1 + P_3)$-free, B can contain at most $s - 1$ connected components other than C_1. □

Lemma 6. *Let $s \geq 0$ be an integer. Let G be a connected $(sP_1 + P_3)$-free graph and let U be a set of vertices in G. Then there is a set of vertices R in G such that $G[R \cup U]$ is connected and $|R| \leq 2s^2 - 2s + 3$.*

Proof. If $G[U]$ is connected, then let $R = \emptyset$. Otherwise, since G cannot now be a complete graph, it contains an induced path P on three vertices in G. The number of connected components of $G[U]$ that do not contain a vertex that is either in P or adjacent to a vertex of P in G is at most $s - 1$, otherwise G contains an induced $sP_1 + P_3$. Let R contain the vertices of P and the internal vertices of shortest paths in G from P to each set of vertices that induces a connected component of $G[U]$. As at most $s - 1$ of these shortest paths have more than zero internal vertices, and as each contains at most $2s$ internal vertices (any longer path contains an induced $sP_1 + P_3$, it follows that $|R| \leq 3 + 2s(s - 1) = 2s^2 - 2s + 3$. As $G[R \cup U]$ is connected, the lemma is proved. □

2.3 Case: *H* is a Linear Forest

The computational complexity of many problems jump from polynomial-time solvable on P_r-free graphs to NP-complete on P_{r+1}-free graphs. For instance, Colouring is polynomial-time solvable for P_4-free graphs but is NP-complete for P_5-free graphs [69].

A *clique transversal* of a graph G is a set $S \subseteq V$ such that S contains a vertex of each maximal clique of G (note that a vertex cover can be viewed as a transversal which contains a vertex of each 2-vertex clique). It is known that computing a smallest clique transversal can be done in polynomial time for comparability graphs [4] and thus for P_4-free graphs, but is NP-hard for cobipartite graphs [57] and thus for P_5-free graphs.

We will use the following result of Bacsó and Tuza [3] in a successive proof.

Lemma 7 ([3]). *Every connected P_5-free graph G has a dominating set D, computable in $O(n^3)$ time, that induces either a P_3 or a complete graph.*

This also follows from a more general result of Camby and Schaudt [25] for P_r-free graphs.

Lemma 8 ([25]). *Let $k \geq 4$ be an integer. Every connected P_k-free graph G has a dominating set D that induces either a P_{k-2} or a P_{k-2}-free graph.*

We use Lemma 7 to prove the next one.

Lemma 9. *Let $s \geq 0$ and let G be a connected $(sP_1 + P_5)$-free graph. Then G has a connected dominating set D that is either a clique or has size at most $2s^2 + s + 2$. Moreover, D can be found in $O(n^{2s^2+s+3})$ time.*

Proof. If G is P_5-free, then we apply Lemma 7 to find, in $O(n^3)$ time, a set D that either induces a P_3 or is a clique. Otherwise, as G is $(sP_1 + P_5)$-free, there exists an integer $0 \leq r \leq s - 1$ such that G contains an induced subgraph H isomorphic to $rP_1 + P_5$. Let $V(H) = \{a_1, \ldots, a_r, b_1, \ldots, b_5\}$ such that the vertex set $\{b_1, b_2, b_3, b_4, b_5\}$ induce a P_5. We choose r to be maximum, so G contains no induced $(r + 1)P_1 + P_5$. Hence, $V(H)$ dominates G. As G is $(sP_1 + P_5)$-free, G is also P_{5+2s}-free. Hence, for each a_i, there exists a path of at most $5 + 2s - 1$ vertices that connects a_i to b_1. Let H^* be the graph that contains H and all these $a_i - b_1$-paths. Then we choose $D = V(H^*)$. As $V(H)$ dominates G, we find that $D \supseteq V(H)$ also dominates G. Moreover, D has size at most $r(5 + 2s - 2) + 5 \leq 2s^2 + s + 2$. We can find D by considering, if needed, every set of at most $2s^2 + s + 2$ vertices in G and by checking if each such a set is dominating. The latter takes $O(n)$ time per set. Hence, this brute force procedure takes $O(n^{2s^2+s+3})$ time in total. □

The following lemma expresses that the class of connected H-free graphs is closed under edge contractions, whenever H is a linear forest.

Lemma 10. *Let H be a linear forest and let G be a connected H-free graph. The graph obtained from G after contracting an edge is also connected and H-free.*

Proof. Let e be an edge of E and consider G/e: the graph obtained from G by contracting the edge e. Let v_e be the vertex of G/e created by the contraction of e. Note that G/e is trivially connected. For contradiction suppose G/e contains an induced subgraph H' that is isomorphic to H and let $H'' \subseteq_i G$ be the graph that is obtained from H' by uncontracting the edge e. If $v_e \notin V(H')$ then $H'' = H'$ and we are done. Now we can assume $v_e \in V(H')$. Since H' is a linear forest, it is easy to note that H'' (and so G) contains H' as an induced subgraph; a contradiction. □

3 Subset Transversal

For a graph $G = (V, E)$ and a set $T \subseteq V$, a *T-edge* or a *T-cycle* is, respectively, an edge or a cycle of G that intersects T. A set $S_T \subseteq V$ is a *T-vertex cover*, a *T-feedback vertex set* or an *odd T-cycle transversal* of G if S_T has at least one vertex of, respectively, every T-edge, every T-cycle or every odd T-cycle. For example, let G be a star, whose leaves form the set T. Then, both $V \setminus T$ and T are T-vertex covers of G but the first is considerably smaller than the second. See Figures 7 and 8 for some more examples.

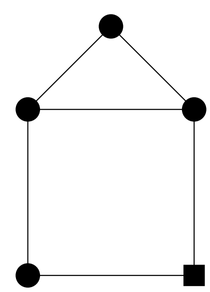


Fig. 7: The square vertex of the House forms the set T. This graph contains only one (not induced) odd T-cycle (containing all the vertices of the graph). Any vertex of the House is a (minimum) odd T-cycle transversal.

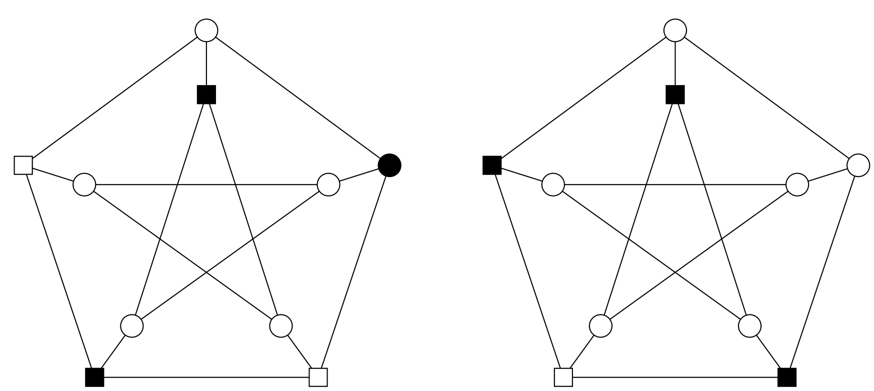


Fig. 8: In both examples, the square vertices of the Petersen graph form a set T and the black vertices form a T-feedback vertex set S_T. In the left example, $S_T \cap (V \setminus T) \neq \emptyset$, and in the right example, $S_T \subseteq T$.

Now we can formally state the three transversal problems of this section.

Subset Vertex Cover *Instance:* a graph $G = (V, E)$, a subset $T \subseteq V$ and a positive integer k. *Question:* does G have a T-vertex cover S_T with $|S_T| \leq k$?

Subset Feedback Vertex Set *Instance:* a graph $G = (V, E)$, a subset $T \subseteq V$ and a positive integer k. *Question:* does G have a T-feedback vertex set S_T with $|S_T| \leq k$?

Subset Odd Cycle Transversal *Instance:* a graph $G = (V, E)$, a subset $T \subseteq V$ and a positive integer k. *Question:* does G have an odd T-cycle transversal S_T with $|S_T| \leq k$?

The Subset Feedback Vertex Set and Subset Odd Cycle Transversal problems are well known. The Subset Vertex Cover problem is introduced in our paper [17], and we are not aware of past work on this problem. On general graphs, Subset Vertex Cover is polynomially equivalent to Vertex Cover: to solve Subset Vertex Cover remove edges in the input graph that are not incident to any vertex of T to yield an equivalent instance of Vertex Cover . However, this equivalence no longer holds for graph classes that are *not* closed under edge deletion.

Since, in the case $T = V$ these subset transversal problems are equivalent to their respective original ones, the three problems are NP-complete [33,67], we consider the restriction of the input to hereditary graph classes in order to better understand which graph properties cause the computational hardness. In order to initiate a *systematic* study, we start our research from the hereditary graph classes defined by forbidding a single graph.

Lemma 11. *Let S be a minimum solution for an instance (G, T) of a subset transversal problem. Then $|S \setminus T| \leq |T \setminus S|$.*

Proof. For contradiction, assume that $|S \setminus T| > |T \setminus S|$. Then $|T| < |S|$ (see also Figure 9). This means that T is a smaller solution than S, a contradiction. □

A subgraph of G is a *T-forest* if it has no T-cycles. A subgraph of G is *T-bipartite* if it has no odd T-cycles. A subgraph of G is *T-path* if it is a path that contains a vertex of T. A T-path is *odd* or *even* depending on the parity of the path.

We will use the following lemma, which proves that T-forests and T-bipartite graphs can be recognized in polynomial-time. It combines results claimed but not proved in [73,88].

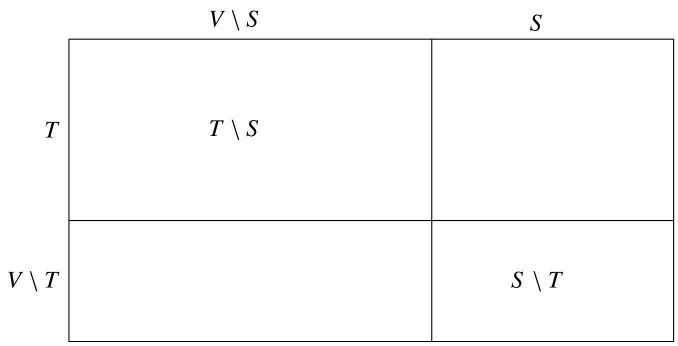


Fig. 9: For a minimum solution S for an instance (G, T) of a subset transversal problem, it must hold that $|S \setminus T| \leq |T \setminus S|$ (see also Lemma 11).

Lemma 12. *Let $G = (V, E)$ be a graph with n vertices and m edges and $T \subseteq V$. Then deciding whether or not G is a T-forest or T-bipartite takes $O(n + m)$ time.*

Proof. Suppose that we have a block decomposition of G; which can be found in $O(n+m)$ time with the breadth-first search algorithm [81]. It is clear that G is a T-forest if and only if no non-trivial block contains a vertex of T. We claim that G is T-bipartite if and only if no non-bipartite block contains a vertex of T. To see this note first that the sufficiency is obvious. We will show that if a vertex t of T belongs to a block B that contains an odd cycle C, then t belongs to an odd cycle. If t is in C, we are done. Otherwise find two paths P and P' from t to, respectively, distinct vertices u and u' in C. We can assume that the paths contain no other vertex of C (else we truncate them) and that, as B is 2-connected, they contain no common vertex other than t. We can form two cycles that contain t by adding to $P + P'$ each of the two paths between u and u' in C. As C is an odd cycle, the lengths of these two paths, and therefore the lengths of the two cycles, have distinct parity. Thus t belongs to an odd cycle. Finally we note that the checks of the block decomposition needed to decide whether or not G is a T-forest or T-bipartite can be done in $O(n + m)$ time. □

One could also define and study the extension version for any (subset) transversal problem. However, such extension version will be polynomially equivalent to the (subset) problem. Indeed, we can solve the extension version on the input (G, W, k) by considering the original problem on the input $(G - W, \max\{0, k - |W|\})$ and adding W to the solution.

3.1 Existing Results.

First we start with results of the classical versions. By Nagamochi and Xiao Vertex Cover can be solved in $O(1.1996^n)$ time using polynomial space on general graphs [102] and in $O(1.0836^n)$ time when restricted to graphs of maximum degree 3 [101]. Using Poljak's construction [89], Vertex Cover is readily seen to be NP-complete for graphs of arbitrarily large girth and thus for H-free graphs whenever H contains a cycle. Moreover Vertex Cover is NP-complete for planar graphs [49], cubic graphs [48] and more generally for k-regular graphs for any fixed k [47].

Vertex Cover becomes polynomial-time solvable on perfect graphs [54,55] and on claw-free graphs [78,95] and thus for line graphs. By combining two classical results [5,96] Vertex Cover is polynomial-time solvable for sP_2-free graphs for any integer $s \geq 1$. These results have been generalized further in four different ways: by Alekseev [2], Lozin and Milanič [74] for $K_{1,3}^+$-free graphs, by Lozin and Mosca for $(P_2 + K_{1,3})$-free graphs [75] and for $2P_3$-free graphs [76] and recently by Brandstädt and Mosca [15] for $sK_{1,3}$-free graphs for any integer $s \geq 1$.

Even the case where H is a single path on r vertices the computational complexity is not settled for Vertex Cover: it is not known if there exists an integer r such that Vertex Cover is NP-complete for P_r-free graphs. Lokshtanov, Vatshelle, and Villanger [72] proved that Independent Set, and thus Vertex Cover, is polynomial-time solvable for P_5-free graphs. Recently, Grzesik, Klimošová, Pilipczuk and Pilipczuk [56] extended this to P_6-free graphs. We also note that if Vertex Cover is polynomial-time solvable on H-free graphs for some graph H, then it is polynomial-time solvable on $(P_1 + H)$-free graphs. This follows from the observation (see, e.g., [82]) that to solve the complementary problem of Independent Set on a $(P_1 + H)$-free graph one solves the problem on each H-free graph obtained by removing a vertex and all its neighbours. This proves the following result:

Theorem 1 ([56]). *For every $s \geq 0$,* Vertex Cover *can be solved in polynomial-time for $(sP_1 + P_6)$-free graphs.*

On general graphs Fomin and Villanger proved Feedback Vertex Set can be solved in $O(1.7347^n)$ time [46], while Raman, Saurabh and Sikdar proved Odd Cycle Transversal can be solved in $O(1.9526^n \cdot n^{O(1)})$ time [92]. By Poljak's construction [89], Feedback Vertex Set is NP-complete for graphs of girth at least g for every integer $g \geq 3$. The same holds for Odd Cycle Transversal [28]. Moreover, Feedback Vertex Set [84] and Odd Cycle Transversal [28] are NP-complete for line graphs and thus for claw-free graphs. Hence, both problems are NP-complete for H-free graphs if H has a cycle or claw. While Okrasa and Rzążewski [85] proved Odd Cycle Transversal is NP-complete for P_{13}-free graphs, there is no known integer r such that Feedback Vertex Set is NP-complete for P_r-free graphs.

Both problems are polynomial-time solvable for P_4-free graphs [13] and for sP_2-free graphs for every $s \geq 1$ [28]. In [35], authors show polynomial-time algorithms that solves these problems for $(sP_1 + P_3)$-free graphs for every $s \geq 1$. Very recently, Abrishami et al. showed that Feedback Vertex Set is polynomial-time solvable for P_5-free graphs [1]. We summarize as follows.

Theorem 2. *For a graph H,* Feedback Vertex Set *on H-free graphs is polynomial-time solvable if* $H \subseteq_i P_5$, $H \subseteq_i sP_1 + P_3$ *or* $H \subseteq_i sP_2$ *for some* $s \geq 1$, *and* NP*-complete if* $H \supseteq_i C_r$ *for some* $r \geq 3$ *or* $H \supseteq_i K_{1,3}$.

Theorem 3. *For a graph H,* Odd Cycle Transversal *on H-free graphs is polynomial-time solvable if* $H = P_4$, $H \subseteq_i sP_1 + P_3$ *or* $H \subseteq_i sP_2$ *for some* $s \geq 1$, *and* NP*-complete if* $H \supseteq_i C_r$ *for some* $r \geq 3$, $H \supseteq_i K_{1,3}$ *or* $H \supseteq_i P_{13}$.

This situation changes for Subset Feedback Vertex Set which is, unlike Feedback Vertex Set, NP-complete for split graphs (that is, $(2P_2, C_4, C_5)$-free graphs), as shown by Fomin et al. [45]. Papadopoulos and Tzimas [87,88] proved that Subset Feedback Vertex Set is polynomial-time solvable for sP_1-free graphs for any $s \geq 1$, co-bipartite graphs, interval graphs and permutation graphs, and thus P_4-free graphs.

We are not aware of any results on Subset Odd Cycle Transversal for H-free graphs, but note that this problem generalizes Odd Multiway Cut, just as Subset Feedback Vertex Set generalizes Node Multiway Cut, another well-studied problem. We refer to a large body of literature [29,34,45,50,60,63,65,68,70,73] for further details, in particular for parameterized and exact algorithms for Subset Feedback Vertex Set and Subset Odd Cycle Transversal. These algorithms are beyond the scope of this thesis.

3.2 Our Results

Our polynomial-time results from [35] for $(sP_1 + P_3)$-free graphs are included in our other paper [17] where we significantly extend the known results for Subset Feedback Vertex Set in Section 3.4 and Subset Odd Cycle Transversal in Section 3.5 on H-free graphs. Moreover in Section 3.5, we prove that Odd Cycle Transversal is NP-complete on $(P_2 + P_5, P_6)$-free graphs and Subset Odd Cycle Transversal is NP-complete on split graphs. These new results lead us to Table 1 and to the following two almost-complete dichotomies:

Theorem 4. *Let H be a graph with $H \neq sP_1 + P_4$ for all $s \geq 1$. Then* Subset Feedback Vertex Set *on H-free graphs is polynomial-time solvable if $H = P_4$ or $H \subseteq_i sP_1 + P_3$ for some $s \geq 1$ and* NP*-complete otherwise.*

Theorem 5. *Let H be a graph with $H \neq sP_1 + P_4$ for all $s \geq 1$. Then* Subset Odd Cycle Transversal *on H-free graphs is polynomial-time solvable if $H = P_4$ or $H \subseteq_i sP_1 + P_3$ for some $s \geq 1$ and* NP*-complete otherwise.*

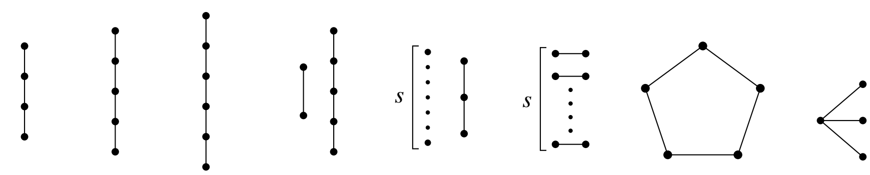


Fig. 10: The forbidden graphs of Theorems 2–5.

Though the proved complexity of Subset Feedback Vertex Set and Subset Odd Cycle Transversal are the same on H-free graphs, the algorithm that we present for Subset Odd Cycle Transversal on $(sP_1 + P_3)$-free graphs is more technical compared to the algorithm for Subset Feedback Vertex Set, and considerably generalizes the transversal algorithms for $(sP_1 + P_3)$-free graphs of [35]. There is further evidence that Subset Odd Cycle Transversal is a more challenging problem than Subset Feedback Vertex Set. For example, the best-known parametrized algorithm for Subset Feedback Vertex Set runs in $O(4^k \cdot n^{O(1)})$ time [63], but the best-known run-time for Subset Odd Cycle Transversal is $O(2^{O(k^3 \log k)} \cdot n^{O(1)})$ [73], where k is the maximum size of a solution.

In Section 3.3 we present some results for Subset Vertex Cover: first we show that Subset Vertex Cover is polynomial-time solvable for $(sP_1 + P_4)$-free graphs for every $s \geq 1$ and later we use this as a subroutine to obtain a polynomial-time algorithm for Subset Odd Cycle Transversal on P_4-free graphs. In Section 3.6 we discuss on future work on the Subset Vertex Cover more in more detail.

3.3 Subset Vertex Cover

In this section we present some results on Subset Vertex Cover.

Lemma 13. Subset Vertex Cover *can be solved in $O(n + m)$ time for P_4-free graphs.*

	girth p	line graphs	sP_2-free	P_r-free	$sP_1 + P_r$-free
VC	NP-c [89]	P [78,95]	P: $s \geq 0$ [15]	P: $r \leq 6$*	P: $s \geq 0$, $r \leq 6$ [56]
FVS	NP-c [89]	NP-c [84]	P: $s \geq 0$ [28]	P: $r \leq 5$ [1]	P: $s \geq 0$, $r \leq 3$*
OCT	NP-c [28]	NP-c [28]	P: $s \geq 0$ [28]	P: $r \leq 4$*	P: $s \geq 0$, $r \leq 3$*
SVC	NP-c*	?	?	P: $r \leq 4$*	P: $s \geq 0$, $r \leq 4$
SFVS	NP-c*	NP-c*	NP-c: $s \geq 2$ [45]	P: $r \leq 4$ [87,88]	P: $s \geq 0$, $r \leq 3$
SOCT	NP-c*	NP-c*	NP-c: $s \geq 2$	P: $r \leq 4$	P: $s \geq 0$, $r \leq 3$

Table 1: The computational complexity of the three transversal problems together with their subset variant on graphs of girth at least p for every (fixed) constant $p \geq 3$, on line graphs, and on H-free graphs for various linear forests H. Results that directly follow for other results in the table while starred and unreferenced results are ours; finally question marks show cases that are left as open problems. Note this table does not completely summarise all the results from our work and from the literature.

Proof. Let G be a P_4-free graph with n vertices and m edges and let $T \subseteq V$. First construct a modified cotree T'_G and then consider each node of T'_G starting at the leaves of T'_G and ending at the root r. Let x be a node of T'_G. We let S_x denote a minimum $(T \cap V(G_x))$-vertex cover of G_x.

If x is a leaf, then G_x is a 1-vertex graph. Hence, we can let $S_x = \emptyset$. Now suppose that x is a +-node. Let y and z be the two children of x. Then, as G_x is the disjoint union of G_y and G_z, we can let $S_x = S_y \cup S_z$. Finally suppose that x is a $\times$-node. Let y and z be the two children of x. As G_x is the join of G_y and G_z we observe the following: if $V(G_x) \setminus S_x$ contains a vertex of $T \cap V(G_y)$, then $V(G_z) \subseteq S_x$. Similarly, if $V(G_x) \setminus S_x$ contains a vertex of $T \cap V(G_z)$, then $V(G_y) \subseteq S_x$. Hence, we let S_x be the smallest set of $S_y \cup V(G_z)$, $S_z \cup V(G_y)$ and $T \cap V(G_x)$.

Constructing T'_G takes $O(n + m)$ time by Lemma 2. As $T_{G'}$ has $O(n)$ nodes and processing a node takes $O(1)$ time, the total running time is $O(n + m)$. □

The following lemma generalizes a corresponding well-known observation (see e.g. [82]) for Subset Vertex Cover .

Lemma 14. *Let H be a graph. If* Subset Vertex Cover *is polynomial-time solvable for H-free graphs, then it is for $(P_1 + H)$-free graphs as well.*

Proof. Let $G = (V, E)$ be a $(P_1 + H)$-free graph and let $T \subseteq V$. Let S_T be a minimum T-vertex cover of G. For each vertex $u \in T$ we consider the option that u belongs to the set $V \setminus S_T$. If so, then $N(u)$ belongs to S_T. Let $G' = G - N[u]$ and let $T' = T \setminus N[u]$.

As G' is H-free, we find a minimum T'-vertex cover $S_{T'}$ of G' in polynomial-time. We remember the smallest set $S_{T'} \cup N(u)$ and compare it with the size of T to find S_T (or some other minimum solution for (G, T)). □

Lemma 13, combined with s applications of Lemma 14, yields the following result.

Theorem 6. *For every integer $s \geq 1$,* Subset Vertex Cover *can be solved in polynomial-time for $(sP_1 + P_4)$-free graphs.*

3.4 Subset Feedback Vertex Set

In this section we prove Theorem 4. Our contribution to it is Theorem 7, which is the case where $H = sP_1 + P_3$. In Section 3.5, we present an analogous result for Subset Odd Cycle Transversal. The proofs are similar in outline, but the latter requires additional insights.

The next lemma shows how we can extend "partial" solutions to full solutions in polynomial-time as follows.

Lemma 15. *Let $G = (V, E)$ be a graph with a set $T \subseteq V$. Let $V' \subseteq V$ and $S'_T \subseteq V'$ such that S'_T is a T-feedback vertex set of $G[V']$, and let $Z = V \setminus V'$. Suppose that $G[Z]$ is P_3-free, and $|N_{G-S'_T}(Z)| \leq 1$. Then there is a polynomial-time algorithm that finds a minimum T-feedback vertex set S_T of G such that $S'_T \subseteq S_T$ and $V' \setminus S'_T \subseteq V \setminus S_T$.*

Proof. Since $G[Z]$ is P_3-free, it is a disjoint union of complete graphs. Let $G' = G - S'_T$. Suppose that C is a T-cycle in G'. Then C contains at least one vertex of Z. If $N_{G'}(Z) = \emptyset$, then C is contained in a connected component of $G[Z]$. On the other hand, if $N_{G'}(Z) = \{y\}$, say, then y is a cut-vertex of G', so there exists a connected component $G[U]$ of $G[Z]$ such that C is contained in $G[U \cup \{y\}]$. Hence, we can consider each connected component of $G[Z]$ independently: for each connected component $G[U]$ it suffices to find the maximum subset U' of U such that $G[U' \cup N_{G'}(U)]$ contains no T-cycles. Then $U' \subseteq F_T$ and $U \setminus U' \subseteq S_T$. So, S_T will be the union of S'_T and the vertex sets $U \setminus U'$, for every component $G[U]$ of $G[Z]$. Hence, it remains to prove how to find the sets U' in polynomial time; we show this below.

Let $U \subseteq Z$ such that $G[U]$ is a connected component of $G[Z]$. Either $N_{G'}(U) \cap T = \emptyset$, or $N_{G'}(U) = \{y\}$ for some $y \in T$. First, consider the case where $N_{G'}(U) \cap T = \emptyset$. We find a set U' that is a maximum subset of U such that $G[U' \cup N_{G'}(U)]$ has no T-cycles. Clearly if $|U| = 1$, then we can set $U' = U$. If $|U'| \geq 3$, then, since U' is a clique, $U' \subseteq V \setminus T$. Thus, if $|U \setminus T| \geq 2$, then we set $U' = U \setminus T$. So it remains to consider when $|U| \geq 2$ but $|U \setminus T| \leq 1$. If there is some $u \in U$ that is anti-complete to $N_{G'}(U)$, then we can set U' to be any 2-element subset of U containing u. Otherwise $N_{G'}(U) = \{y\}$ and y is complete to U. In this case, for any $u \in U$, we set $U' = \{u\}$.

Now we may assume that $N_{G'}(U) = \{y\}$ and $y \in T$. Again, we find a set U' that is a maximum subset of U such that $G[U' \cup \{y\}]$ has no T-cycles. Partition U into $\{U_0, U_1\}$ where $u \in U_1$ if and only if u is a neighbour of y. Since $y \in V' \setminus S'_T$, observe that U' contains at most one vertex of U_1, otherwise $G[U' \cup \{y\}]$ has a T-cycle. Since U' is a clique, if $|U'| \geq 3$ then $U' \subseteq U \setminus T$. So if $|U_0 \setminus T| \geq 2$ and there is an element $u \in U_1 \setminus T$, then we can set $U' = \{u\} \cup (U_0 \setminus T)$. If $|U_0 \setminus T| \geq 2$ but $U_1 \setminus T = \emptyset$, then we can set $U' = U_0 \setminus T$. So we may now assume that $|U_0 \setminus T| \leq 1$. If $U_0 \neq \emptyset$ and $|U| \geq 2$, then we set U' to any 2-element subset of U containing some $u \in U_0$. Clearly if $|U| = 1$, then we can set $U' = U$. So it remains to consider when $U_0 = \emptyset$ and $|U_1| \geq 2$. In this case, we set $U' = \{u\}$ for an arbitrary $u \in U_1$. □

Before stating the main result of this section, let us recall the function a on non-negative integers defined by $a(s) := \max\{7, 4s - 2\}$ used in Lemma 3.

Theorem 7. *For every integer $s \geq 0$,* Subset Feedback Vertex Set *can be solved in polynomial time for $(sP_1 + P_3)$-free graphs.*

Proof. Let $G = (V, E)$ be an $(sP_1 + P_3)$-free graph for some $s \geq 0$, and let $T \subseteq V$. We describe a polynomial-time algorithm for the optimization version of the problem on input (G, T). Let $S_T \subseteq V$ such that S_T is a minimum T-feedback vertex set of G, and let $F_T = V \setminus S_T$, so $G[F_T]$ is a maximum T-forest. Note that $G[F_T \cap T]$ is a forest. We consider three cases: either

1. $G[F_T \cap T]$ has at least $2s$ connected components;
2. $G[F_T \cap T]$ has fewer than $2s$ connected components, and each of these connected components consists of at most $a(s)$ vertices; or
3. $G[F_T \cap T]$ has fewer than $2s$ connected components, one of which consists of more than $a(s)$ vertices.

We describe polynomial-time subroutines that find a set F_T such that $G[F_T]$ is a maximum T-forest in each of these three cases, giving a minimum solution $S_T = V \setminus F_T$ in each case. We obtain an optimal solution by running each of these subroutines in turn: of the (at most) three potential solutions, we output the one with minimum size.

Case 1: *$G[F_T \cap T]$ has at least $2s$ connected components.*

We begin by proving a sequence of claims that describe properties of a maximum T-forest F_T, when in Case 1. Since G is $(sP_1 + P_3)$-free, $F_T \cap T$ induces a P_3-free forest, so $G[F_T \cap T]$ is a disjoint union of graphs isomorphic to P_1 or P_2. Let $A \subseteq F_T \cap T$ such that $G[A]$ consists of precisely $2s$ connected components. Note that $|A| \leq 4s$. We also let $Y = N(A) \cap F_T$, and partition Y into $\{Y_1, Y_2\}$ where $y \in Y_1$ if y has only one neighbour in A, whereas $y \in Y_2$ if y has at least two neighbours in A.

Claim 1: $|Y_2| \leq 1$.

Let $v \in Y_2$. Then v has neighbours in at least $s+1$ of the connected components of $G[A]$, otherwise $G[A \cup \{v\}]$ contains an induced $sP_1 + P_3$. Note also that v has at most one neighbour in each connected component of $G[A]$, otherwise $G[F_T]$ has a T-cycle. Now suppose that Y_2 contains distinct vertices v_1 and v_2. Then, of the $2s$ connected components of $G[A]$, the vertices v_1 and v_2 each have some neighbour in $s+1$ of these connected components. So there are at least two connected components of $G[A]$ containing both a vertex adjacent to v_1, and a vertex adjacent to v_2. Let A' and A'' be the vertex sets of two such connected components. Then $A' \cup A'' \cup \{v_1, v_2\} \subseteq F_T$, but $G[A' \cup A'' \cup \{v_1, v_2\}]$ has a T-cycle; a contradiction. This proves Claim 1.

Claim 2: $|Y| \leq 2s + 1$.

By Claim 1, it suffices to prove that $|Y_1| \leq 2s$. We argue that each connected component of $G[A]$ has at most one neighbour in Y_1, implying that $|Y_1| \leq 2s$. Indeed, suppose that there is a connected component C_A of $G[A]$ having two neighbours in Y_1, say u_1 and u_2. Then $G[V(C_A) \cup \{u_1, u_2\}]$ contains an induced P_3 that is anti-complete to $A \setminus V(C_A)$, contradicting that G is $(sP_1 + P_3)$-free. This proves Claim 2.

Claim 3: Y_1 is independent, and no connected component of $G[A]$ of size 2 has a neighbour in Y_1.

Suppose that there are adjacent vertices u_1 and u_2 in Y_1. Let a_i be the unique neighbour of u_i in A for $i \in \{1, 2\}$. Note that $a_1 \neq a_2$, for otherwise $G[F_T]$ has a T-cycle. Then $\{a_1, u_1, u_2\}$ induces a P_3, so $G[\{u_1, u_2\} \cup A]$ contains an induced $sP_1 + P_3$, which is a contradiction. We deduce that Y_1 is independent.

Now let $\{a_1, a_2\} \subseteq A$ such that $G[\{a_1, a_2\}]$ is a connected component of $G[A]$, and suppose that $u_1 \in Y_1$ is adjacent to a_1. Then a_1 is the unique neighbour of u_1 in A, so $G[\{u_1, a_1, a_2\}] \cong P_3$. Thus $G[\{u_1\} \cup A]$ contains an induced $sP_1 + P_3$, which is a contradiction. This proves Claim 3.

Claim 4: Let $Z = V \setminus N[A]$. Then $N(Z) \cap F_T \subseteq Y_2$.

Suppose that there exists $y \in Y_1$ that is adjacent to a vertex $c \in Z$. Let a be the unique neighbour of y in A. Then $G[\{c, y\} \cup A]$ contains an induced $sP_1 + P_3$, which is a contradiction. So Y_1 is anti-complete to Z. Now, if $c \in Z$ is adjacent to a vertex in $N[A] \cap F_T$, then c is adjacent to y_2 where $Y_2 = \{y_2\}$. This proves Claim 4.

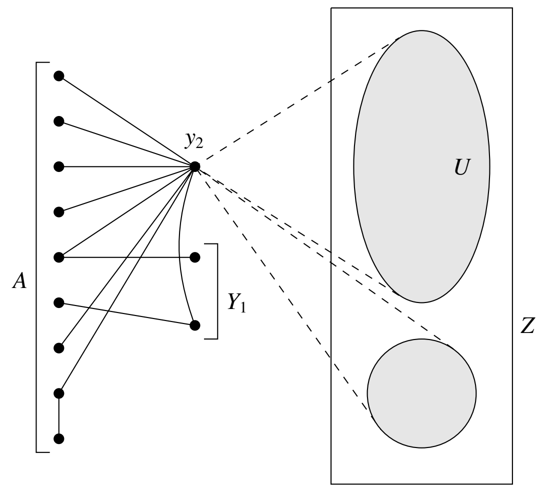


Fig. 11: An example of the structure obtained in Case 1 when $Y_2 = \{y_2\}$.

We now describe the subroutine that finds an optimal solution in Case 1. In this case, for any maximum forest F_T, there exists some set $A \subseteq T$ of size at most $4s$ such that $A \subseteq F_T$, and $G[A]$ consists of exactly $2s$ connected components, each isomorphic to either P_1 or P_2. Since $G[A]$ consists of components of $G[F_T \cap T]$, there is such an A for which $N(A) \cap T \subseteq S_T$. Thus we guess a set $A' \subseteq T$ in $O(n^{4s})$ time, discarding those sets that do not induce a forest with exactly $2s$ connected components, and those that induce a connected component consisting of more than two vertices.

For any such F_T and A', the set $N(A') \cap F_T$ has size at most $2s+1$, by Claim 2. Thus, in $O(n^{2s+1})$ time, we guess $Y' \subseteq N(A')$ with $|Y'| \leq 2s + 1$, and assume that $Y' \subseteq F_T$ whereas $N(A') \setminus Y' \subseteq S_T$. Let Y'_2 be the subset of Y' that contains vertices that have at least two neighbours in A'. We discard any sets Y' that do not satisfy Claims 1 or 3, or those sets for which $G[A' \cup Y']$ has a T-cycle on three vertices, one of which is the unique vertex of Y'_2.

Let $Z = V \setminus N[A']$ (for example, see Figure 11). Since $G[A']$ contains an induced sP_1, the subgraph $G[Z]$ is P_3-free. Now $N(Z) \cap F_T \subseteq Y'_2$ by Claim 4, where $|Y'_2| \leq 1$ by Claim 1. Thus, by Lemma 15, we can extend a partial solution $S'_T = N[A'] \setminus (A' \cup Y')$ of $G[N[A']]$ to a solution S_T of G, in polynomial-time.

Case 2: $G[F_T \cap T]$ *has at most* $2s - 1$ *connected components, each of size at most* $a(s)$*.*

We guess sets $F \subseteq T$ and $S \subseteq V \setminus T$ such that $F_T \cap T = F$ and $S_T \setminus T = S$. Since F has size at most $(2s-1)a(s) = (2s-1)\max\{7, 4s-2\}$ vertices, there are $O(n^{\max\{14s-7, 8s^2-8s+2\}})$

possibilities for F. By Lemma 11, we may assume that $|S_T \setminus T| \leq |F|$. So for each guessed F, there are at most $O(n^{\max\{14s-7, 8s^2-8s+2\}})$ possibilities for S. For each S and F, we set $S_T = (T \setminus F) \cup S$ and check, in $O(n+m)$-time by Lemma 12, if $G - S_T$ is a T-forest. In this way we exhaustively find all solutions satisfying Case 2, in $O(n^{\max^2\{14s-7, 8s^2-8s+2\}})$ time; we output the one of minimum size.

Case 3: $G[F_T \cap T]$ *has at most* $2s-1$ *connected components, one of which has size more than* $a(s)$*.*

By Lemma 3, there is some subset $B_T \subseteq F_T \cap T$ such that $|B| > a(s)$, and $G[B]$ is a connected component of $G[F_T \cap T]$ that is a tree satisfying Lemma 3(ii), as illustrated in Figure 5. In particular, there is a unique vertex $r \in B$ such that r has degree more than 2 in $G[B]$. Moreover, $G[F_T]$ has a connected component $G[D]$ that contains B, where $G[D]$ is a tree that also satisfies Lemma 3(ii). Note that there are at most $s-1$ vertices in $N_{G[B]}(r)$ having a neighbour in $V \setminus T$.

We guess a set $B' \subseteq T$ such that $|B'| = a(s) + 1 = \max\{8, 4s-1\}$. We also guess a set $L' \subseteq V \setminus T$ such that $|L'| \leq s-1$. Let $D' = B' \cup L'$. We check that $G[D']$ has the following properties:

- $G[D']$ is a tree,
- $G[D']$ has a unique vertex r' of degree more than 2, with $r' \in B'$,
- $G[D']$ has at most $s-1$ vertices with distance 2 from r', and each of these vertices has degree 1, and
- each vertex $v \in L'$ has degree 1 in $G[D']$, and distance 2 from r'.

We assume that D' induces a subtree of the large connected component $G[D]$, where $r = r'$, and D' contains r, all neighbours of r with degree 2 in $G[D]$, and all vertices at distance 2 from r. In other words, $G[D']$ can be obtained from $G[D]$ by deleting some subset of the leaves of $G[D]$ that are adjacent to r. In particular, $D' \subseteq F_T$. We also assume that L' is the set of all vertices of $V(D) \setminus T$ that have distance 2 from r.

It follows from these assumptions that $N(D' \setminus \{r\}) \setminus \{r\} \subseteq S_T$. Let $Z = V \setminus N[D' \setminus \{r\}]$, and observe that each $z \in Z$ has at most one neighbour in D' (if it has such a neighbour, this neighbour is r). So $N(Z) \cap F_T \subseteq \{r\}$.

In order to apply 15, it remains to show that $G[Z]$ is P_3-free. Let $B_1 = B' \cap N(r)$. As r has at least $3s-1$ neighbours in $G[B']$, by Lemma 3, $G[B_1]$ contains an induced sP_1. Moreover, $N(B_1) \cap F_T \subseteq D'$. Since G is $(sP_1 + P_3)$-free, $G[Z]$ is P_3-free. Thus, by Lemma 15, we can extend a partial solution $S'_T = N(D' \setminus \{r\}) \setminus \{r\}$ of $G[N[D' \setminus \{r\}]]$ to a solution S_T of G, in polynomial time. □

We are now ready to prove the following result.

Theorem 4 (restated). *Let H be a graph with $H \neq sP_1 + P_4$ for all $s \geq 1$. Then* Subset Feedback Vertex Set *on H-free graphs is polynomial-time solvable if $H = P_4$ or $H \subseteq_i sP_1 + P_3$ for some $s \geq 1$ and* NP*-complete otherwise.*

Proof. If H has a cycle or a claw, we use Theorem 2. The cases $H = P_4$ and $H = 2P_2$ follow from the corresponding results for permutation graphs [87] and split graphs [45]. The remaining case $H \subseteq_i sP_1 + P_3$ follows from Theorem 7. ◻

3.5 Subset Odd Cycle Transversal

We start this Section by proving that Odd Cycle Transversal is NP-complete on $(P_2 + P_5, P_6)$-free graphs. We do this by modifying the construction used in [85] for proving that this problem is NP-complete on P_{13}-free segment graphs.

Theorem 8. Odd Cycle Transversal *is* NP*-complete on $(P_2 + P_5, P_6)$-free graphs.*

Proof. To prove NP-hardness we reduce from Vertex Cover (recall this problem is NP-complete, see e.g. [49]). Let (G, k) be an instance of Vertex Cover . Let n and m be the number of vertices and edges, respectively, in G. Let $v_1, \ldots, v_n$ be the vertices of G. We construct a graph G^* from G as follows.

1. For $i \in \{1, \ldots, n\}$ create vertices a_i, b_i, c_i, x_i and y_i. Let A, B, C, X and Y be the sets of, respectively, a_i, b_i, c_i, x_i and y_i vertices.
2. For $i, j \in \{1, \ldots, n\}$, add the edges $x_i y_j$ and $b_i y_j$ (so we make Y complete to both X and B).
3. For each $i \in \{1, \ldots, n\}$, add edges $x_i a_i, x_i b_i, a_i b_i, b_i c_i, c_i y_i$ (a *vertex gadget*, see also Figure 12(a) and note that b_i is adjacent to y_i by the previous step).
4. For each edge $v_i v_j$ in G with $i < j$, add a vertex $d_{i,j}$ adjacent to both x_i and y_j (an *edge gadget*, see also Figure 12(b)). Let D be the set of $d_{i,j}$ vertices.

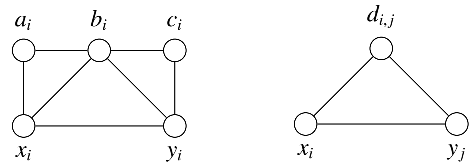
We first claim that the following statements are equivalent:

(i) G has a vertex cover of size at most k;
(ii) G^* has an odd cycle transversal of size at most $n + k$;

Below we prove (i) $\Rightarrow$ (ii) and (ii) $\Rightarrow$ (i).

(i) $\Rightarrow$ (ii). Suppose that G has a vertex cover Q of size at most k. We define the set

$$S = \bigcup_{v_i \in Q} \{x_i, y_i\} \cup \bigcup_{v_i \notin Q} \{b_i\}$$



(a) Vertex gadget (b) Edge gadget

Fig. 12: The two gadgets used in the proof of Theorem 8.

and observe that $|S| = 2|Q| + (n - |Q|) = n + |Q| \leq n + k$. We claim that S is an odd cycle transversal of G^*. This can be seen as follows. The only induced odd cycles in G^* are the three triangles in each vertex gadget and the triangle in each edge gadget. By construction of S, for every $i \in \{1, \ldots, n\}$, either S contains both x_i and y_i or S contains b_i, thus every triangle in every vertex gadget intersects S. Furthermore, since Q is a vertex cover of G, for every edge gadget $\{x_i, y_j, d_{i,j}\}$, either $x_i \in S$ or $y_j \in S$. Therefore S intersects every odd cycle in G^*.

(ii) $\Rightarrow$ (i). Suppose that G^* has an odd cycle transversal S of size at most $n + k$. Consider an edge gadget on $\{x_i, y_j, d_{i,j}\}$. If $d_{i,j} \in S$ then $S' := (S \setminus \{d_{i,j}\}) \cup \{x_i\}$ is an odd cycle transversal of G with $|S'| \leq |S|$. We may therefore assume that S contains no vertices of D. For $i \in \{1, \ldots, n\}$, the vertex b_i intersects all odd cycles in the vertex gadget on $\{a_i, b_i, c_i, x_i, y_i\}$. If $b_i \notin S$ then $|S \cap \{a_i, b_i, c_i, x_i, y_i\}| \geq 2$ since S intersects all induced odd cycles of the vertex gadget. Note that $\{x_i, y_i\}$ intersects all odd cycles of the vertex gadget. Therefore, if $|S \cap \{a_i, b_i, c_i, x_i, y_i\}| \geq 2$, then $S' := (S \setminus \{a_i, b_i, c_i\}) \cup \{x_i, y_i\}$ is an odd cycle transversal of G^* with $|S'| \leq |S|$. We may therefore assume that for every $i \in \{1, \ldots, n\}$, either $b_i \in S$ or $\{x_i, y_i\} \subseteq S$ and there are no other vertices in S. Let $B_S = B \cap S$, $X_S = S \cap X$ and $Y_S = S \cap Y$. Then $|S| = |B_S| + |X_S| + |Y_S| = n + |X_S|$. Let $Q = \bigcup_{x_i \in S} \{v_i\}$. Then $|Q| = |X_S| = |S| - n \leq n + k - n = k$.

We claim that Q is a vertex cover of G. This can be seen as follows. Consider an edge $v_i v_j$ of G (without loss of generality assume $i < j$). Then $|\{x_i, y_j, d_{i,j}\} \cap S| \geq 1$, as S is an odd cycle transversal of G^*. By assumption on S, $d_{i,j} \notin S$ and if $y_j \in S$ then $x_j \in S$. It follows that $x_i \in S$ or $x_j \in S$ and so $v_i \in Q$ or $v_j \in Q$. We conclude that Q is a vertex cover of G of size at most k.

It only remains to show that G^* is $(P_2 + P_5, P_6)$-free. Suppose, for contradiction, that $H \in \{P_2 + P_5, P_6\}$ is an induced subgraph of G^*. Every vertex in $A \cup C \cup D$ has degree 2 and its two neighbours are adjacent. Therefore no vertex in $V(H) \cap (A \cup C \cup D)$ is an internal vertex of a path of H. That is, if $x \in V(H) \cap (A \cup C \cup D)$ then x has degree 1 in H. Furthermore, $A \cup C \cup D$ is an independent set in G^*. Hence, if $H = P_2 + P_5$, then at most one vertex of the P_2 connected component of H can be in $A \cup C \cup D$. We conclude that $G^*[V(H) \cap (B \cup X \cup Y)]$ contains an induced subgraph H' on four vertices that is isomorphic to $P_1 + P_3$ if $H = P_2 + P_5$ or P_4 if $H = P_6$. Since Y is an independent set and $B \cup X$ is a perfect matching, H' must contain at least one vertex of $B \cup X$ and at least one vertex of Y. As Y is complete to $B \cup X$, we find that H' contains either C_4 or $K_{1,3}$ as a (not necessarily induced) subgraph, a contradiction. This completes the proof. □

The next result uses the same reduction of [87] which proved the analogous result for Subset Feedback Vertex Set.

Theorem 9. Subset Odd Cycle Transversal *is* NP*-complete for the class of split graphs (or equivalently,* $(C_4, C_5, 2P_2)$*-free graphs).*

Proof. We observe that the problem belongs to NP. To show NP-hardness, we reduce from Vertex Cover . Let a graph $G = (V, E)$ and a positive integer k be an instance of Vertex Cover . From G, we construct a graph G' as follows. Let $V(G') = V \cup E$. Add an edge between $e \in E$ and $v \in V$ in G' if and only if v is an end-vertex of e in G. Add edges so that V induces a clique of G'. Hence, G' is a split graph with independent set E and clique V. For example, when $G = P_4$, see Figure 13. Let $T = E$. We show that G has a vertex cover of size at most k if and only if G' has an odd T-cycle transversal of size at most k.

First suppose that G has a vertex cover S of size at most k. Then S is an odd T-cycle transversal of G'. Now suppose that G' has an odd T-cycle transversal S_T of size at most k. As every vertex of E in G' has degree 2, we can replace every vertex of E that belongs to S_T by one of its neighbours to obtain an odd T-cycle transversal of the same size as S_T. Hence we may assume, without loss of generality, that $S_T \cap E = \emptyset$. As a vertex of E and its two neighbours in V form a triangle, this means that S_T contains at least one neighbour of every $e \in E$. Hence, S_T is a vertex cover of G. □

Recall that Subset Feedback Vertex Set can be solved in polynomial time for P_4-free graphs (see e.g. [87,88]). Now we are ready to prove the same for Susbset Odd Cycle Transversal.

Theorem 10. Subset Odd Cycle Transversal *can be solved in polynomial-time for* P_4*-free graphs.*

Proof. Let G be a cograph with n vertices and m edges. First construct the modified cotree T'_G and then consider each node of T'_G starting at the leaves of T'_G and ending in

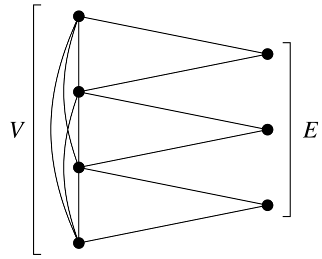


Fig. 13: The graph P'_4: an example of the construction in the proof of Theorem 9.

its root r. Let x be a node of T'_G. We let S_x denote a minimum odd $(T \cap V(G_x))$-cycle transversal of G_x.

If x is a leaf, then G_x is a 1-vertex graph. Hence, we can let $S_x = \emptyset$. Now suppose that x is a +-node. Let y and z be the two children of x. Then, as G_x is the disjoint union of G_y and G_z, we let $S_x = S_y \cup S_z$.

Finally suppose that x is a $\times$-node. Let y and z be the two children of x. Let $T_y = T \cap V(G_y)$ and $T_z = T \cap V(G_z)$. Let $B_x = V(G_x) \setminus S_x$. As G_x is the join of G_y and G_z we observe the following. If $B_x \cap V(G_y)$ contains two adjacent vertices, at least one of which belongs to T_x, then $B_x \cap V(G_z) = \emptyset$ (as otherwise $G[B_x]$ has a triangle containing a vertex of T) and thus $V(G_z) \subseteq S_x$. In this case we may assume that $S_x = S_y \cup V(G_z)$. Similarly, if $B_x \cap V(G_z)$ contains two adjacent vertices, at least one of which belongs to T_z, then $B_x \cap V(G_y) = \emptyset$ and thus $V(G_y) \subseteq S_x$. In this case we may assume that $S_x = S_z \cup V(G_y)$. From the two sets $S_y \cup V(G_z)$ and $S_z \cup V(G_y)$ we remember the smallest one.

It remains to examine the case where $B_x \cap V(G_y)$ and $B_x \cap V(G_z)$ induce subgraphs of G in which the vertices of $T_y \cap B_x$ and $T_z \cap B_x$, respectively, are singleton components.

First suppose that $T_y \cap B_x$ and $T_z \cap B_x$ are both non-empty. Then $B_x \cap V(G_y)$ and $B_x \cap V(G_z)$ are both independent sets, as otherwise $G[B_x]$ would contain a T-triangle. We examine this situation by computing a largest independent set I_y in G_y and a largest independent set I_z in G_z; it is well-known that this can be done in polynomial time (for example, it follows from Lemma 13). We remember $V(G_x) \setminus (I_y \cup I_z)$.

Now suppose that $T_y \cap B_x$ is non-empty, but $T_z \cap B_x$ is empty. Then $B_x \cap V(G_z)$ must be an independent set, as otherwise we obtain a T-triangle by taking a vertex of $T_y \cap B_x$ and two adjacent vertices of $B_x \cap V(G_z)$. First assume that $B_x \cap V(G_z)$ has size at least 2. We observe that $(B_x \cap V(G_y)) \setminus T_y$ is also an independent set; otherwise two adjacent vertices of $(B_x \cap V(G_y)) \setminus T_y$, two vertices of $B_x \cap V(G_z)$ and one vertex of $T_y \cap B_x$ would form a T-cycle on five vertices. Hence, both $B_x \cap V(G_y)$ and $B_z \cap V(G_z)$ are independent sets, and we already dealt with this case above.

Now assume that $B_x \cap V(G_z)$ has size at most 1. In this case $B_x \cap V(G_y)$ is a minimum T_y-vertex cover of G_y. We can compute a minimum T_y-vertex cover S of G_y in polynomial time by Theorem 6. We remember $S \cup (V(G_z) \setminus \{z\})$ where z is an arbitrary vertex of $V(G_z) \setminus T_z$ if the latter set is non-empty; otherwise we just remember $S \cup (V(G_z)$.

We deal with the case where $T_z \cap B_x$ is non-empty, but $T_y \cap B_x$ is empty in the same way and remember the output. We also consider the possible situation where $T_z \cap B_x = T_y \cap B_x = \emptyset$, in which case we remember T. Finally, we take as set S_x a set of minimum size over the sets that we remembered.

Constructing T'_G takes $O(n + m)$ time by Lemma 2. As $T_{G'}$ has $O(n)$ nodes and processing a node takes $O(n + m)$ time (due to the application of Lemma 13), the total running time is $O(n^2 + mn)$. ◻

The following theorem is the main result of the section and it is our contribution to Theorem 5. Its proof uses the same approach as the proof of Theorem 7 but we need more advanced arguments.

Theorem 11. *For every integer $s \geq 0$,* Subset Odd Cycle Transversal *can be solved in polynomial-time for $(sP_1 + P_3)$-free graphs.*

Proof. Let $G = (V, E)$ be an $(sP_1 + P_3)$-free graph and let $T \subseteq V$. If $s = 0$, then we can apply Theorem 10, so we may assume that $s \geq 1$. We describe a polynomial-time algorithm to solve the optimization problem on input (G, T). That is, we describe how to find a smallest odd T-cycle transversal. In fact, we will solve the equivalent problem of finding a maximum size T-bipartite subgraph B_T of G which is, of course, the complement of a smallest odd T-cycle transversal, that is $S_T = V \setminus B_T$. We separate into two cases that separately seek to find T-bipartite subgraphs with complementary constraints on the size of the intersection of this subgraph with T. The largest one found overall is the desired output.

Case 1: *Compute a largest T-bipartite subgraph B_T of G such that $|B_T \cap T| \leq \max\{3, 4s - 3\}$.*

Note that $B^* = V \setminus T$ is a candidate solution. We must see if we can find something larger. Consider each set $B' \subseteq T$ of size at most $\max\{3, 4s - 3\}$, discarding any set that does not induce a bipartite graph. There are $O(n^{\max\{3,4s-3\}})$ possible sets. For each choice of B', consider all sets $S \subseteq V \setminus T$ of size less than $|B'|$. Then $B' \cup (V \setminus T) \setminus S$ is a candidate solution if it induces a T-bipartite subgraph, which is checked in $O(n + m)$-time by

Lemma 12. For each B', there are $O(n^{\max\{3,4s-3\}})$ possible choices of S to consider. Note that we do not need to examine larger S since then $B' \cup (V \setminus T) \setminus S$ is no larger than B^*.

Case 2: *Compute a largest T-bipartite subgraph B_T of G such that $|B_T \cap T| \geq \max\{4, 4s-2\}$.*

Note that B_T might not exist in which case the output of Case 1 is our result. We make some observations regarding the subgraph B_T that we seek. As $G[B \cap T]$ is a bipartite graph on at least $\max\{4, 4s-2\}$ vertices, it contains an independent set A of size $\max\{2, 2s-1\}$. Let $Y = B_T \cap N(A)$ and consider a partition $\{Y_1, Y_2\}$ of Y where y is in Y_1 if y has precisely one neighbour in A, and otherwise y is in Y_2. Let $Z = V \setminus N[A]$.

Claim 1: Y_1 is an independent set, no two vertices of Y_1 have a common neighbour in A and $|Y_1| \leq |A|$.
Suppose that there are adjacent vertices $y, y' \in Y_1$, and let a be the unique neighbour of y in A. Then, according to whether or not y' is adjacent to a, either $\{y, y', a\}$ induces an odd T-cycle, or $G[A \cup \{y, y'\}]$ contains an induced $sP_1 + P_3$; both are contradictions. If there are vertices $y, y' \in Y_1$ that have the same neighbour a in A, then, again, $G[A \cup \{y, y'\}]$ contains an induced $sP_1 + P_3$, a contradiction. It follows that $|Y_1| \leq |A|$. This proves Claim 1.

Claim 2: Y_2 is an independent set, each $y \in Y_2$ has at least s neighbours in A and any two vertices of Y_2 share at least one neighbour in A.
Let y and y' be distinct vertices in Y_2. Since $G[A \cup \{y\}]$ is $(sP_1 + P_3)$-free, y is non-adjacent to at most $s-1$ vertices of A. So y has at least $2s - 1 - (s-1) = s$ neighbours in A. Similarly, y' is non-adjacent to at most $s-1$ vertices of A, so y and y' have a neighbour of A in common, a say. If y and y' are adjacent, then $\{y, y', a\}$ induces an odd T-cycle; a contradiction. This proves Claim 2.

Claim 3: $N(Z) \cap B_T \subseteq Y_2$.
By definition, $N(Z) \cap B_T \subseteq Y$. Suppose that $z \in Z$ is adjacent to a vertex $y \in Y_1$. Let a be the unique neighbour of y in A. Since $|A| = \max\{2, 2s-1\} \geq s+1$ for all $s \geq 0$, it follows that $G[\{z, y\} \cup A]$ contains an induced $sP_1 + P_3$, a contradiction. So Y_1 is anti-complete to Z, and the claim follows. This proves Claim 3.

Armed with these definitions and claims we consider how to find B_T. The basic idea is to consider all possible choices of A and Y. We have two subcases.

Case 2a: *Compute a largest T-bipartite subgraph B_T of G such that $|B_T \cap T| \geq \max\{4, 4s-2\}$ and, for some choice of A, we have $|Y| < \max\{s+3, 3s\}$.*

Consider each set $A \subseteq T$ of size $\max\{2, 2s-1\}$ such that A is an independent set. There are $O(n^{\max\{2,2s-1\}})$ choices. For each A, we consider each set $Y_1 \subseteq N(A)$ of vertices that each has a single neighbour in A such that Y_1 satisfies Claim 1. As we require that $|Y_1| \leq |A|$, there are again $O(n^{\max\{2,2s-1\}})$ choices. Then consider each set $Y \subseteq N(A)$ of size at most $\max\{s+3, 3s\}$ such that $Y_1 \subseteq Y$ and $Y_2 = Y \setminus Y_1$ is a set of vertices that each has at least two neighbours in A and satisfies Claim 2. We also require that $A \cup Y$ does not contain any odd T-cycles, which is checked in $O(n+m)$-time by Lemma 12. There are $O(n^{\max\{s+3,3s\}})$ choices for Y.

Note that $G[A \cup Y]$ is bipartite since $G[Y]$ can contain only even cycles as Y_1 and Y_2 are independent sets, and any odd cycle is an odd T-cycle, since $A \subseteq T$, which we have proscribed. By Claim 2, vertices of Y_2 all belong to the same connected component of $G[A \cup Y]$ and, as, by definition and Claim 1, each vertex in $G[A \cup Y_1]$ has degree at most 1, we deduce that every vertex of degree at least 2 in $G[A \cup Y]$ belongs to the same connected component. We denote this connected component by $G[D]$, or we let D be the empty graph if there is no such connected component (which only occurs when $Y_2 = \emptyset$). See Figure 14 for an illustration.

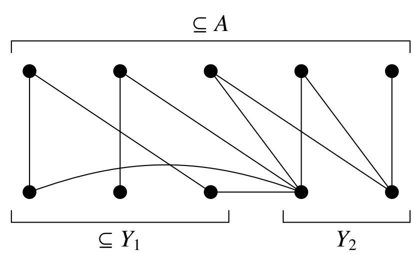


Fig. 14: An example of a connected component D of $G[A \cup Y]$.

Recall that $Z = V \setminus N[A]$. Since A contains an induced sP_1 subgraph, $G[Z]$ is P_3-free, and so is a disjoint union of complete graphs. For a connected component $G[U]$ of $G[Z]$, let U^+ contain each vertex u of U such that $G[A \cup Y \cup \{u\}]$ does not contain an odd T-cycle through u, which is checked in $O(n+m)$-time by Lemma 12.

The aim in the remainder of this subcase is to find the largest possible T-bipartite subgraph B_T that contains $A \cup Y$ and a subset of Z. Clearly for each connected component $G[U]$ in $G[Z]$, any vertex that might be in B_T must belong to U^+. We shall see later that we can consider each connected component of $G[Z]$ independently and that it suffices to find for each the maximum size subset of U^+ that can be added to B_T. We first investigate the possible edges between U^+ and D. Note that by Claim 3, the neighbours of U^+ in D belong to Y_2.

Claim 4: Either $|N(U^+ \cap B_T) \cap V(D)| \leq 1$ or $|N(D) \cap (U^+ \cap B_T)| \leq 1$.
We can assume that there are two vertices u_1, u_2 of $U^+ \cap B_T$ that each have a neighbour in D else the claim follows immediately. Moreover we can assume that these neighbours, say y_1 and y_2 respectively, are distinct. By Claim 2, y_1 and y_2 have a common neighbour a in A. Thus we have a path $u_1y_1ay_2u_2$. As U^+ is a clique, this can be extended to a cycle by the edge u_1u_2, but, as $A \subseteq T$, this is an odd T-cycle, a contradiction. This proves Claim 4.

Claim 5: For each component $G[U]$ of $G[Z]$, let U^{++} be a subset of U^+; and let Z^{++} be the union of each U^{++} over all components $G[U]$ of $G[Z]$. If $G[A \cup Y \cup Z^{++}]$ contains an odd T-cycle, then $G[A \cup Y \cup U^{++}]$ contains an odd T-cycle for some component $G[U]$ of $G[Z]$.
Suppose that C is an odd T-cycle of $G[A \cup Y \cup Z^{++}]$. First we show that C contains two vertices of some U^{++}. Towards a contradiction, suppose C is a subgraph of $G[A \cup Y \cup Z^*]$, where Z^* is a subset of Z^{++} with at most one vertex from each component of $G[Z]$. Recall that D is a bipartite graph that (if non-empty) is a component of $G[A \cup Y]$. By Claim 3, all neighbours of Z^* are contained in Y_2, which, in turn, is contained in one side of the bipartition of D. Hence $G[A \cup Y \cup Z^*]$ has no odd T-cycles and, in particular, C is not an odd T-cycle. From this contradiction we deduce that there is some component $G[U]$ of $G[Z]$ such that C contains two vertices of U^{++}. Let u_1 and u_2 be distinct vertices of $V(C) \cap U^{++}$. If C is not contained in $G[A \cup Y \cup U^{++}]$, then there are distinct vertices $y_1 \in N_C(u_1) \cap Y_2$ and $y_2 \in N_C(u_2) \cap Y_2$. But, by Claim 2, y_1 and y_2 have a common neighbour $a \in A$, so $u_1y_1ay_2u_2u_1$ is an odd T-cycle contained in $G[A \cup Y \cup U^{++}]$. This proves Claim 5.

Let Z^+ be the union of U^+ over all connected components U of $G[Z]$. Suppose that C is an odd T-cycle of $G[A \cup Y \cup Z^+]$. We show that C contains two vertices of some set U^+. Assume that C is a subgraph of $G[A \cup Y \cup Z^*]$, where Z^* is a subset of Z^+ with at most one vertex from each connected component. But this is a contradiction as $G[A \cup Y \cup Z^*]$ is bipartite: $G[A \cup Y]$ is bipartite and the vertices of Z^* are adjacent to Y_2 whose vertices are separated by paths of length 2. Thus to extend $A \cup Y$ to the largest possible T-bipartite graph, for each connected component U of $G[Z]$, we must find U^{++}, a maximum subset of U^+ such that $G[A \cup Y \cup U^{++}]$ has no odd T-cycle. By the preceding argument and Claim 4, we can consider each connected component separately.

We describe how to find such a set U^{++}. We first suppose that for the set we seek $|U^{++}| \geq 3$. Partition U^+ into $\{U_0^+, U_1^+, U_2^+\}$ where $u \in U_0^+$ if $u \in U^+$ has no neighbours in

$V(D)$, $u \in U_1^+$ if u has exactly one neighbour in $V(D)$, and otherwise $u \in U_2^+$. By Claim 4, and since we are assuming that $|U^{++}| \geq 3$, we have $|U_2^+ \cap U^{++}| \leq 1$. And $U_2^+ \cap U^{++} = \{u_2\}$ if this set is not empty. Let $N(U_1^+) \cap V(D) = \{d_1, \ldots, d_m\}$, for some $m \geq 1$, if U_1^+ is not empty. We partition U_1^+ into classes $\{Q_1, \ldots, Q_m\}$ such that $u \in Q_i$ if $N(u) \cap V(D) = \{d_i\}$. Using Claim 4 again, we have that $U^{++} \cap U_1^+ \subseteq Q_i$ for some $i \in \{1, \ldots, m\}$. So we choose the i with $d_i \notin T$ that maximises $|Q_i \setminus T|$, and set $U^{++} = (U_0^+ \cup Q_i) \setminus T$. If $d_i \in T$ for all $i \in \{1, \ldots, m\}$ but $U_1^+ \setminus T \neq \emptyset$, then $U^{++} = (U_0^+ \setminus T) \cup \{u\}$ for an arbitrarily chosen $u \in U_1^+ \setminus T$. Otherwise $U^{++} = (U_0^+ \cup U_2^+) \setminus T$. This process finds a maximum U^{++} of size at least 3 if such a set exists.

Now consider the case where $|U^{++}| \leq 2$. Recall that no vertex of U^+ creates an odd T-cycle with vertices of $A \cup Y$. So any odd T-cycle of $G[A \cup Y \cup \{u_1, u_2\}]$ contains $\{u_1, u_2\}$. We require one more claim to handle this case, which shows that we may also consider each of these remaining connected components independently.

Claim 5: If C is an odd T-cycle of $G[A \cup Y \cup Z]$ with $|C \cap U| \leq 2$ for each connected component U of $G[Z]$, then there is a connected component $U*$ and an odd T-cycle C' of $G[A \cup Y \cup Z]$ such that $C' \cap Z = C \cap U*$.

Let C be such an odd T-cycle of $G[A \cup Y \cup Z]$. Since $N(Z) \cap (A \cup Y) \subseteq Y_2$, by Claim 3, and the vertices of Y_2 are contained in one part of the bipartition of D, C must contain at least one edge u_1u_2 with u_1, u_2 in some connected component $U*$ of $G[Z]$. By assumption and Claim 3, C contains the path yu_1u_2y' for some $y, y' \in Y_2$. Then there is some $a \in N(y) \cap N(y') \cap A$, by Claim 2, and $C' = \{a, y, u_1, u_2, y', a\}$ is an odd T-cycle. This proves Claim 5.

Now, to extend $A \cup Y$ to the largest possible T-bipartite graph, for each component $G[U]$ of $G[Z]$, we must find a maximum subset U^{++} of U^+ such that $G[A \cup Y \cup U^{++}]$ has no odd T-cycle. By the contrapositive of Claim 5, if $G[A \cup Y \cup U^{++}]$ does not contain an odd T-cycle for each component $G[U]$ of $G[Z]$, then $G[A \cup Y \cup Z^{++}]$ does not contain an odd T-cycle.

We describe how to find such a set U^{++} in polynomial time. We first suppose that for the set we seek $|U^{++}| \geq 3$. Note that in this case we have $U^{++} \cap T = \emptyset$, since U^{++} is a clique. Partition $U^+ \setminus T$ into $\{U_0^+, U_1^+, U_2^+\}$ where $u \in U_0^+$ if $u \in U^+ \setminus T$ has no neighbours in $V(D)$, $u \in U_1^+$ if u has exactly one neighbour in $V(D)$, and otherwise $u \in U_2^+$. If U_1^+ is not empty, then let $N(U_1^+) \cap V(D) = \{d_1, \ldots, d_m\}$, for some $m \geq 1$. We partition U_1^+ into classes $\{Q_1, \ldots, Q_m\}$ such that $u \in Q_i$ if $N(u) \cap V(D) = \{d_i\}$. Using Claim 4, either $U^{++} \cap U_1^+ = \emptyset$ or $U^{++} \cap U_2^+ = \emptyset$. Moreover, when $U^{++} \cap U_1^+ \neq \emptyset$, then $U^{++} \cap U_1^+ \subseteq Q_i$ for some $i \in \{1, \ldots, m\}$; and when $U^{++} \cap U_2^+ \neq \emptyset$, then $|U^{++} \cap U_2^+| = 1$. So, if there exists some $d_i \notin T$, then we choose such an i that maximises $|Q_i|$, and set $U^{++} = U_0^+ \cup Q_i$.

If $U_1^+ \neq \emptyset$ but $d_i \in T$ for all $i \in \{1, \ldots, m\}$, then set $U^{++} = U_0^+ \cup \{u\}$ for an arbitrarily chosen $u \in U_1^+$. Now suppose U_1^+ is empty, and recall that in this case $|U_2^+ \cap U^{++}| \leq 1$. If U_2^+ is non-empty, then set $U^{++} = U_0^+ \cup \{u_2\}$ for some $u_2 \in U_2^+$. Finally, if U_2^+ is also empty, then set $U^{++} = U_0^+$. This process finds a maximum U^{++} of size at least 3 if such a set exists.

Now consider the case where $|U^{++}| \leq 2$. We exhaustively check all pairs of vertices in U^+, of which there are $O(n^2)$. Let u_1, u_2 be such a pair of distinct vertices. By Claim 5, we need only check that $G[A \cup Y \cup \{u_1, u_2\}]$ is T-bipartite; if it is, then we set $U^{++} = \{u_1, u_2\}$. Recall that this check runs in polynomial time, by Lemma 12. Finally, if no pair is found, we set U^{++} to be the singleton set consisting of any arbitrarily chosen vertex of U^+.

Case 2b: *Compute a largest T-bipartite subgraph B_T of G such that $|B_T \cap T| \geq \max\{4, 4s-2\}$ and, for some choice of A, we have $|Y| \geq \max\{s+4, 3s+1\}$.*

Note that as A has size $\max\{2, 2s-1\}$ and $|Y_1| \leq |A|$, we have that $|Y_2| \geq s+2$. So suppose that Y_2' is a subset of Y with $|Y_2'| = s+2$. Let $A_0 = N(Y_2') \cap A$, and let $Y_0 = N(A_0) \cap B_T$. Observe that $s \leq |A_0| \leq \max\{2, 2s-1\}$ and $Y_2' \subseteq Y_0 \subseteq Y$. Finally let $Y_0' = N(A_0)$ and note that $Y_2' \subseteq Y_0 \subseteq Y_0'$.

Claim 6: Let $y \in Y_2'$ and $y' \in Y_0$ be distinct vertices. Then there is an even T-path in $G[A_0 \cup Y_2' \cup \{y'\}]$ between y and y'.
Assume that y and y' have no common neighbour in A_0 else the claim is immediate. First let us assume at least one vertex between y and y' is contained in Y_2'. Without loss of generality, $y \in Y_2'$. By Claim 2 and the definitions of A_0 and Y_0', we can assume that $y' \in Y_0' \setminus Y_2'$ and that y' has a neighbour a' in A_0, and, moreover, that a' is the neighbour of some vertex $y'' \in Y_2' \setminus \{y\}$. Again by Claim 2, y and y'' share a common neighbour $a'' \in A_0$. Thus $ya''y''a'y'$ is an even T-path in $G[A_0 \cup Y_2' \cup \{y'\}]$.

Now we consider the case where $y, y' \in Y_0' \setminus Y_2'$. Let y^* be a vertex of Y_2'. By the previous case there is an even T-path in $G[A_0 \cup Y_2' \cup \{y, y^*\}]$ between y and y^* and an even T-path in $G[A_0 \cup Y_2' \cup \{y', y^*\}]$ between y' and y^*. By joining these two even T-paths we obtain an even T-path in $G[A_0 \cup Y_2' \cup \{y, y', y^*\}] = G[A_0 \cup Y_2' \cup \{y, y'\}]$ between y and y'. This proves Claim 6.

Recall that $A_0 \subseteq A$ and A is an independent set. Hence, $G[A_0]$ has an induced sP_1.

Claim 7: $N(A_0) \cap N(Y_2') \cap B_T = \emptyset$.
Assume there is a vertex $v \in N(A_0) \cap N(Y_2')$ else the claim is immediate. By assumption there are vertices $a \in A_0 \cap N(v)$ and $y \in Y_2' \cap N(v)$. By definition of A_0, there is a vertex $y' \in Y_2' \cap N(a)$. If $y' = y$ then $v \notin B_T$ or $\{a, v, y\}$ induces an odd T-cycle in B_T. Suppose now that $y' \neq y$, then by Claim 6 there is an even T-path in $G[A_0 \cup Y_2' \cup \{y, y'\}]$ between

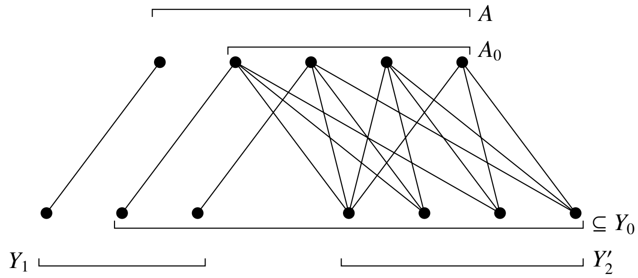


Fig. 15: An example of $G[A \cup Y_1 \cup Y_2']$ when $s = 3$.

y and y'. Then $v \notin B_T$ or the cycle obtained by linking this even T-path between y and y' with the path $yvay'$ would be an odd T-cycle of B_T. This proves Claim 7.

Claim 8: If $u \in U_2$, then u has at least two neighbours in $Y_2' \subseteq Y_2$.
Suppose that $u \in U_2$. Since $Y_0'| \geq |Y_2'| = s + 2$, the graph $G[Y_0']$ contains an induced sP_1 subgraph by Claim 2. Consider when $s \geq 1$. First we show that no vertex of $Y_0' \cup N(u)$ is adjacent to Y_2', that is $Y_0' \cap N(u) \cap N(Y_2') = \emptyset$. Assume, to reach a contradiction, that there is a vertex $y \in Y_0' \cap N(u) \cap N(Y_2')$. By Claim 2, $y \in Y_0' \setminus Y_2'$ and by definition of Y_0', y has a neighbour in A_0: then $y \notin Y_0'$, which is a contradiction (recall Y_0' now contains no vertex from $N(A_0) \cap N(Y_2')$).

Let x and y be neighbours of u in Y_0' that are contained in distinct components of $G[Y_0']$. By what we have just proved, the set $Y_2' \cup \{x, y\}$ is independent. As $G[Y_2' \cup \{u, x, y\}]$ is $(sP_1 + P_3)$-free, u is non-adjacent to at most $s-1$ of the vertices in Y_2'. Since $|Y_2'| = s+2$, the claim holds. The case where $s = 0$ follows, in a similar manner, since $|Y_2| \geq 2$. This proves Claim 8.

Claim 9: Either $U_0 = \emptyset$ or $U_1 = \emptyset$. Moreover, $|N(U_1) \cap Y_0'| = 1$ if $U_1 \neq \emptyset$.
Suppose that U_0 and U_1 are both non-empty. Let $u_0 \in U_0$, $u_1 \in U_1$ and $y \in N(u_1) \cap Y_0'$. By the argument used in Claim 8, the set $Y_2' \cup \{y\}$ is independent. Then $\{u_0, u_1, y\}$ induces a P_3, so $G[\{u_0, u_1\} \cup Y_0']$ contains an induced $sP_1 + P_3$; a contradiction. Similarly, let $u_1, u_1' \in U_1$, $y_1 \in N(u_1) \cap Y_0'$ and $y_1' \in N(u_1') \cap Y_0'$. If $y_1 \neq y_1'$, then $\{y_1, u_1, u_1'\}$ induces a P_3 and by the same argument used in Claim 8, the set $Y_2' \cup \{y_1, y_1'\}$ is independent. Since $|Y_2'| = s + 2$, then $G[\{u_1, u_1'\} \cup Y_0']$ contains an induced $sP_1 + P_3$; a contradiction. This proves Claim 9.

Claim 10: $|U_2 \cap B_T| \leq 1$.
Assume there exist $u, u' \in U_2 \cap B_T$ with $u \neq u'$. By Claim 8, u and u' each have at least two neighbours in Y_2'. Hence, there exist vertices $y, y' \in Y_2'$ such that $y \in N(u)$, $y' \in N(u')$

and $y \neq y'$. By Claim 6, there is an even T-path P in $G[A'_0 \cup Y_2]$ between y and y'. Using the path $yuu'y'$, P can be extended to an odd T-cycle; a contradiction. This proves Claim 10.

Claim 11: Suppose that $u_1, u_2 \in B_T$ for some $u_1 \in U_1$ and $u_2 \in U_2$. Let $N(U_1) \cap Y'_0 = \{y\}$. Then $y \in Y'_0 \setminus Y_2$ and $y \notin B_T$.
Since u_2 has at least two neighbours in Y'_2, by Claim 8, u_2 has a neighbour $y' \in Y'_2$ such that $y' \neq y$. By Claim 6, there is an even T-path P in $G[A_0 \cup Y_2 \cup \{y\}]$ between y and y'. Using the path yu_1u_2y', the path P can be extended to an odd T-cycle. Since $V(P) \setminus \{y\} \subseteq A_0 \cup Y_2 \subseteq B_T$ and $u_1, u_2 \in B_T$, we deduce that $y \notin B_T$. This proves Claim 11.

Our approach is to consider each possible pair of sets A_0 and Y'_2 with $s \leq |A_0| \leq \max\{2, 2s-1\}$ and $|Y'_2| = s+2$ that conform with the definitions of this subcase and Claim 6. We want to find the largest possible B_T that contains them. Thus we want to include in B_T as many vertices as possible from $Y'_0 \setminus Y_2$ and Z. We first describe, for each component $G[U]$ of $G[Z]$, how to find the largest possible set of vertices U' in U to add to $A_0 \cup Y'_2$. As before, we let $S_T = V \setminus B_T$. We then prove, as Claim 12, the correctness of the approach of considering each component independently; that is, we prove that we cannot introduce any odd T-cycles that meet multiple components of $G[Z]$. We then complete the proof by considering which vertices of $Y'_0 \setminus Y_2$ to add to B_T.

First consider whether it is possible to find U' such that $|U'| \geq 3$. Then U' contains no vertex of T, otherwise $G[U']$ has an odd T-cycle, since U is a clique. By Claim 10, $|U' \cap U_2| \leq 1$. By Claim 9, at most one of U_0 and U_1 is non-empty. Hence, if $U_0 \setminus T \neq \emptyset$, then we let U' contain $(U_0 \setminus T)$, and, if $U_2 \setminus T \neq \emptyset$, we also add to U' an arbitrary $u \in U_2 \setminus T$.

If $U_0 \setminus T = \emptyset$, then possibly $U_1 \setminus T \neq \emptyset$. By Claim 9, there exists $y \in Y'_0$ such that $N(u) \cap Y'_0 = \{y\}$, for all $u \in U_1$. As $U_1 \cup \{y\}$ is a clique, we assume that $y \notin Y'_2 \cap T$; otherwise $|U_1 \cap U'| \leq 1$ and hence $|U'| \leq 2$ by Claim 10. If $U_2 \setminus T \neq \emptyset$, then $U' = (U_1 \setminus T) \cup \{u\}$ for an arbitrary $u \in U_2 \setminus T$, and, by Claim 11, we also have $y \in S_T$. If $U_2 \setminus T = \emptyset$, then we set $U' = U_1 \setminus T$ and if $y \in T$, then $y \in S_T$.

We now assume that we want to find U' such that $|U'| \leq 2$. First consider when $U_0 \neq \emptyset$ (so, by Claim 9, $U_1 = \emptyset$). If $|U_0| \geq 2$, then we set $U' = \{u, u'\}$ for any distinct $u, u' \in U_0$. If $U_0 = \{u_0\}$ and $|U_2| \geq 1$, then we set $U' = \{u_0, u_2\}$ for an arbitrary $u_2 \in U_2$. Finally, if $U_0 = \{u_0\}$ and $U_2 = \emptyset$, then $U' = \{u_0\}$.

Now consider when $U_0 = \emptyset$. If $U_1 \neq \emptyset$, then, by Claim 9, there is some $y \in Y'_0$ such that $U_1 \cup \{y\}$ is a clique. If $y \notin Y'_2 \cap T$ and $|U_2 \setminus T| \geq 2$, then set $U' = U_2 \setminus T$ and put $y \in S_T$. If $y \in Y'_2 \cap T$ then set $U' = \{u_1, u_2\}$ for an arbitrary $u_1 \in U_1$ and some

$u_2 \in U_2 \setminus T$ such that $y \notin N(u_2)$, if such an element u_2 exists. Otherwise, $|U'| \leq 1$, and we set $U' = \{u\}$ for an arbitrary $u \in U_1 \cup U_2$.

Claim 12: Let Z^* be a subset of Z. If the graph $G[A \cup Y \cup Z^*]$ contains an odd T-cycle C, then there exists a component $G[U^*]$ of $G[Z]$ such that the graph $G[A \cup Y \cup U^*]$ contains an odd T-cycle C' and $C \cap U^* = C' \cap Z^*$.

Let us assume there is an odd T-cycle C in $G[A \cup Y \cup Z^*]$. Without loss of generality, we may assume that for each component $G[U]$ of $G[Z]$ that intersects C, $C \cap U$ induces a path; if not, then there is a shorter odd T-cycle of $G[A \cup Y_0' \cup U]$ having the same property. The cycle C is the concatenation of a number of the following two types of paths: a path is of type (1) if it starts and ends in Y_0' and is contained $G[A \cup Y_0']$; a path is of type (2) if it starts and ends in Y_0' and all the internal vertices are contained in a component of $G[Z]$.

Since $G[A \cup Y_0']$ is bipartite, all the sub-paths of C of type (1) are even. Moreover, since C is an odd cycle, there is a path P of type (2) that is odd. Recall that P is a path starting and ending in Y_0' with all the internal vertices in a component, say $G[U^*]$, of $G[Z]$. By Claim 6, P can be extended to an odd T-cycle of $G[A \cup Y_0' \cup U^*]$. This proves Claim 12.

Finally we ask which vertices in $Y_0' \setminus Y_2$ to add to B_T. First note that $G[Y_0' \setminus Y_2]$ is P_3-free; indeed, if a component $G[W]$ of $G[Y_0' \setminus Y_2]$ contains an induced P_3, then $G[Y_2 \cup W]$ has an induced $sP_1 + P_3$ subgraph, as Y_2 is anti-complete to $Y_0' \setminus Y_2$ by Claim 7. So $G[Y_0' \setminus Y_2]$ is a disjoint union of complete graphs. By Claim 6, there is an even T-path between any pair of vertices of $G[Y_0']$, so we keep at most one vertex of each clique. For some component $G[U]$ of $G[Z]$ such that $N(U_1) \cap Y_0' = \{y\}$ and $y \in T$, we may have forced $y \in S_T$, when $|U'| \geq 3$. It is always optimal to have $y \in S_T$ in such a case else we would have $|U'| = 1$, since $U' \cup \{y\}$ is a clique. So for each clique $G[W]$ of $G[Y_0' \setminus Y_2]$, we include a vertex of $W \setminus S_T$ in B_T. □

We are now ready to prove our almost-complete classification.

Theorem 5 (restated). *Let H be a graph with $H \neq sP_1 + P_4$ for all $s \geq 1$. Then* Subset Odd Cycle Transversal *on H-free graphs is polynomial-time solvable if $H = P_4$ or $H \subseteq_i sP_1 + P_3$ for some $s \geq 1$ and* NP*-complete otherwise.*

Proof. If H has a cycle or claw, we use Theorem 3. The cases $H = P_4$ and $H = 2P_2$ follow from Theorems 9 and 10, respectively. The remaining case, where $H \subseteq_i sP_1 + P_3$, follows from Theorem 11. □

3.6 Conclusions

We showed that Odd Cycle Transversal is NP-complete on (P_2+P_5, P_6)-free and Subset Odd Cycle Transversal is NP-complete on split graphs. Moreover we gave almost-complete classifications of the complexity of Subset Feedback Vertex Set and Subset Odd Cycle Transversal for H-free graphs. The only open case in each classification is when $H = sP_1 + P_4$ for some $s \geq 1$, which is also open for Feedback Vertex Set and Odd Cycle Transversal for H-free graphs.

Open Problem 1 *Determine the complexity of* (Subset) Feedback Vertex Set *and* (Subset) Odd Cycle Transversal *for $(sP_1 + P_4)$-free graphs, when $s \geq 1$.*

One of the main obstacles to solve Open Problem 1 is the case where there is a solution S such that $G - S$ is a forest that contains (many) arbitrarily large stars. In particular, Lemma 3 no longer holds.

Open Problem 2 *Determine whether there exists an integer $r \geq 5$ such that* (Subset) Feedback Vertex Set *is* NP*-complete for P_r-free graphs.*

The vertex-weighted version of Subset Feedback Vertex Set has also been studied for H-free graphs. Papadopoulos and Tzimas [88] proved that Weighted Subset Feedback Vertex Set is polynomial-time solvable for $4P_1$-free graphs but NP-complete for $5P_1$-free graphs (in contrast to the unweighted version). Bergougnoux et al. [8] proved that Weighted Subset Feedback Vertex Set is polynomial-time solvable for P_4-free graphs. Recently Brettell et al. [18] solved the cases $H \in \{P_1 + P_2, P_1 + P_3\}$ with polynomial-time algorithms. Combining these results with Theorem 4 still leaves three gaps.

Open Problem 3 *Determine the complexity of* Weighted Subset Feedback Vertex Set *for H-free graphs when $H \in \{2P_1 + P_3, P_1 + P_4, 2P_1 + P_4\}$.*

For the weighted variant, a vertex in T may have a large weight that prevents it from being deleted in any solution; in particular, Lemma 11, which plays a crucial role in our proofs, no longer holds.

We note that the NP-completeness proof given by Papadopoulos and Tzimas for Weighted Subset Feedback Vertex Set on $5P_1$-free graphs [88] can also be used to show that the Weighted Subset Odd Cycle Transversal is NP-complete for $5P_1$-free graphs. Brettell et al. [18] proved that for $H \in \{3P_1 + P_2, P_1 + P_3\}$ this problem is polynomial time solvable. Combining these results with Theorem 5 still leaves three gaps.

Open Problem 4 *Determine the complexity of* Weighted Subset Odd Cycle Transversal *for H-free graphs when* $H \in \{2P_1 + P_3, P_1 + P_4, 2P_1 + P_4\}$.

We also introduced the Subset Vertex Cover problem and showed that this problem is polynomial-time solvable on $(sP_1 + P_4)$-free graphs for every $s \geq 0$.

Open Problem 5 *Determine the complexity of* Subset Vertex Cover *for* P_5*-free graphs.*

Open Problem 6 *Determine whether there exists an integer* $r \geq 5$ *such that* (Subset) Vertex Cover *is* NP*-complete for* P_r*-free graphs.*

Let us recall that Vertex Cover becomes polynomial-time solvable on $K_{1,3}$-free graphs [78,95] and on sP_2-free graphs [15]. We did not research the complexity of Subset Vertex Cover on either $K_{1,3}$-free or sP_2-free graphs and also leave these as open problems for future work.

Open Problem 7 *Determine the complexity of* Subset Vertex Cover *for* $K_{1,3}$*-free graphs.*

Open Problem 8 *Determine the complexity of* Subset Vertex Cover *for* sP_2*-free graphs.*

Finally, several related transversal problems have been studied but not yet for H-free graphs. For example, the parameterized complexity of Even Cycle Transversal and Subset Even Cycle Transversal has been addressed in [80] and [65], respectively. Moreover, several other transversal problems have been studied for H-free graphs, but not the subset version; see [12,28,36,37,64] for a number of recent results. It would be interesting to solve the subset versions of those transversal problems for H-free graphs and to determine the connections amongst all these problems in a more general framework.

4 Connected Vertex Cover Extension

For a graph $G = (V, E)$ and a subset $W \subseteq V$, a set $S_W \subseteq V$ is a *connected vertex cover* for W if it is a vertex cover that induces a connected subgraph and contains W.

This chapter is entirely dedicated to the following problem.

Connected Vertex Cover Extension
Instance: a graph $G = (V, E)$, a set $W \subseteq V$ and a positive integer k.
Question: does G have a connected vertex cover S_W for W and $\lvert S_W \rvert \leq k$?

This problem is NP-complete by [83]. For this reason we consider the restriction of the input to hereditary graph classes in order to better understand which graph properties cause the computational hardness. Moreover we aim to extend and strengthen existing complexity results on hereditary graph classes, expecially those found by forbidding a unique induced subgraph.

4.1 Existing Results

In 1977, Garey and Johnson [48] proved that Connected Vertex Cover is NP-complete for planar graphs of maximum degree 4. More recently, Priyadarsini and Hemalatha [90] and Fernau and Manlove [43] strengthened this result to 2-connected planar graphs of maximum degree 4 and planar bipartite graphs of maximum degree 4, respectively. Wanatabe, Kajita and Onaga [99] proved that Connected Vertex Cover is NP-complete even for 3-connected graphs. Very recently, Munaro [83] proved the same for line graphs of planar cubic bipartite graphs and for planar bipartite graphs of arbitrarily large girth, and Li, Yang, and Wang [71] showed NP-completeness for 4-regular graphs. Chiarelli, Hartinger, Johnson, Milanič, and Paulusma in [28] observed that the results of Munaro [83] imply that Connected Vertex Cover is NP-complete for H-free graphs if H contains a cycle or a claw. It is not known if there exists an integer r such that Connected Vertex Cover is NP-complete for P_r-free graphs.

We now turn to tractable cases. Ueno, Kajitani and Gotoh [97] proved that Connected Vertex Cover is polynomial-time solvable for graphs of maximum degree at most 3. Escoffier, Gourvès and Monnot [41] proved the same result for chordal graphs. By using the concept of the price of connectivity [22,26,59], Chiarelli et al. [28] proved that Connected Vertex Cover is polynomial-time solvable for sP_2-free graphs for any integer $s \geq 1$.

4.2 Our Results and Method

The main result of the chapter, which is proved in Section 4.4, largely extends tractable cases for Connected Vertex Cover:

Theorem 12. *For every* $s \geq 0$, Connected Vertex Cover Extension *can be solved in* $O(n^{19s^3+24})$ *time for* $(sP_1 + P_5)$*-free graphs.*

Remark 1 *Let* (G, W, k) *be an input of* Connected Vertex Cover Extension. *Then we may assume the graph G is connected. If it is not, then either at most one connected component of G intersects W and has edges, in which case isolated vertices do not need to be considered, or the answer is immediately no. Testing whether or not an input has an immediate no answer can be done in* $O(n + m)$*-time.*

It is easy to construct graphs with a minimum connected vertex cover that do not contain a minimum vertex cover; see the graph G_1 in Figure 16. We also note that the difference in size between a minimum vertex cover and a minimum connected vertex cover in an $(sP_1 + P_5)$-free graph is at most 3 if $s = 0$, and at most $3s + 10$ if $s \geq 1$ [59]. We cannot exploit this property directly as that would require an algorithm to enumerate all minimum vertex covers in polynomial time. Moreover, the graph G_2 in Figure 16 shows that even if this was possible, it is not immediately obvious how to proceed; one cannot necessarily hope to find a minimum connected vertex cover by extending a minimum vertex cover. As an extra complication, for Connected Vertex Cover one cannot extend results on H-free graphs to results on $(sP_1 + H)$-free graphs in a straightforward way, like as in Lemma 14.

Our method is based on a structural analysis of dominating sets in $(sP_1 + P_5)$-free graphs using the characterization of P_5-free graphs due to Bacsó and Tuza [3] given in Lemma 7. We translate the problem into a problem in which we try to extend a partial vertex cover into a full connected vertex cover. We solve this variant of Connected Vertex Cover by using Theorem 1 (applied to the smaller class of $(sP_1 + P_5)$-free graphs). We show how to do this in Section 4.3 and then show how to use this result to prove Theorem 12 in Section 4.4.

An important ingredient of our proof is that we reduce the size of the input graph by contracting an edge between two vertices u and v whenever we detect that u and v will both belong to the connected vertex cover. This idea stems from the observation that a connected graph G on n vertices has a connected vertex cover of size k if and only if G contains the star $K_{1,n-k}$ on $n - k + 1$ vertices as a contraction. If G has a connected vertex cover S of size k, then contracting every edge between vertices in S modifies G into

$K_{1,n-k}$. If G contains $K_{1,n-k}$ as a contraction, then $V(G)$ can be partitioned into sets A, $B_1, \ldots, B_{n-k}$ that each induce a connected graph such that there exists at least one edge between a vertex from A and a vertex from B_i for $i = 1, \ldots, n-k$ and no edges between two vertices from different B-sets. If $|B_i| \geq 2$, then we move every vertex that is adjacent to a vertex of A to A until we have only one vertex in B_i left. This gives us a connected vertex cover of size k.

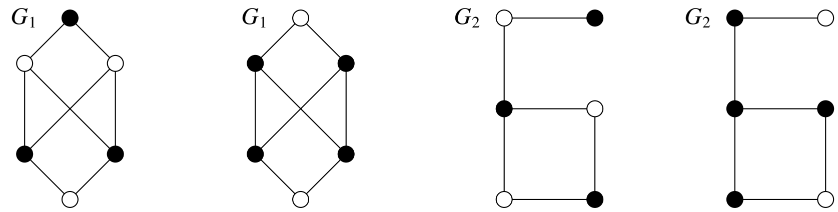


Fig. 16: An example of a P_5-free graph G_1 with a minimum connected vertex cover (coloured black in the right-hand drawing) that contains no minimum vertex cover (there are exactly two, indicated by the sets of black and white vertices in the left-hand drawing). The graph G_2 is an example of a $(P_1 + P_5)$-free graph with a minimum vertex cover (coloured black in the left hand drawing) that is not contained in any minimum connected vertex cover: clearly any connected vertex cover that contains it has at least five vertices and an example of a minimum connected vertex cover on four vertices is indicated by the vertices coloured black in the right-hand drawing.

Finally in Section 4.5, we prove Theorem 12 can be extended to Weighted Connected Vertex Cover Extension .

4.3 An Auxiliary Problem

In this section we prove that a variant of Connected Vertex Cover can be solved in polynomial time for $(sP_1 + P_5)$-free graphs for every integer $s \geq 0$. To prove Theorem 12 we will solve a polynomial number of instances of this variant, which we show can be solved in polynomial time for $(sP_1 + P_5)$-free graphs for every $s \geq 0$. We introduce the variant by first describing its input. Let $G = (V, E)$ be a connected graph, let $J \subseteq V$ be a subset of the vertex set of G and let y be a vertex of J. We call the triple (G, J, y) *cover-complete* if it has the following properties (see also Figure 17):

(A) J is an independent set;
(B) y is adjacent to every vertex of $G - J$;

(C) the neighbours of each vertex in $J \setminus \{y\}$ form an independent set in $G - J$.

We now describe the problem.

Connected Vertex Cover Completion
Instance: a cover-complete triple (G, J, y).
Task: find a smallest connected vertex cover S of G such that $J \subseteq S$.

We will show how to solve this problem in polynomial time for $(sP_1 + P_5)$-free graphs for any $s \geq 0$. We first give some further definitions and then prove a number of lemmas.

Let (G, J, y) be a cover-complete triple, where G is a connected $(sP_1 + P_5)$-free graph. For a vertex $w \in N_G(J \setminus \{y\})$, we write $J_w = N_G(w) \cap J$. Note that, by (B), $y \in J_w$. Let G' be the graph obtained from G by contracting every edge of $G[J_w \cup \{w\}]$. As $G[J_w \cup \{w\}]$ is connected, contracting its edges reduces it to a single vertex which we denote y_w. We say that we have *set-contracted* G into G' via w and that we *contracted* $J_w \cup \{w\}$ into y_w; see Figure 17 for an example.

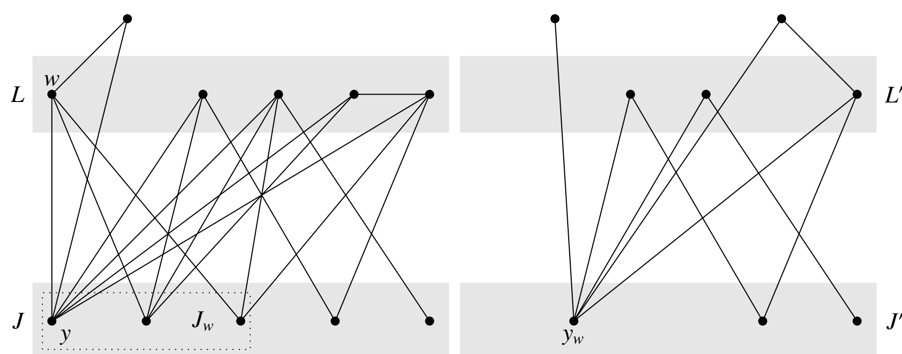


Fig. 17: An example of a cover-complete triple (G, J, y) and the cover-complete triple (G', J', y_w) obtained from set-contracting G via vertex w. The sets $J' = (J \setminus J_w) \cup \{y_w\}$, $L = N_G(J \setminus \{y\})$ and $L' = N_{G'}(J' \setminus \{y_w\})$ are also displayed (the latter two sets will be formally introduced later).

The following lemma is crucial.

Lemma 16. *Let (G, J, y) be a cover-complete triple, where G is a connected $(sP_1 + P_5)$-free graph for some $s \geq 0$. Let $w \in N_G(J \setminus \{y\})$, and let G' be the graph obtained from G after set-contracting via w. Let $J' = (J \setminus J_w) \cup \{y_w\}$ and $y' = y_w$. Then the following statements hold:*

1. *G' is a connected $(sP_1 + P_5)$-free graph;*
2. *(G', J', y') is a cover-complete triple;*
3. *A set $S \subseteq V_G$ is a (smallest) connected vertex cover of G that contains $J \cup \{w\}$ if and only if $(S \setminus (J \cup \{w\})) \cup J'$ is a (smallest) connected vertex cover of G' that contains J'.*

Proof. We will prove 1-3 separately.

1. By Lemma 10, G' is connected and $(sP_1 + P_5)$-free. This proves 1.

2. We will prove (A)-(C) for (G', J', y'). Before we do this we first observe the following. As (B) holds for (G, J, y), we find that $y \in J$ is adjacent to w in G. Hence y belongs to J_w and thus to $J_w \cup \{w\}$, which is contracted to the single vertex y' in G'. Hence, y is not in G' and its role has been taken over by y', as we show below.

We first prove (A). As J is an independent set in G, we find that $J \setminus J_w$ is an independent set in G'. For contradiction, suppose that y' is adjacent to a vertex in $J \setminus J_w$. Then there is an edge between a vertex of $J \setminus J_w$ and a vertex of $J_w \cup \{w\}$ in G. However, this not possible as J is independent in G, and thus every edge in $G[J \cup \{w\}]$ is incident with w. Hence $J' = (J \setminus J_w) \cup \{y'\}$ is an independent set in G'. This proves (A).

We now prove (B). Recall that y belongs to $J_w \cup \{w\}$, which is contracted to y' in G'. Hence, as y is adjacent to every vertex of $G - J$ in G, we find that y' is adjacent to every vertex of $G' - J'$. This proves (B).

Finally we prove (C). Let $x \in J' \setminus \{y'\}$. Then x is not adjacent to y', as we showed above that J' is an independent set in G'. Then $N_{G'}(x) = N_G(x)$ is an independent set, as (C) holds for (G, J, y). This proves (C) and 2.

3. Let S be a connected vertex cover of G that contains $J \cup \{w\}$. Then S contains every vertex of $J_w \cup \{w\}$. Hence, contracting $J_w \cup \{w\}$ to y' yields a connected vertex cover $(S \setminus (J \cup \{w\})) \cup J'$ of G' that contains J'. Any connected vertex cover S' of G' that contains J' contains y'. Hence uncontracting the edges of $G[J_w \cup \{w\}]$ yields a connected vertex cover $(S' \setminus J') \cup J \cup \{w\}$ of G that contains $J \cup \{w\}$. Moreover, a set S^* of G that contains $J \cup \{w\}$ is a connected vertex cover of G that is smaller than S if and only if the set $(S^* \setminus (J \cup \{w\})) \cup J'$, which contains J', is a connected vertex cover of G' that is smaller than $(S \setminus (J \cup \{w\})) \cup J'$. This proves 3. □

Let (G, J, y) be a cover-complete triple. We define $L_J = N_G(J \setminus \{y\})$. If there is no ambiguity, we will just write $L = L_J$ (see also Figure 17). Note that, by (C), $N_G(z)$ is an independent set in $G - J$ for every $z \in J \setminus \{y\}$, but L itself might not be independent. However, we can deduce the following lemma, which follows immediately from (C).

Observation 1 *Let (G, J, y) be a cover-complete triple. If w_1 and w_2 are two adjacent vertices in L, then no vertex of $J \setminus \{y\}$ is adjacent to both w_1 and w_2.*

We introduce two key definitions for a cover-complete triple (G, J, y). Two vertices $w_1, w_2 \in L$ form a *pseudo-dominating pair* if

- w_1 and w_2 are non-adjacent;
- w_1 has a neighbour $x_1 \in J$ not adjacent to w_2; and
- w_2 has a neighbour $x_2 \in J$ not adjacent to w_1.

Three vertices $w_1, w_2, w_3 \in L$ form a *pseudo-dominating triple* if

- w_1 is adjacent to neither w_2 nor w_3;
- w_2 and w_3 are adjacent;
- J contains two distinct vertices x_1 and x_2 such that
  - $x_1 \in N_G(w_1) \setminus N_G(\{w_2, w_3\})$ and
  - $x_2 \in (N_G(w_1) \cap N_G(w_2)) \setminus N_G(w_3)$.

See the illustrations in Figure 18, from which we also observe that no pseudo-dominating pair or pseudo-dominating triple can be found in a P_5-free graph.

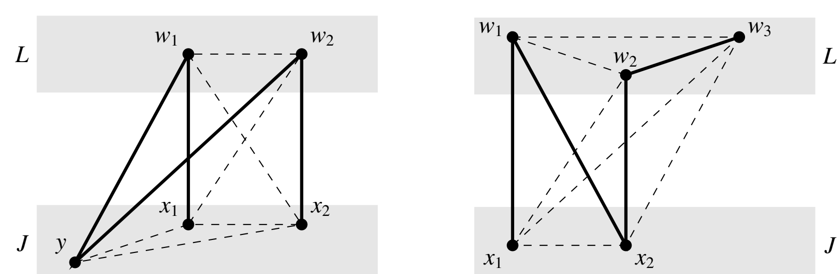


Fig. 18: Examples, on the left, of a pseudo-dominating pair (w_1, w_2), and, on the right, of a pseudo-dominating triple (w_1, w_2, w_3). As easily seen, the presence of either implies the existence of at least one induced P_5. To explain our notion of pseudo-domination, note that the vertices of any induced $(s-1)P_1 + P_5$ dominate the graph.

Let S be a connected vertex cover of G that contains J. Recall that J is an independent set. A subset $L^* \subseteq L \cap S$ is a *connector* of S if $J \cup L^*$ is connected. We present the following two lemmas.

Lemma 17. *Let (G, J, y) be a cover-complete triple, where G is an $(sP_1 + P_5)$-free graph for some $s \geq 0$. Let S be a connected vertex cover of G that contains J. If S contains both vertices of a pseudo-dominating pair w_1, w_2, then S has a connector of size at most $s + 1$ that contains both w_1 and w_2.*

Proof. By definition, there exist two vertices x_1 and x_2 in J, such that w_1 is not adjacent to x_2 and w_2 is not adjacent to x_1. As J is an independent set by (A) and each vertex of L is adjacent to y by (B), we find that $\{x_1, w_1, y, w_2, x_2\}$ induces a P_5 in that order. As G is $(sP_1 + P_5)$-free and J is an independent set, this means that $\{w_1, w_2\}$ dominates all vertices of J except for a subset $I \subseteq J$ of at most $s - 1$ vertices. We choose L^* to consist of w_1, w_2 and a neighbour in $L \cap S$ of each vertex of I (note that such a neighbour must exist for each vertex of I as S is connected). Then $J \cup L^*$ is connected, that is, L^* is a connector, as each vertex of J is adjacent to some vertex of L^* and each vertex of L^* is adjacent to $y \in J$ due to (B). Moreover, L^* has size at most $s + 1$. □

Lemma 18. *Let (G, J, y) be a cover-complete triple, where G is an $(sP_1 + P_5)$-free graph for some $s \geq 0$. Let S be a connected vertex cover of G that contains J. If S contains all three vertices of a pseudo-dominating triple w_1, w_2, w_3, then S has a connector of size at most $s + 2$ that contains $\{w_1, w_2, w_3\}$.*

Proof. By definition, there exist two vertices x_1 and x_2 in J such that x_1 is adjacent to w_1 but not to w_2 and w_3, and x_2 is adjacent to w_1 and w_2 but not w_3. Then $\{x_1, w_1, x_2, w_2, w_3\}$ induce a P_5 in that order. As G is $(sP_1 + P_5)$-free and J is an independent set, this means that $\{w_1, w_2, w_3\}$ dominates all vertices of J except for a subset $I \subseteq J$ of at most $s - 1$ vertices. We choose L^* to consist of w_1, w_2, w_3 and a neighbour in $L \cap S$ of each vertex of I (note that such a neighbour must exist for each vertex of I as S is connected). Then $J \cup L^*$ is connected, that is, L^* is a connector, as each vertex of J is adjacent to some vertex of L^* and each vertex of L^* is adjacent to $y \in J$ due to (B). Moreover, L^* has size at most $s + 2$. □

Let (G, J, y) be a cover-complete triple. Let S be a connected vertex cover of G that contains J. If S contains both vertices of some pseudo-dominating pair of G or all three vertices of some pseudo-dominating triple of G, then S is of *type 1*. Otherwise S must contain at most one vertex of any pseudo-dominating pair and at most two vertices of any pseudo-dominating triple of G. In that case we say that S is of *type 2*. We observe that G might have connected vertex covers of only one type.

We will now see, in Lemma 20, how to find a smallest type 1 connected vertex cover of a graph G of a cover-complete triple (G, J, y) in polynomial time (if it exists). After that we shall prove how to find a smallest type 2 connected vertex cover of G

in polynomial time (if it exists). To compute these sets we need the following lemma, which uses Theorem 1 in its proof.

Lemma 19. *Let $(G, \{y\}, y)$ be a cover-complete triple, where G is an $(sP_1 + P_5)$-free graph for some $s \geq 0$. Then it is possible to compute a smallest connected vertex cover of G that contains y in $O(n^{s+14})$ time.*

Proof. As $G - y$ is $(sP_1 + P_5)$-free, we can, by Theorem 1, compute in polynomial time a smallest vertex cover S of $G - y$. As $(G, \{y\}, y)$ is a cover-complete triple, y dominates G. Hence, $S \cup \{y\}$ is a smallest connected vertex cover of G that contains y.

To establish the bound on the running time we need only describe how to compute a smallest vertex cover of $G - y$ in $O(n^{s+14})$ time. This is achieved by presenting an algorithm for the complementary problem of computing a maximum independent set in $G - y$. We first determine by brute force, in time $O(n^s)$, the largest integer $s' \leq s$, such that $G - y$ has an independent set of size s'. If $s' \leq s - 1$, then s' is the size of a largest independent set of $G - y$ and we are done. Otherwise, if $s' = s$, we consider each set S' of s independent vertices of $G - y$. For each choice, we remove the vertices of S' and their neighbours from $G - y$. The remaining graph is P_5-free and we use the algorithm of [72], which runs in $O(n^{14})$ time, to find a maximum independent set therein. This set is added to S' to give an independent set of $G - y$. The largest independent set found in this way must be of maximum size. □

Using Lemmas 17–19, we are now ready to deal with type 1 smallest connected vertex covers.

Lemma 20. *Let (G, J, y) be a cover-complete triple. It is possible to find in $O(n^{2s+16})$ time a smallest type 1 connected vertex cover of G.*

Proof. We can compute all pseudo-dominating pairs of G by examining each pair of vertices in turn. This takes $O(n)$ time per pair. As the number of pseudo-dominating pairs is $O(n^2)$, this takes $O(n^3)$ time in total.

For each pseudo-dominating pair (w_1, w_2) of G, we describe how to compute a smallest connected vertex cover S_{w_1,w_2} of G that contains $J \cup \{w_1, w_2\}$. By Lemma 17, such a vertex cover must have a connector L^* of size at most $s + 1$ that contains w_1 and w_2. We find each such connector L^* by considering all sets of up to $s - 1$ vertices and asking whether, combined with w_1 and w_2, they form such a connector.

For each such set L^*, we do as follows. We first check if $J \cup L^*$ is connected. If so, then we apply Lemma 16 recursively for each $w \in L^*$. This takes $O(n^2)$ time, as we can use Breadth First Search and set contract at the same time. Let (G', J', y') be the resulting cover-complete triple. Then $J' = \{y'\}$, which means we can apply Lemma 19 to find a smallest connected vertex cover S' of G' in $O(n^{14+s})$ time. By Lemma 16, we can translate S' into the desired vertex cover S_{w_1,w_2} by uncontracting any contracted edges. As, for each pseudo-dominating pair, the number of sets L^* that contain them is $O(n^{s-1})$, and the number of pseudo-dominating pairs is $O(n^2)$, the time needed to find these vertex covers is $O(n^{2s+15})$.

For each pseudo-dominating triple (w_1, w_2, w_3) of G we compute a smallest connected vertex cover S_{w_1,w_2,w_3} of G that contains $J \cup \{w_1, w_2, w_3\}$. We can do this in $O(n^{2s+16})$ time by exactly the same arguments: the only differences are that the number of pseudo-dominating triples is $O(n^3)$ and that we need to apply Lemma 18 instead of Lemma 17.

From all the computed sets S_{w_1,w_2} and S_{w_1,w_2,w_3} we keep track (in constant time) of a smallest one, and in the end this yields a smallest type 1 connected vertex cover of G. This proves Lemma 20. ◻

Let (G, J, y) be a cover-complete triple. Using Lemma 20 we can find a smallest type 1 connected vertex cover of G in polynomial time. However, it might be possible that G has a smaller connected vertex cover of type 2. To investigate this, we introduce two reduction rules that will transform a cover-complete triple (G, J, y) into a triple (G', J', y') with $|J'| < |J|$. We say that such a rule is *safe* if the following three conditions hold:

1. If G is $(sP_1 + P_5)$-free and connected, then G' is $(sP_1 + P_5)$-free and connected.
2. (G', J', y') is cover-complete.
3. Given a smallest connected vertex cover S' of G' that contains J', it is possible, in $O(n^{2s+16})$ time, to find a smallest connected vertex cover S of G that contains J.

Rule 1. Set-contract via x whenever x is a vertex in $L \cap N_G(w_1) \cap N_G(w_2)$ for some pseudo-dominating pair (w_1, w_2).

Rule 2. For any vertex $w_5 \in L$ that is not adjacent to any vertex of a clique of four vertices w_1, w_2, w_3, w_4 in L, delete w_5 and set-contract via u for every $u \in L \cap N_G(w_5)$.

Lemma 21. *Rules 1 and 2 are safe.*

Proof. We first consider Rule 1.

Let (G', J', y') be the resulting triple after an application of Rule 1, where $J' = (J \setminus J_x) \cup \{y_x\}$ and $y' = y_x$. By Lemma 16, (G', J', y') is a cover-complete triple. By the same lemma, G' is $(sP_1 + P_5)$-free and connected if G is $(sP_1 + P_5)$-free and connected. Hence we have proven that conditions 1 and 2 hold.

We are left to prove condition 3. Let S' be a smallest connected vertex cover in G' that contains J'. Then $S = (S' \setminus \{y'\}) \cup J_x \cup \{x\}$ is a smallest connected vertex cover of G that contains $J \cup \{x\}$ due to Lemma 16. We prove the following claim.

Claim 1: For any type 2 connected vertex cover T of G, it holds that $|T| \geq |S|$.
We prove Claim 1 as follows. Let T be a connected vertex cover T of G that is of type 2. Suppose $x \notin T$. Then, as x is adjacent to both w_1 and w_2, we find that T contains both w_1 and w_2. Thus T is not of type 2, a contradiction. Hence T contains x. This implies that the set $T' = (T \setminus (J \cup \{x\})) \cup J'$ is a connected vertex cover of G' that contains J'. As S' is a smallest connected vertex cover of G' that contains J', we find that $|T'| \geq |S'|$. Hence $|T| = |T'| + |J_x| \geq |S'| + |J_x| = |S|$. This proves Claim 1.

The above means that we can do as follows. Given S' we compute $S = (S' \setminus \{y'\}) \cup J_x \cup \{x\}$ in constant time. By Lemma 20 we can also compute, in $O(n^{2s+16})$ time, a smallest type 1 connected vertex cover S^* of G (note that $S = S^*$ is possible). If S is of type 2, then S is a smallest type 2 connected vertex cover of G, due to Claim 1. We compare $|S|$ and $|S^*|$ and choose the smallest one. If S is of type 1, then S^* is a smallest connected vertex cover of G, again due to Claim 1. This proves condition 3 and completes the proof that Rule 1 is safe.

We now consider Rule 2. We first show that w_5 cannot be in any connected vertex cover S of G that is of type 2. For contradiction, suppose that w_5 is in such a connected vertex cover S. Because S is a vertex cover and $\{w_1, w_2, w_3, w_4\}$ is a clique, S contains at least three of $\{w_1, w_2, w_3, w_4\}$, say w_1, w_2, w_3.

For $i = 1, \ldots, 5$, let X_i be the set of neighbours of w_i in J. As $w_i \in L$, every $X_i \neq \emptyset$ by definition of L. By Observation 1, we find that X_1, X_2 and X_3 are pairwise disjoint. Let $x \in X_1$. If $x \notin X_5$, then $X_5 \subseteq X_1$, as otherwise (w_1, w_5) is a pseudo-dominating pair of vertices that are both contained in S, which is not possible as S is of type 2. As $X_1 \cap X_2 = \emptyset$, we find that $X_5 \cap X_2 = \emptyset$. This means that (w_2, w_5) is a pseudo-dominating pair of vertices that are both contained in S, which is not possible either. Hence $x \in X_5$. We conclude that $X_1 \subseteq X_5$. For the same reason, we find that $X_2 \subseteq X_5$ and $X_3 \subseteq X_5$.

Recall that $X_1 \cap X_2 \cap X_3 = \emptyset$. Hence we can pick a vertex $x_1 \in X_1$ and a vertex $x_3 \in X_3$, which are both adjacent to w_5 but not to w_2, and so find that (w_5, w_1, w_2) is a pseudo-dominating triple. As all three vertices w_1, w_2, w_5 belong to S, while S is of type 2, this is not possible. Hence S does not contain w_5.

If $G - w_5$ is disconnected, then w_5 belongs to every connected vertex cover of G. From the above it follows that it is not possible to find a connected vertex cover of G that contains J of type 2 in this case. Now suppose that $G - w_5$ is connected. As no connected vertex cover of G of type 2 may contain w_5, any connected vertex cover of G

that is of type 2 must contain all neighbours of w_5, and we can delete w_5. The proof of conditions 1–3 is identical to the proof for Rule 1 where the neighbours of w_5 in L take the role of the vertex x in the proof for Rule 1. □

We call a cover-complete triple (G, J, y) *free* if G has no pseudo-dominating pair with a common neighbour in L, and moreover, $G[L]$ is $(P_1 + K_4)$-free. By exhaustively applying Rules 1 and 2 in arbitrary order, which we may safely do due to Lemma 21, we have the following lemma.

Lemma 22. *A cover-complete triple (G, J, y) can be modified, in $O(n^6)$ time, into a free cover-complete triple (G', J', y') with the following properties:*

1. *If G is $(sP_1 + P_5)$-free and connected, then G' is $(sP_1 + P_5)$-free and connected.*
2. *Given a smallest connected vertex cover S' of G' that contains J', it is possible to find in $O(n^{2s+17})$ time a smallest connected vertex cover S of G that contains J.*

Proof. We exhaustively apply Rules 1 and 2 in arbitrary order. Checking if Rule 1 can be applied takes $O(n^3)$ time, as there are $O(n^2)$ pairs of vertices and for each pair it takes $O(n)$ time to check if it is pseudo-dominating. Similarly, checking if Rule 2 can be applied takes $O(n^5)$ time. As each application of each of these rules takes $O(n)$ time, and reduces the size of G, this procedure will complete in $O(n^6)$ time. By repeated use of Lemma 21, this results in a cover-complete triple (G', J', y') that satisfies the two properties of the lemma; in particular given a a smallest connected vertex cover S' of G' that contains J', it is possible to find in $O(n^{2s+17})$ time a smallest connected vertex cover S of G that contains J, as we applied Rules 1 and 2 at most n times and by condition 3 we need $O(n^{2s+16})$ time per application. Moreover, G' contains no pseudo-dominating pair with a common neighbour in $L' = L_{J'}$ and $G'[L']$ is $(P_1 + K_4)$-free, as otherwise we could still apply Rule 1 or Rule 2, respectively. Hence (G', J', y') is a free cover-complete triple. □

Let (G, J, y) be a free cover-complete triple. A connector of a connected vertex cover S of G is *minimal* if it does not properly contain a smaller connector of S. The next three lemmas are on free cover-complete triples; the second makes use of the first.

Lemma 23. *Let (G, J, y) be a free cover-complete triple. Then every minimal connector L^* of every type 2 connected vertex cover S of G is a clique.*

Proof. For contradiction, suppose that L^* is not a clique. Then L^* contains two non-adjacent vertices w_1 and w_2. As L^* is a minimal connector, w_1 has a neighbour in J not adjacent to w_2, and vice versa. However, then (w_1, w_2) is a pseudo-dominating pair of G. This is not possible, as S is of type 2. □

Lemma 24. *Let (G, J, y) be a free cover-complete triple that has a pseudo-dominating pair (w_1, w_2). Then every minimal connector L^* of every type 2 connected vertex cover S of G has size at most* 5.

Proof. For contradiction, suppose that $|L^*| \geq 6$. By Lemma 23, L^* is a clique. As (G, J, y) is free, $G'[L']$ is $(P_1 + K_4)$-free by definition. Hence w_1 must be adjacent to at least three vertices of L^*, which we denote by x_1, x_2, x_3. Note that $\{w_1, x_1, x_2, x_3\}$ induces a K_4 in $G[L]$. By definition of a pseudo-dominating pair, w_1 and w_2 are non-adjacent. As (G, J, y) is free, w_2 is not adjacent to any neighbour of w_1 in L by definition. Hence w_2 is not adjacent to any vertex of $\{x_1, x_2, x_3\}$. This means that the set $\{w_1, w_2, x_1, x_2, x_3\}$ induces a $P_1 + K_4$ in $G[L]$, a contradiction. □

Lemma 25. *Let (G, J, y) be a free cover-complete triple that has no pseudo-dominating pair. It is possible to find in $O(n^3)$ time a clique $K \subseteq L$ with $N_G(K) \cap J = J$.*

Proof. We describe how to construct K. Consider a vertex $w_1 \in L$ that has maximal neighbourhood in J, that is, there is no vertex $w \in L$ with $N_G(w_1) \cap J \subsetneq N_G(w) \cap J$. We put w_1 in K. Suppose that at some point we have constructed a clique $K = \{w_1, \ldots, w_i\}$ for some $i \geq 1$. If $N_G(K) \cap J = J$, then we stop. Otherwise we pick a vertex w_{i+1} with maximal neighbourhood in $J \setminus N_G(K)$ over all vertices in L (or equivalently, all vertices in $L \setminus \{w_1, \ldots, w_i\}$). Note that w_{i+1} exists as G is connected.

Suppose that w_{i+1} is adjacent to some $x \in N_G(K) \cap J$. Then, by Observation 1, we find that x is adjacent to a unique vertex w_h in K. By the same lemma, w_{i+1} is not adjacent to w_h. As G has no pseudo-dominating pair and w_{i+1} has a neighbour in $J \setminus N_G(K)$ (that is, a neighbour not adjacent to w_h), we find that $N_G(w_h) \subsetneq N_G(w_{i+1})$. This means that we would have chosen w_{i+1} earlier, namely instead of w_h. Hence, w_{i+1} is not adjacent to any $x \in N_G(K) \cap J$. As G has no pseudo-dominating pairs, this means that w_{i+1} is adjacent to every w_j with $1 \leq j \leq i$. That is, we can extend K into a larger clique by adding w_{i+1}.

As we increase $N_G(K) \cap J$ each time we add a new vertex to K, our procedure will stop with the desired output $K = \{w_1, \ldots, w_r\}$ for some $r \geq 1$. We note that constructing K takes $O(n^3)$ time. □

We are now ready to prove the following theorem.

Theorem 13. *For every $s \geq 0$,* Connected Vertex Cover Completion *can be solved in $O(n^{2s+19})$ time for cover-complete triples (G, J, y), where G is an $(sP_1 + P_5)$-free graph.*

Proof. Let $s \geq 0$ and let (G, J, y) be a cover-complete triple, where G is an $(sP_1 + P_5)$-free graph. We first apply Lemma 22 to obtain a free cover-complete triple (G', J', y') in $O(n^6)$

time. By the same lemma, G' is $(sP_1 + P_5)$-free. Our aim is to find a smallest connected vertex cover of G' that contains J' in polynomial time, so that we can apply statement 2 of Lemma 22. We first compute in $O(n^{2s+16})$ time a smallest type 1 connected vertex cover S^* of G' using Lemma 20. We now need to compute a smallest type 2 connected vertex cover S' of G' and compare $|S'|$ with $|S^*|$.

We check if G' contains a pseudo-dominating pair. This takes $O(n^3)$ time, as G' contains $O(n^2)$ pairs of vertices and for each pair it takes $O(n)$ time to check if it is pseudo-dominating.

First suppose that G' contains a pseudo-dominating pair. For each set of at most five vertices, we check if it is a minimal connector of size at most 5, and if so we apply Lemma 16 on its vertices. This takes $O(n^2)$ time per set. If we obtain an instance of the form $(G'', \{y''\}, y'')$, then we apply Lemma 19, which takes $O(n^{s+14})$ time. Then we uncontract all contracted edges in $O(n)$ time to get a connected vertex cover of G' of type 2. By Lemma 24, doing this for every possible minimal connector of size at most 5 gives us a smallest type 2 connected vertex cover S' of G'. As we process each set of at most five vertices in $O(n^{s+14})$ time and the number of such sets is $O(n^5)$, we find S' in $O(n^{s+19})$ time. We compare S' and S^* and choose the smaller of the two.

Now suppose that G' has no pseudo-dominating pair. Let $L' = N_{G'}(J' \setminus \{y'\})$. By Lemma 25, we can obtain in $O(n^3)$ time a clique $K \subseteq L'$ with $N_{G'}(K) \cap J' = J'$. Let $K = \{w_1, \ldots, w_r\}$ for some $r \geq 1$. As K is a clique, every vertex cover contains at least $r - 1$ vertices of K. We will do as follows: first we will find in $O(n^{s+14})$ time a smallest connected vertex cover of G' that contains $J' \cup K$, and then we will find in $O(n^{s+17})$ time, for $i = 1, \ldots, r$, a smallest connected vertex cover of G' that contains $J' \cup (K \setminus \{w_i\})$ and that does not contain w_i. As there are $O(n)$ cases, the total time of processing this case is $O(n^{s+18})$.

We start by computing a smallest connected vertex cover of G' that contains $J' \cup K$ by set-contracting via each vertex of K. This takes $O(n^2)$ time. By Lemma 16, this yields a cover-complete triple $(G'', \{y''\}, y'')$ to which we apply Lemma 19 in $O(n^{s+14})$ time. Uncontracting all contracted edges yields, by Lemma 16, a smallest connected vertex cover S_K of G' that contains $J' \cup K$; this takes $O(n)$ time. Hence, the total running time for this step is $O(n^{s+14})$, as we claimed above.

We now show how to compute, in $O(n^{s+17})$ time, a smallest connected vertex cover of G' that contains $J' \cup (K \setminus \{w_1\})$ and that does not contain w_1. The cases where $i \geq 2$ are done in the same way.

We first note that if $G - w_1$ is disconnected, then w_1 belongs to every connected vertex cover of G'. Hence, in that case there is no connected vertex cover of G' that contains

$J' \cup (K \setminus \{w_1\})$ but does not contain w_1. Now suppose that $G - w_1$ is connected. Let $A = L' \setminus N_{G'}(w_1)$ consist of all non-neighbours of w_1 in L'. As $G'[L']$ is $(K_4 + P_1)$-free by definition, we find that $G'[A]$ is K_4-free. As w_1 is not in the connected vertex cover we are looking for, we remove w_1. Then we set-contract, in $O(n^2)$ time, via each neighbour of w_1 in L. By Lemma 16, we may now consider the resulting cover-complete triple (G'', J'', y'') where G'' is connected and $(sP_1 + P_5)$-free. As G' had no pseudo-dominating pairs, we have that G'' has no pseudo-dominating pairs. We write $L'' = N_{G''}(J'' \setminus \{y''\})$. As $L'' \subseteq A$, we find that $G''[L'']$ is K_4-free.

Claim 1: Every minimal connector L^* of every connected vertex cover of G'' that contains J'' has size at most 3.

We prove the claim by showing that L^* is a clique, which implies that L^* has size at most 3, as $G''[L'']$ is K_4-free. Suppose instead that L^* is not a clique. Then L^* contains two non-adjacent vertices w_1 and w_2. As L^* is a minimal connector, w_1 has a neighbour in J'' not adjacent to w_2, and vice versa. But then (w_1, w_2) is a pseudo-dominating pair of G'': this is not possible, as G'' has no pseudo-dominating pairs. This contradiction proves Claim 1.

We now consider all subsets in L'' that have size at most 3. For each set we check if it is a minimal connector, and if so we apply Lemma 16 on its vertices. This takes $O(n^2)$ time per subset. If we obtain an instance $(G''', \{y'''\}, y''')$, then we apply Lemma 19 in $O(n^{s+14})$ time. Then uncontracting all contracted edges yields a connected vertex cover of G'' that contains J''. As there are $O(n^3)$ subsets in L'' of size at most 3, the total running time is $O(n^{s+17})$, as we claimed above. We keep track (in constant time) of the smallest one of these connected vertex covers of G''. For this connected vertex cover of G'', we uncontract all contracted edges again to obtain a smallest connected vertex cover S_{w_1} of G' that contains $J' \cup (K \setminus \{w_1\})$ and that does not contain w_1.

As mentioned, we pick the smallest one out of the connected vertex covers S_K and S_{w_i}, $1 \leq i \leq r$, to obtain a smallest type 2 connected vertex cover of G', the size of which we compare with the size of S^*. We pick the smallest one.

Thus we obtain in $O(n^6) + O(n^{2s+16}) + O(n^3) + O(n^{s+19}) + O(n^{s+18}) = O(n^{2s+19})$ time a smallest connected vertex cover of G' that contains J' (both in the case where G' has a pseudo-dominating pair and in the case where G' has no pseudo-dominating pair). As stated, it remains to apply statement 2 of Lemma 22 to find in $O(n^{2s+17})$ time a smallest connected vertex cover of G that contains J. Hence the total running time is $O(n^{2s+19})$. The correctness of our algorithm follows immediately from the above case analysis and the description of the cases. □

4.4 Our Main Result

In this section we prove Theorem 12, that is, we show that Connected Vertex Cover Extension can be solved in polynomial time for $(sP_1 + P_5)$-free graphs for every integer $s \geq 0$. The proof relies heavily on Theorem 13. The main idea is to reduce an $(sP_1 + P_5)$-free input graph G of Connected Vertex Cover Extension to a polynomial number of instances (G_i, J_i, y_i) of Connected Vertex Cover Completion . We can then solve each of these instances (G_i, J_i, y_i) in polynomial time by Theorem 13. Then we translate the resulting connected vertex covers of G_i (which contain J_i) into connected vertex covers of G that contains the input set W. We pick the smallest of these sets as our final output.

We need one more lemma.

Lemma 26. *Let J be an independent set in a connected graph G such that J has a vertex y that is adjacent to every vertex of $G - J$. Let J' consist of those vertices of $J \setminus \{y\}$ that have two adjacent neighbours in $G - J$ (or equivalently, in G). Then a subset S is a connected vertex cover of G that contains J if and only if $S \setminus J'$ is a connected vertex cover of $G - J'$ that contains $J \setminus J'$.*

Proof. Let $w \in J \setminus \{y\}$ be a vertex in G with two neighbours a and b that are adjacent in $G - J$ (or equivalently in G). Let S be a subset of G. First suppose that S is a connected vertex cover of G that contains J. Then $S \setminus \{w\}$ is a vertex cover of $G - w$ that contains $J \setminus \{w\}$. As $y \in J$ and $y \neq w$, we find that $S \setminus \{w\}$ contains y. Then every vertex of $S \setminus \{w\}$ that belongs to $G - J$ is adjacent to y in $G[S \setminus \{w\}]$. Moreover, as S is connected and J is independent, every vertex of $J \setminus \{w\}$ must be adjacent in $G[S \setminus \{y\}]$ to a vertex of $G - J$. Hence, $S \setminus \{w\}$ is connected in $G - w$.

Now suppose that $S \setminus \{w\}$ is a connected vertex cover of $G - w$ that contains $J \setminus \{w\}$. Then S is a vertex cover of G that contains J. As $y \in J$, we find that S contains y. As ab is an edge, S contains at least one of a and b. Then w and y are connected in S either due to the edges ya, aw (if a is in S) or due to the edges yb, bw (if a is not in S, as then $b \in S$). Hence S is connected in G.

We now consider the graph $G - w$ and repeat the arguments above for any vertex in $J' \setminus \{w\}$. □

We are now ready to prove our main result.

Theorem 12 (restated) *For every $s \geq 0$,* Connected Vertex Cover Extension *can be solved in $O(n^{19s^3+24})$ time for $(sP_1 + P_5)$-free graphs.*

Proof. Let G be an $(sP_1 + P_5)$-free graph on n vertices for some $s \geq 0$ and let $W \subseteq V(G)$ be a subset of vertices of G. By Remark 1, we may assume that G is connected. By Lemma 9 we can first compute in $O(n^{2s^2+s+3})$ time a connected dominating set D that either has size at most $2s^2 + s + 2$ or is a clique. We note that, if D is a clique, any vertex cover of G contains all but at most one vertex of D. This leads to a case analysis where we guess the subset $D^* \subseteq D \setminus W$ of vertices not in a smallest connected vertex cover of G that contains W. That is, we choose a set of at most one vertex if D is a clique and a set of at most $|D \setminus W|$ vertices otherwise, and eventually look at all such sets. As $|D \setminus W| \leq |D| \leq 2s^2 + s + 2$ if D is not a clique, the number of guesses is $O(n^{2s^2+s+2})$. For each guess of D^*, we compute a smallest connected vertex cover S_{D^*} that contains all vertices of $(D \setminus D^*) \cup W$ and no vertex of D^*. Then, in the end, we return one that has minimum size overall. In particular we note that, since D is a connected dominating set of G, $D \cup W$ is also a connected dominating set of G.

Let D^* be a guess. Before we start our case analysis we first prove the following claim.

Claim 1: We may assume, at the expense of an $O(n^{14s^3+2})$ factor in the running time, that $D \setminus D^*$ is connected.

We prove Claim 1 as follows. Suppose $D \setminus D^*$ is not connected. Recall that $G[D]$ is either a complete graph or has size at most $2s^2 + s + 2$. In the first case, $G[D \setminus D^*]$ is connected. Hence, the second case applies so D has size at most $2s^2 + s + 2$. Let $v \in D \setminus D^*$. As G is $(sP_1 + P_5)$-free, G is also P_{5+2s}-free. Hence, for each $u \in D \setminus (D^* \cup \{v\})$, any connected vertex cover of G contains a path of at most $5 + 2s - 1$ vertices that connects u to v. We will guess all these paths from u to v (using only vertices from $G - D^*$) and add their vertices to D. As the number of paths is at most $2s^2 + s + 1$, this branching adds an $O(n^{(5+2s-3)(2s^2+s+1)}) = O(n^{14s^3+2})$ factor to our running time. We have proven Claim 1.

We distinguish two cases.

Case 1: $D^* = \emptyset$.

We compute a minimum vertex cover S' of $G-(D \cup W)$ in polynomial time by Theorem 31. To be more precise, this takes $O(n^{s+14})$ time by using the same arguments as in the proof of Lemma 19. Clearly $S' \cup D \cup W$ is a vertex cover of G. As D is a connected dominating set, $S' \cup D \cup W$ is even a connected vertex cover of G. Let $S_\emptyset = S' \cup D \cup W$. As S' is a minimum vertex cover of $G - (D \cup W)$, $S_\emptyset$ is a smallest connected vertex cover of G that contains all vertices of $D \cup W$. We remember $S_\emptyset$. Note that $S_\emptyset$ is found in $O(n^{s+14})$ time.

Case 2: $1 \leq |D^*| \leq |D|$ *(recall that $|D| \leq 2s^2 + s + 3$).*

Recall that we are looking for a smallest connected vertex cover of G that contains every vertex of $(D \setminus D^*) \cup W$, but does not contain any vertex of D^*. Hence D^* must be an independent set, disjoint from W, and $G - D^*$ must be connected (if one of these conditions is false, then we stop considering the guess D^*). Moreover, a vertex cover that contains no vertex of D^* must contain all vertices of $N_G(D^*)$. Hence we can safely contract not only any edge between two vertices of $(D \setminus D^*) \cup W$, but also any edge between two vertices in $N_G(D^*)$ or between a vertex of $(D \setminus D^*) \cup W$ and a vertex in $N_G(D^*)$. We perform edge contractions recursively and as long as possible while remembering all the edges that we contract. This takes $O(n)$ time. Let G^* be the resulting graph.

Note that the set D^* still exists in G^*, as we did not contract any edges with an endpoint in D^*. By Claim 1, the set $D \setminus D^*$ in G corresponds to exactly one vertex of G^*. We denote this vertex by y. The set W of G corresponds to an independent set of G^*. We denote this set by W^*. We observe the following equivalence, which is obtained after uncontracting all the contracted edges.

Claim 2: Every smallest connected vertex cover of G^* that contains $\{y\} \cup W^*$ and that does not contain any vertex of D^* corresponds to a smallest connected vertex cover of G that contains $(D \setminus D^*) \cup W$ and that does not contain any vertex of D^*, and vice versa.

As we obtained G^* in $O(n)$ time, and we can uncontract all contracted edges in $O(n)$ time as well, Claim 2 tells us that we may consider G^* instead of G. As G is connected and $(sP_1 + P_5)$-free, G^* is connected and $(sP_1 + P_5)$-free as well by Lemma 10.

We write $J^* = N_{G^*}(D^*) \cup W^*$ and note that y belongs to $N_{G^*}(D^*) \subseteq J^*$ as D is connected in G. We now consider the graph $G^* - D^*$. As $G - D^*$ is connected, $G^* - D^*$ is connected. By Claim 2, our new goal is to find a smallest connected vertex cover of $G^* - D^*$ that contains J^*. By our procedure, J^* is an independent set of $G^* - D^*$. As D dominates G, we find that $D \setminus D^*$ dominates every vertex of $G - D^*$ that is not adjacent to a vertex of D^*. Hence the vertex y, to which the vertices of $D \setminus D^*$ have been contracted, is adjacent to every vertex of $(G^* - D^*) - J^*$ in the graph $G^* - D^*$.

Let $J \subseteq J^*$ consist of y and those vertices in J^* whose neighbourhood in $G^* - D^*$ is an independent set. As y is adjacent to every vertex of $(G^* - D^*) - J^*$ in $G^* - D^*$, and we can remember the set $J^* \setminus J$, we can apply Lemma 26 and remove $J^* \setminus J$. That is, it suffices to find a smallest connected vertex cover of the graph $G' = (G^* - D^*) - (J^* \setminus J)$ that contains J.

As J^* is an independent set of $G^* - D^*$, we find that J is an independent set of G'. By definition, $y \in J$. As y is adjacent to every vertex of $(G^* - D^*) - J^*$ in $G^* - D^*$, we find that y is adjacent to every vertex in $G' - J$. By definition, the neighbours of each vertex in $J \setminus \{y\}$ form an independent set in $G' - J$. Hence the triple (G', J, y) is cover-complete. This means that we can apply Theorem 13 to find in $O(n^{2s+19})$ time a smallest connected vertex cover S' of G' that contains J.

We translate S' in constant time into a smallest connected vertex cover S^* of $G^* - D^*$ that contains J^* by adding $J^* \setminus J$ to S'. We translate S^* in $O(n)$ time into a smallest connected vertex cover S_{D^*} of G that contains that contains $(D \setminus D^*) \cup W$ but no vertex of D^* by uncontracting any contracted edges. It takes $O(n^{2s+19})$ time to find S_{D^*}.

As mentioned, in the end we pick a smallest set of the sets S_{D^*}. This set is then a smallest connected vertex cover of G that contains W. As there are $O(n^{2s^2+s+3} \cdot n^{14s^3+2})$ of such sets, each of which is found in $O(n^{2s+19})$ time, the total running time is $O(n^{19s^3+24})$. The correctness of our algorithm follows immediately from the above case analysis and the description of the cases. □

Note that the algorithm in Theorem 12 not only solves the decision problem, but also finds a minimum connected vertex cover of a given $(sP_1 + P_5)$-free graph.

4.5 Weighted Connected Vertex Cover Extension

Recall that a *vertex-weighting* for a graph $G = (V, E)$ is a function $w_V : V \to \mathbb{Q}^+$ that assign a positive rational weight to every vertex v. The *weight* of a subset $S \subseteq V$ is defined as $w_V(S) = \sum_{v \in S} w_V(v)$. A vertex cover S of G is a *minimum weight vertex cover* if G has no vertex cover S' with $w_V(S') < w_V(S)$. The Weighted Vertex Cover problem is to find a minimum weight vertex cover of a vertex-weighed graph G. As mentioned, Theorem 1 can be generalized to hold for Weighted Vertex Cover [56]. As we use Theorem 1 to prove Theorem 12, this allows us to solve the following more general problem in polynomial time for $(sP_1 + P_5)$-free graphs $(s \geq 0)$.

Weighted Connected Vertex Cover Extension
Instance: a graph G, a vertex weighting function w_V, a subset $W \subseteq V$ and an integer k
Question: does G have connected vertex cover S_W for W and $w_V(S_W) \leq k$?

In order to prove this result we first need to generalize the Connected Vertex Cover Completion problem.

Weighted Connected Vertex Cover Completion
Instance: a cover-complete triple (G, J, y) and a vertex weighting function w_V.
Task: find a minimum weight connected vertex cover S of G that contains J.

We first prove the following theorem.

Theorem 14. *For every $s \geq 0$,* Weighted Connected Vertex Cover Completion *can be solved in $O(n^{2s+19})$ for cover-complete triples (G, J, y), where G is an $(sP_1 + P_5)$-free graph.*

Proof. We can follow the same approach as in the proof of Theorem 13. We first note that Lemma 10 is a structural lemma unrelated to the vertex weight function w_V. Lemma 7 was not needed for the proof of Theorem 13 and we do not need it here either. For Lemma 16, we do not have to adjust statements 1 and 2 and only have to replace statement 3 by its weighted version. In order to do so, we define the weight of the new vertex y_w, obtained from set-contracting via a vertex w, as the sum of the weights of all the vertices in $J_w \cup \{w\}$. We can then use the same arguments. Observation 1 and Lemmas 17–18 are structural lemmas that are unrelated to the vertex weight function w, so we can still use them. We need to replace Lemma 19 by its weighted version. We can then use the same arguments; in particular, as we may replace Theorem 1 by its weighted version [56]. We can also replace Lemma 20 by its weighted version: its proof uses brute force searching, and instead of remembering and updating the smallest size of a connected vertex cover, we keep track of the smallest weight. Lemma 21 still holds in our setting as well. That is, after replacing condition 3 by its weighted version, we can still use the same arguments (modified for weights of sets instead of their sizes). The same holds for Lemma 22 (we need to replace property 2). Lemmas 23 and 24 are structural lemmas unrelated to the vertex weight function w_V, so we can still use them. Lemma 25 is algorithmic, but as this lemma is not related to vertex weight functions we can still use it. That is, any clique $K \subseteq L$ with $N_G(K) \cap J = J$ found by Lemma 25 suffices, as every (connected) vertex cover must use all but at most one vertices of a clique. Hence, for proving Theorem 14 we can use the same arguments as in the proof of Theorem 13; in particular the claim inside the proof of Theorem 13 is still valid and instead of remembering the smallest size of the vertex covers found by the algorithm so far, we remember the smallest weight. □

We are now ready to show the following result.

Theorem 15. *For every $s \geq 0$,* Weighted Connected Vertex Cover Extension *can be solved in $O(n^{19s^3+24})$-time for $(sP_1 + P_5)$-free graphs.*

Proof. Let $s \geq 0$, and let G be an $(sP_1 + P_5)$-free graph. We first recall that Lemma 10 is unrelated to the vertex weight function w_V. The same holds for Lemma 7. Hence we may still use both lemmas. In particular this implies that Lemma 9 still holds. Lemma 26 is a structural lemma that is unrelated to the vertex weight function w_V, so we can safely use it. By these observations and Theorem 14, we can now follow the same arguments as used in the proof of Theorem 12. This proof is based on brute force searching. The only thing we need to do is to remember the smallest weight of the vertex covers found during the execution of the algorithm instead of their sizes. □

4.6 Conclusions

We proved that Weighted Connected Vertex Cover Extension is polynomial-time solvable for $(sP_1 + P_5)$-free graphs for every integer $s \geq 0$. We finish this chapter by posing the following open problems.

Open Problem 9 *Determine the complexity of* Connected Vertex Cover *for P_6-free graphs.*

Open Problem 10 *Determine whether there exists an integer r such that* Connected Vertex Cover *is* NP*-complete for P_r-free graphs.*

For Open Problem 9, it might be easier to consider first the class of $(P_2 + P_3)$-free graphs, for which we do not know the complexity of Connected Vertex Cover either.

For Open Problem 10, we need a better understanding of P_r-free graphs. The Connected Vertex Cover problem belongs to a range of problems which we only know to be polynomial-time solvable on P_r-free graphs up to some value of r. These problems include, for example, Vertex Cover, Feedback Vertex Set, Connected Feedback Vertex Set, Independent Feedback Vertex Set, Independent Odd Cycle Transversal, 3-Colouring and (Dominating) Induced Matching, see [11,51] for further details. Even our understanding of bipartite P_r-free graphs is limited. For instance, we only know that Hypergraph 2-Colourability is polynomial-time solvable on P_7-free incidence graphs (which are bipartite) [25].

We conclude this section with the following conjecture.

Conjecture 1 *Let G be a P_5-free graph and S be a minimum vertex cover of G. Then either S is contained in a minimum connected vertex cover S' of G or S has the same size of S'.*

This conjecture is false for the case of $(sP_1 + P_5)$-free graphs, with $s \geq 1$; for example, see the graph G_2 in Figure 16. A proof of this conjecture would allow to increase our knowledge of the Connected Vertex Cover problem when restricted to P_5-free graphs.

5 Connected Cycle Transversal Extensions

For a graph $G = (V, E)$, a set $S_W \subseteq V$ is a *connected feedback vertex set extension* or *connected odd cycle transversal extension* for a set $W \subseteq V$ if it is a feedback vertex set or odd cycle transversal, respectively, that induces a connected subgraph and contains W. With these definitions we can formally state the corresponding transversal problems of this section.

CONNECTED FEEDBACK VERTEX SET EXTENSION
Instance: a graph $G = (V, E)$, a subset $W \subseteq V$ and a positive integer k.
Question: does G have a connected feedback vertex set S_W for W and $|S_W| \leq k$?

CONNECTED ODD CYCLE TRANSVERSAL EXTENSION
Instance: a graph $G = (V, E)$, a subset $W \subseteq V$ and a positive integer k.
Question: does G have a connected odd cycle transversal S_W for W and $|S_W| \leq k$?

Since in the case $W = \emptyset$ these connected transversal extension are equivalent to their respective connected original ones, the two problems are NP-complete [28,52], we consider the restriction of the input to hereditary graph classes in order to better understand which graph properties cause the computational hardness.

Remark 2 *Let* (G, W, k) *be an input of* CONNECTED FEEDBACK VERTEX SET EXTENSION *or* CONNECTED ODD CYCLE TRANSVERSAL EXTENSION. *Then we may assume the graph G is connected. If it is not, then either at most one connected component of G contains W and has (odd) cycles, in which case tree (bipartite) connected components do not need to be considered, or the answer is immediately no. Testing whether or not an input has an immediate no answer can be done in* $O(n + m)$*-time.*

5.1 Existing Results

We focus on proving new complexity results for CONNECTED FEEDBACK VERTEX SET and CONNECTED ODD CYCLE TRANSVERSAL on H-free graphs. As we will use algorithms for VERTEX COVER and CONNECTED VERTEX COVER restricted to H-free graphs as subroutines for our new algorithms, we include these two problems in our discussion. A list of complexity results on these problems can be found in Sections 3 and 4.

Both CONNECTED FEEDBACK VERTEX SET and CONNECTED ODD CYCLE TRANSVERSAL remain NP-complete on graphs of arbitrarily large girth and on line graphs [28] and

Grigoriev and Sitters [52] proved that Connected Feedback Vertex Set is NP-complete even on planar graphs with maximum degree 9.

A small modification of the construction by Okrasa and Rzążewski [85] proves that Connected Odd Cycle Transversal is NP-complete on P_{13}-free graphs. The complexity of Connected Feedback Vertex Set is unknown when restricted to P_r-free graphs for $r \geq 5$. For every $s \geq 1$, both connected problems are polynomial-time solvable on sP_2-free graphs [28], using the price of connectivity for feedback vertex set [7,59]. See Table 2 for an overview on the complexity results on these problems.

5.2 Our Results

In Section 5.3 with our work [35] we prove that Connected Feedback Vertex Set is polynomial-time solvable for P_4-free and $(sP_1 + P_3)$-free graphs, for every $s \geq 0$. Moreover in Section 5.4 we show that the same results hold for Connected Odd Cycle Transversal and, in addition, that this problem is NP-complete on $(P_2 + P_5, P_6)$-free graphs, like we did in Theorem 8.

	girth p	line graphs	sP_2-free	P_r-free	$sP_1 + P_r$-free
CVC	NP-c [83]	NP-c [83]	P : $s \geq 0$ [28]	P : $r \leq 5$*	P : $s \geq 0, r \leq 5$
CFVS	NP-c [28]	NP-c [28]	P : $s \geq 0$ [28]	P : $r \leq 4$	P : $s \geq 0, r \leq 3$
COCT	NP-c [28]	NP-c [28]	P : $s \geq 0$ [28]	P : $r \leq 4$	P : $s \geq 0, r \leq 3$

Table 2: The complexity of the three connected transversal problems on graphs of girth at least p for every (fixed) constant $p \geq 3$, on line graphs, and on H-free graphs for various linear forests H. Results that directly follow from other results in the table are starred while unreferenced results are ours. Note this table does not completely summarise all the results from our work and from the literature.

We will prove all our results for connected feedback vertex sets and connected odd cycle transversals for the extension version. These extension versions will serve as auxiliary problems for some of our inductive arguments, but this approach also leads to slightly stronger results.

Recall that for (Subset) Vertex Cover we have Lemma 14 that allows to prove polynomial-time solvability for (P_1+H)-free graphs when it is the case for H-free graphs. As for the cases of Feedback Vertex Set, Odd Cycle Transversal and Connected Vertex Cover, this strategy can not be used for Connected Feedback Vertex Set and Connected Odd Cycle Transversal (see Figure 16 for example).

5.3 Connected Feedback Vertex Set Extension

In this section, we will prove our polynomial time results for Connected Feedback Vertex Set Extension for P_4-free graphs in Theorem 16 and for $(sP_1 + P_3)$-free graphs in Theorem 17.

Theorem 16. Connected Feedback Vertex Set Extension *can be solved in polynomial time on P_4-free graphs.*

Proof. Let $G = (V, E)$ be a P_4-free graph on n vertices and let W be a subset of V. By Remark 2, we may assume that G is connected. By Lemma 1, in polynomial time we can find a spanning complete bipartite subgraph $G' = (X, Y, E')$, and we note that, by definition, every edge in G' is dominating. Below, in Case 1, in polynomial time we compute a smallest connected feedback vertex set of G that contains W and intersects both X and Y. In Case 2, in polynomial time we compute a smallest connected feedback vertex set of G that contains W and that is a subset of either X or Y (if such a set exists). Then the smallest set found is a smallest connected feedback vertex set of G that contains W.

Case 1: *Compute a smallest connected feedback vertex set S of G such that $W \subseteq S$, $S \cap X \neq \emptyset$ and $S \cap Y \neq \emptyset$.*

We perform Case 1 as follows. Consider two vertices $u \in X$ and $v \in Y$. We shall describe how to find a smallest connected feedback vertex set of G that contains $W \cup \{u, v\}$. We find a smallest feedback vertex set S' in $G - (W \cup \{u, v\})$. As $G - (W \cup \{u, v\})$ is P_4-free, this takes polynomial time by the result of [1]. Then $S' \cup W \cup \{u, v\}$ is a smallest feedback vertex set of G that contains $W \cup \{u, v\}$ and is connected, since uv is a dominating edge. By repeating this polynomial-time procedure for all $O(n^2)$ possible choices of u and v, we will find S in polynomial time.

Case 2: *Compute a smallest connected feedback vertex set S of G such that $W \subseteq S$ and $S \subseteq X$ or $S \subseteq Y$.*

For Case 2 we describe only the $S \subseteq X$ case, as the $S \subseteq Y$ case is symmetric. Thus we may assume that $W \subseteq X$, otherwise no such set exists. Clearly, we may also assume that $G[Y]$ contains no cycles. If $G[Y]$ contains an edge it follows that $S = X$, otherwise $G - S$ would contain a triangle. Suppose instead that Y is an independent set. If $|Y| = 1$, then $X \setminus S$ must be an independent set, otherwise $G - S$ contains a triangle. So S is a smallest connected vertex cover of $G[X]$ that contains W. As $G[X]$ is P_4-free, we can find such an S in polynomial time by Theorem 12. If $|Y| \geq 2$, then $|X \setminus S| \leq 1$, as otherwise $G - S$ contains a 4-cycle. Thus, we check, in polynomial time, if there exists a vertex $x \in X \setminus W$, such that $X \setminus \{x\}$ is connected. If so, $S = X \setminus \{x\}$. □

Before stating the main result of this section, let us recall the function b on non-negative integers by $b(s) := \max\{3, 2s - 1\}$ used for Lemma 5.

Theorem 17. *For every $s \geq 0$,* Connected Feedback Vertex Set Extension *can be solved in polynomial time on $(sP_1 + P_3)$-free graphs.*

Proof. There are similarities to the proof of Theorem 7, but different arguments are needed. Let $s \geq 0$ be an integer, let $G = (V, E)$ be an $(sP_1 + P_3)$-free graph and let W be a subset of V. By Remark 2, we may assume that G is connected. We must show how to find a smallest connected feedback vertex set of G that contains W in polynomial time. We show how to solve the complementary problem in polynomial time: how to find a largest induced forest F of G that does not include any vertex of W and $V \setminus F$ is connected. We will say that an induced forest F is *good* if it has these two properties.

Our algorithm computes the following three cases in polynomial time. Together, these three cases cover all possibilities.

Case 1: *Compute a largest good induced forest F such that there is a connected component of F that has at least $b(s)$ vertices.*

By Lemma 5 we know that F has exactly one connected component on at least $b(s)$ vertices and there are at most $s - 1$ other connected components of F, each on at most two vertices. By Lemma 4, the connected component on at least $b(s)$ vertices has at most $4s$ internal vertices. We consider $O(n^{4s+2(s-1)})$ choices of a non-empty set U of at most $4s$ vertices that induces a tree and a set U' of at most $2(s-1)$ vertices that induces a disjoint union of vertices and edges such that $U \cup U'$ does not intersect W, U is disjoint from U' and no vertex of U has a neighbour in U'. Let R be the set of vertices that each have exactly one neighbour in U and no neighbour in U', but do not belong to W. We then add to $U \cup U'$ the largest possible set L of vertices that are independent and belong to the set R such that $G - (L \cup U \cup U')$ is connected. This is achieved by taking the complement of the smallest connected vertex cover of $G - (U \cup U')$ that contains $V \setminus (R \cup U \cup U')$. By Theorem 12, this can be done in polynomial time.

Case 2: *Compute a largest good induced forest F such that F has at most $s-1$ connected components and each connected component has at most $b(s) - 1$ vertices.*

Since the number of vertices in F is bounded by the constant $(s - 1)(b(s) - 1)$, we can simply check all sets containing at most that many vertices to see if they induce such a good forest.

Case 3: *Compute a largest good induced forest F such that F has at least s connected components and each connected component has at most $b(s) - 1$ vertices.*

We consider $O(n^{s(b(s)-1)})$ choices of a non-empty set L of at most $s(b(s)-1)$ vertices. We reject L unless $G[L]$ is a good induced forest on s connected components with no connected component of more than $b(s)-1$ vertices. Assuming our choice of L is correct, the connected components of $G[L]$ will become connected components of $G[F]$.

Let $U = N(L)$ and note that no vertex of U is in F. If $G - U$ is a good forest, then we are done. Otherwise we consider every set R of at most $2s^2 - 2s + 3$ vertices of $G - (L \cup U \cup W)$ such that $G[R \cup U \cup W]$ is connected; see also Figure 19. We note that if there is a largest induced forest F such that the connected components of $G[L]$ are also connected components of $G[F]$, then Lemma 6 applied to $G - F$ implies that such a set R exists.

Let $S = R \cup U \cup W$. If $G - S$ is a forest, then we are done. Otherwise note that $G - (L \cup S)$ is the disjoint union of one or more complete graphs: $G - (L \cup S)$ cannot contain an induced P_3, as it is anti-complete to L which contains an induced sP_1.

As G is connected, each of the complete graphs in $G - (L \cup S)$ contains at least one vertex that is adjacent to some vertex of S. Hence in polynomial time we can find a set S' of vertices containing all but $\min\{2, |X|\}$ vertices from each of the complete graphs X in such a way that $G[S \cup S']$ is connected. Then $G - (S \cup S')$ is a largest good induced forest that contains L and no vertex of $R \cup U$.

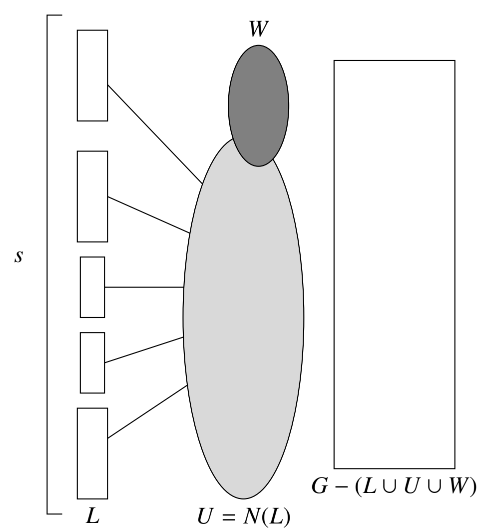


Fig. 19: The decomposition of the $(sP_1 + P_3)$-free graph G, as given in Case 3 of the algorithm from the proof of Theorem 17.

After considering each of the $O(n^{2s^2-2s+3})$ choices for R, in polynomial time we find a largest good induced forest that contains L and no vertex of U. After considering each of the $O(n^{s(b(s)-1)})$ choices for L, we find in polynomial time a largest good induced forest that has at least s connected components, each with at most $b(s)-1$ vertices. □

5.4 Connected Odd Cycle Transversal Extension

In this section, we will prove that Connected Odd Cycle Transversal Extension can be solved in polynomial time for P_4-free graphs in Theorem 18 and for $(sP_1 + P_3)$-free graphs in Theorem 19. Finally in Theorem 20, we show that Connected Odd Cycle Transversal Extension is NP-complete on $(P_2 + P_5, P_6)$-free graphs.

Theorem 18. Connected Odd Cycle Transversal Extension *can be solved in polynomial time on P_4-free graphs.*

Proof. We only provide an outline, as the proof follows that of Theorem 16. We consider the same two cases. In Case 1, we need to find a smallest odd cycle transversal S' in $G - (W \cup \{u, v\})$ and can use the result of [13]. In Case 2, we again note that if $G[Y]$ contains an edge, then $S = X$. Suppose that Y is an independent set. Then $G - S$ contains no odd cycles if and only if $X \setminus S$ is independent, so S is a smallest connected vertex cover of $G[X]$ that contains W. (That is, the $|Y| = 1$ case from the proof of Theorem 16 can be used for all values of $|Y|$, as we are no longer concerned with whether $G - S$ might contain cycles of even length.) □

Before stating this important result of the section, let us recall the function b on non-negative integers by $b(s) := \max\{3, 2s - 1\}$ used for Lemma 5.

Theorem 19. *For every $s \geq 0$,* Connected Odd Cycle Transversal Extension *can be solved in polynomial time on $(sP_1 + P_3)$-free graphs.*

Proof. Let $s \geq 0$ be an integer, let $G = (V, E)$ be an $(sP_1 + P_3)$-free graph and let W be a subset of V. By Remark 2, we may assume that G is connected. We must describe how to find a smallest connected odd cycle transversal of G that contains W. We will solve the complementary problem: how to find a largest induced bipartite graph of G that does not include any vertex of W and whose complement is connected. We will say that an induced bipartite graph B is *good* if it has these two properties. Our algorithm consists of three cases, which can each be performed in polynomial time and which together cover all the possible cases.

Case 1: *Compute a largest good induced bipartite subgraph B such that B has a bipartition $\{X, Y\}$ in which one set, say X, has size $|X| \leq s$.*

We consider $O(n^s)$ choices of an independent set X of at most s vertices of G that does not intersect W. We wish to find Y, the largest possible independent set in $G - (W \cup X)$ such that $G - (X \cup Y)$ is connected. By Theorem 12, we can do this in polynomial time by computing a minimum connected vertex cover of $G - X$ that contains W and taking its complement (in $G - X$).

Case 2: *Compute a largest good induced bipartite subgraph B such that B has at least s connected components and each connected component has at most two vertices.*

Note that $2 \leq b(s) - 1$. The algorithm mimics Case 3 of the algorithm in the proof of Theorem 17, but checks for a good bipartite graph instead of a good forest.

Case 3: *Compute a largest good induced bipartite subgraph B such that there is a connected component of B that has at least three vertices and B has a bipartition $\{X, Y\}$ with $|X| \geq s + 1$ and $|Y| \geq s + 1$.*

It is in this case that we must do most of the work in proving the theorem, and here we will need ideas beyond those already met in this section.

As B contains a connected component on at least three vertices, it will contain an induced P_3 and so $|X| \geq 1$ and $|Y| \geq 1$. We consider $O(n^{2s+2})$ choices of disjoint independent sets X' and Y' that each contain $s + 1$ vertices of G and do not intersect W. If $G[X' \cup Y']$ contains an induced P_3, our aim is to compute a largest good induced bipartite graph B with bipartition $\{X, Y\}$ such that $X' \subseteq X$ and $Y' \subseteq Y$; otherwise we discard the choice of X', Y'.

We define (see also Figure 20) a partition of $V \setminus (X' \cup Y')$:

$$\begin{aligned}
U &= (N(X') \cap N(Y')) \cup W \\
V_X &= N(X') \setminus (Y' \cup N(Y') \cup W) \\
V_Y &= N(Y') \setminus (X' \cup N(X') \cup W) \\
Z &= V \setminus (X' \cup Y' \cup N(X') \cup N(Y') \cup W)
\end{aligned}$$

There are a number of steps where our procedure branches as we consider all possible ways of choosing whether or not to add certain vertices to B. Note that assuming our choice of X' and Y' is correct, no vertex of U can be in B. If we decide that a vertex will not be in B, we will then add it to U.

Step 1: *Reduce Z to the empty set.*

Notice that Z does not contain an independent set on more than $s - 1$ vertices otherwise $G[X' \cup Y' \cup Z]$ would contain an induced $sP_1 + P_3$. We consider $O(n^{2s-2})$ choices of

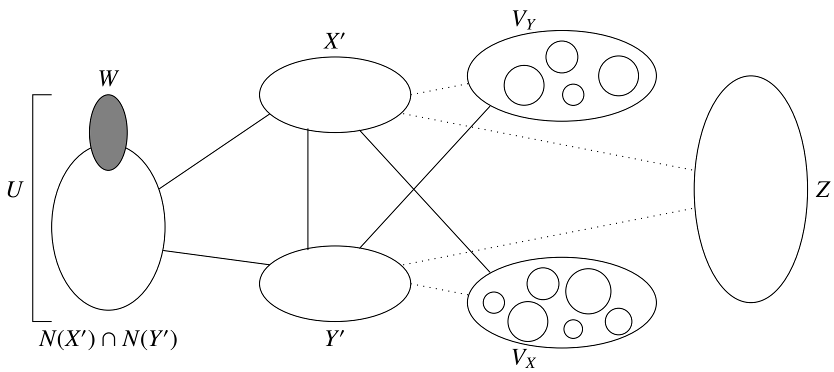


Fig. 20: The decomposition of G in Case 3. Full and dotted lines indicate when two sets are complete or anti-complete to each other, respectively. The absence of a full or dotted lines indicates that edges may or may not exist between two sets. The circles in V_X and V_Y represent disjoint unions of complete graphs.

disjoint independent sets Z_X and Z_Y that are each subsets of Z and each contain at most $s-1$ vertices. We move the vertices of Z_X and Z_Y by adding them to X' and Y', respectively. We move the vertices of $Z \setminus (Z_X \cup Z_Y)$ by adding them to U. If after this process is complete there are vertices in $V_X \cup V_Y$ with neighbours in both X' and Y', we move these vertices by adding them to U. We note that now:

- Z is the empty set,
- V_X still contains vertices with neighbours in X' but not in Y',
- V_Y still contains vertices with neighbours in Y' but not in X', and
- U contains vertices that will not be in B.

So our task is to decide how best to add vertices of V_X to Y' and vertices of V_Y to X', but first there is another step: as $G - B$ must be connected, and $G[U]$ is a subgraph of $G - B$, we choose some vertices that will not be in B, but will connect together the connected components of $G[U]$. This will not be possible if the vertices of U belong to more than one connected component of $G - (X' \cup Y')$. Hence, in that case we discard this choice of Z_X, Z_Y.

Step 2: *Make $G[U]$ connected.*

We consider $O(n^{2s^2-2s+3})$ choices of set R of vertices of $G - (X' \cup Y')$ such that each contains at most $2s^2 - 2s + 3$ vertices. If $G[R \cup U]$ is connected, we move the vertices of R by adding them to U, and so $G[U]$ becomes connected. Note that since all vertices of U are in the same connected component of $G - (X' \cup Y')$, Lemma 6 implies that at least one such set R can be found.

Step 3: *Add vertices from V_X to Y' and from V_Y to X'.*

We note that $G[V_X]$ is P_3-free, as no vertex of V_X has a neighbour in Y', $|Y'| \geq s$, and G is $(sP_1 + P_3)$-free. By symmetry, $G[V_Y]$ is P_3-free. Thus both $G[V_X]$ and $G[V_Y]$ are disjoint unions of complete graphs. Note that B can contain at most one vertex from each of these complete graphs. We consider two subcases.

Case 3a: *Compute a largest good induced bipartite subgraph B with bipartition $\{X, Y\}$ such that $X' \subseteq X$, $Y' \subseteq Y$ and $G - B$ contains no edges between V_X and V_Y.*

As $G - B$ must be connected, each clique of V_X and V_Y that contains at least two vertices must contain a vertex adjacent to U (otherwise such a set B cannot exist). Thus we can form X from X' by adding to X' one vertex from each clique of V_Y and form Y by adding to Y' one vertex from each clique of V_X in such a way that $G - B$ is connected. (If we do this, it is possible that $G - B$ will contain an edge from V_X to V_Y, but then this solution is at least as large as one where such edges are avoided.)

Case 3b: *Compute a largest good induced bipartite subgraph B with bipartition $\{X, Y\}$ such that $X' \subseteq X$, $Y' \subseteq Y$ and $G - B$ has an edge xy where $x \in V_X$, $y \in V_Y$.*

We consider $O(n^2)$ choices of an edge xy, $x \in V_X$, $y \in V_Y$. Let $v_X \in X'$ be a neighbour of x and note that v_X, x and y induce a P_3 in G. Therefore, since G is $(sP_1 + P_3)$-free, x must be complete to all but at most $s-1$ cliques of V_Y. By symmetry, y must be complete to all but at most $s - 1$ cliques of V_X. A clique in V_X or V_Y is *bad* if it is not complete to y or x, respectively. Note that the cliques containing x and y may be bad. We move x and y to U.

We consider $O(n^{2s-2})$ choices of a set S of at most $2s - 2$ vertices that each belong to a distinct bad clique and move each to X' or Y' if they are in V_Y or V_X respectively. We move the other vertices of the bad cliques to U. If the vertices of U are not in the same connected component of $G - (X' \cup Y')$, we discard this choice of S. We consider $O(n^{2s^2-2s+3})$ choices of sets R' of vertices of $G-(X' \cup Y')$ such that each contains at most $2s^2 - 2s + 3$ vertices. If $G[R' \cup U]$ is connected we move the vertices of R' to U, so $G[U]$ becomes connected. Since the vertices of U are in the same connected component of $G - (X' \cup Y')$, Lemma 6 implies that at least one such set R' can be found.

Note that some cliques might have been completely removed from V_X and V_Y by the choice of R'. It only remains to pick one vertex from each remaining clique of V_X and V_Y, and add these vertices to Y' or X', respectively to finally obtain B. As all vertices in these cliques are adjacent to x or y we know that $G - B$ will be connected. □

Noting that the odd cycle transversal S in the proof of Theorem 8 is connected, is enough to prove the following result.

Theorem 20. Connected Odd Cycle Transversal *is* NP*-complete on* $(P_2 + P_5, P_6)$*-free graphs.*

5.5 Steiner Tree Problem and Graph Transversals

Let $G = (V, E)$ be a graph and $W \subseteq V$ be a set of vertices, a *Steiner tree* for W of G is a tree T_W of G that contains W. Now we can formally define two decision problems.

Edge Steiner Tree
Instance: a graph $G = (V, E)$, an edge-weighting function w_E, a subset $W \subseteq V$ of terminals and a positive integer k.
Question: does G have a Steiner tree T_W for W with $w_E(T_W) \leq k$?

Edge Steiner Tree is often known simply as Steiner Tree, but we wish to distinguish it from a closely related problem. The following problem is sometimes known as Node-Weighted Steiner Tree.

Vertex Steiner Tree
Instance: a connected graph $G = (V, E)$, a vertex-weighting function w_V, a subset $W \subseteq V$ and a positive integer k.
Question: does G have a Steiner tree T_W for W with $w_V(T_W) \leq k$?

We say that an instance of a problem is *unweighted* if the weighting is constant. Note that Edge Steiner Tree is a generalization of the Spanning Tree problem (set $W = V$). We refer to the textbooks of Du and Hu [38] and Prömel and Steger [91] for further background information on Steiner trees.

Remark 3 *Let* (G, W, k) *be an input of* Steiner Tree*. Then we may assume the graph* G *is connected. If it is not, then either* W *is contained in at most one connected component of* G*, in which case other connected components can be ignored, or the answer is immediately no. Testing whether or not an input has an immediate no answer can be done in* $O(n + m)$*-time.*

Edge Steiner Tree has been known to be NP-complete [67]. Now we need to give a number of results for Steiner Tree that are going to be used to prove the main result of the section.

Theorem 21. *Unweighted* Vertex Steiner Tree *is* NP*-complete for line graphs.*

Proof. First note that unweighted Edge Steiner Tree is NP-complete (see [48] for example). Let (G, W, k) be an instance of this problem. From G we construct a new graph G' by introducing a new vertex v_u for each terminal $u \in W$, which we make only adjacent to u. We let W' consist of all these new vertices. We observe that G' has a Steiner tree T' for W' with at most $k + |W|$ edges if and only if G has a Steiner tree T for W with at most k edges.

We now consider the line graph $L(G')$ with set of terminals $W^* = \{uv_u \mid u \in U\}$; this is a set of edges in G' and a set of vertices in $L(G')$. To complete the proof, we show that G' has a Steiner tree for W' on, say, ℓ edges if and only if $L(G')$ has a Steiner tree for W^* on ℓ vertices. We first note that the edge set E' of a Steiner tree for W' of G' must contain the set W^*. Further, E', considered as a set of vertices of $L(G')$, induces a connected subgraph and has $|E'| = \ell$ vertices. Conversely, if there is a Steiner tree for W^* in $L(G')$ on ℓ vertices, then these vertices, considered as edges in G', form a Steiner tree for W' in G'. □

Before we are able to prove our main result regarding Steiner trees, we need the following result.

Theorem 22. *For every $s \geq 0$,* Vertex Steiner Tree *can be solved in time $O(n^{2s^2-s+5})$ for connected $(sP_1 + P_4)$-free graphs on n vertices.*

Proof. Let $s \geq 0$ be an integer. Let $G = (V, E)$ be an $(sP_1 + P_4)$-free graph with a vertex weighting $w_V : V \to \mathbb{Q}^+$ and set of terminals W. By Remark 3, we may assume that G is connected. We show how to solve the optimization version of Vertex Steiner Tree on G. Let $R \subseteq V \setminus W$ be such that $G[W \cup R]$ is connected and, subject to this condition, $W \cup R$ has minimum weight $w_V(U \cup R)$. Thus any spanning tree of $G[W \cup R]$ is an optimal solution. Let us consider the possible size of R.

First suppose that $G[W \cup R]$ is P_4-free. Then, by Lemma 1, $G[W \cup R]$ has a spanning complete bipartite subgraph. That is, there is a bipartition (A, B) of $W \cup R$ such that every vertex in A is joined to every vertex in B. We may assume without loss of generality that $|W| \geq 2$. Then $|W \cup R| \geq 2$, and thus neither A nor B is the empty set. If W intersects both A and B, then $G[W]$ is connected and $|R| = 0$. So let us assume that $W \subseteq A$, and so $R \supseteq B$. Then $R \cap A = \emptyset$ since $G[W \cup B]$ is connected. As we know that every vertex in $A = W$ is joined to every vertex in $B = R$, we find that $|R| = 1$.

Suppose instead that $G[W \cup R]$ contains an induced path P on four vertices. We call the connected components of $G[W]$ *bad* if they do not intersect P or the neighbours of P in G. There are at most $s - 1$ bad connected components; else, G contains an $sP_1 + P_4$. Let W^* be a subset of U that includes one vertex from each of these bad connected components. Then each vertex of $G[W \cup R]$ belongs either to W or P or is an internal vertex of a shortest path in $G[W \cup R]$ from P to a vertex of W^*. The number of internal vertices in such a shortest path is at most $2s + 1$; else, the path contains an induced $sP_1 + P_4$. As R is a subset of $V(P)$ and these internal vertices, we find that $|R| \leq 4 + (2s + 1)(s - 1) = 2s^2 - s + 3$.

So in all cases R contains at most $2s^2 - s + 3$ vertices and our algorithm is just to consider every such set R and check, in each case, whether $G[W \cup R]$ is connected. Our solution is one with minimum weight that satisfies the connectivity constraint. As there are $O(n^{2s^2-s+3})$ sets to consider, and checking connectivity takes $O(n^2)$ time, the algorithm requires $O(n^{2s^2-s+5})$ time. □

We are finally ready to prove the following complete dichotomy.

Theorem 23. *Let H be a graph. If H is an induced subgraph of $sP_1 + P_4$ for some $s \geq 0$, then* Vertex Steiner Tree *is polynomial-time solvable for H-free graphs, otherwise even unweighted* Vertex Steiner Tree *is* NP*-complete.*

Proof. If H has a cycle then, due to the results on chordal bipartite graphs [16] and on split graphs [100], the problem is NP-complete. Hence, we may assume that H has no cycle, so H is a forest. If H contains a vertex of degree at least 3, then the class of H-free graphs contains the class of claw-free graphs, which in turn contains the class of line graph. Hence, we can apply Theorem 21. Thus we may assume that H is a linear forest. If H contains a connected component with at least five vertices or two non-trivial connected components, then the class of H-free graphs contains the class of $2P_2$-free graphs and so we can apply the NP-completeness result on split graphs [100]. It remains to consider the case where H is an induced subgraph of $sP_1 + P_4$, for which we can apply Theorem 22. □

Now we can prove the following result.

Theorem 24. *For any graph set $\mathcal{H}$, there is a polynomial-time reduction of* Weighted Connected $\mathcal{H}$-Transversal Extension *to* Vertex Steiner Tree *whenever in the input (G, w_V, W, k), the set W is an $\mathcal{H}$-transversal of G.*

Proof. Let (G, w_V, W, k) be an input of Weighted Connected $\mathcal{H}$-Transversal Extension and assume W is an $\mathcal{H}$-transversal of G. We claim that if G has a Steiner tree for the set W of vertex-weight at most k then G has a connected $\mathcal{H}$-transversal that contains W with vertex-weight at most k.

Indeed let T_W be a Steiner tree for W with $w_V(T_W) \leq k$. By definition T_W induces a connected subgraph and contains W, moreover W is $\mathcal{H}$-transversal by assumption and $w_V(T_W) \leq k$. □

The following result is the direct application of Theorem 23 to Theorem 24.

Corollary 1. *For any graph set $\mathcal{H}$ and every integer $s \geq 0$,* Weighted Connected $\mathcal{H}$-Transversal Extension *can be solved in polynomial time for inputs (G, w_V, W, k), where G is an $(sP_1 + P_4)$-free graph and W is an $\mathcal{H}$-transversal of G.*

5.6 Conclusions

We proved polynomial-time solvability of Connected Feedback Vertex Set Extension and Connected Odd Cycle Transversal Extension on H-free graphs, when $H = P_4$ or $H = sP_1 + P_3$; see also Table 2, where we place these results in the context of known results for these problems on H-free graphs. We also showed that Connected Odd Cycle Transversal is NP-complete on $(P_2 + P_5, P_6)$-free graphs.

Natural cases for future work are the cases when $H = sP_1 + P_4$ for $s \geq 1$ and $H = P_5$ for all four problems (in particular the case when $H = P_5$ is the only open case for Odd Cycle Transversal and Connected Odd Cycle Transversal restricted to P_r-free graphs).

Open Problem 11 *Determine the complexity of* Connected Odd Cycle Transversal *for $(sP_1 + P_4)$-free graphs.*

Open Problem 12 *Determine the complexity of* Connected Odd Cycle Transversal *for P_5-free graphs.*

One of the main obstacles to solving Open Problem 11 is that Lemma 5 does not hold on $(sP_1 + P_4)$-free graphs: the disjoint union of any number of arbitrarily large stars is even P_4-free.

Recall that Vertex Cover and Connected Vertex Cover are polynomial-time solvable even on $(sP_1 + P_6)$-free graphs by Theorem 1 and $(sP_1 + P_5)$-free graphs by Theorem 12, respectively, for every $s \geq 0$. In contrast to the case for Odd Cycle Transversal and Connected Odd Cycle Transversal, it is not known whether there is an integer r for which any of the problems Vertex Cover, Feedback Vertex Set or their connected variants is NP-complete on P_r-free graphs. Determining whether such an r exists is an interesting research question which has been collected in Open Problems 2, 6, 10 and in the following one.

Open Problem 13 *Determine whether these exists an integer r such that* Connected Feedback Vertex Set *is* NP*-complete for P_r-free graphs.*

We note that a similar complexity study has also been undertaken for the independent variants of the problems Feedback Vertex Set and Odd Cycle Transversal, while Independent Vertex Cover is polynomial-time solvable. In particular, Independent Feedback Vertex Set and Independent Odd Cycle Transversal are polynomial-time solvable on P_5-free graphs [12], but their complexity status is unknown on P_6-free graphs. It is not known whether there is an integer r such that Independent Feedback Vertex Set or Independent Odd Cycle Transversal is NP-complete on P_r-free graphs.

We conclude that in order to make any further progress, we must better understand the structure of P_r-free graphs. This topic has been well studied in recent years, see also for example [51,53]. However, more research and new approaches will be needed.

6 Independent Transversals

For each studied transversal we have introduced a vast research literature and developed original work regarding the computational complexity of the respective transversal problems.

Recall that, for a graph $G = (V, E)$, a transversal is *independent* if every two vertices are non-adjacent. In this section we are interested in the following research question:

> *How is the minimum size of a transversal in a graph affected by adding the requirement that the transversal is independent?*

Of course, this question can be interpreted in many ways. In this section, we focus on the following: is the size of a smallest possible independent transversal (assuming one exists) bounded in terms of the minimum size of a transversal? That is, one might say, what is the *price of independence*?

6.1 Existing Results

To the best of our knowledge, the term price of independence was first used by Camby [20] in a recent unpublished manuscript for dominating sets. As she acknowledged, though first to coin the term, she was building on past work. In fact, Camby and her co-author Plein had given a forbidden induced subgraph characterization of those graphs G for which, for every induced subgraph of G, there are minimum size dominating sets that are already independent [23], and there are a number of further papers on the topic of the price of independence for dominating sets (see the discussion in [20]).

We observe that this incipient work on the price of independence is a natural companion to recent work on the *price of connectivity*, investigating the relationship between minimum size transversals and minimum size connected transversals. This work began with the work of Cardinal and Levy in their paper [26] and has since been taken in several directions; see, for example, [7,21,22,24,28,52,59].

Some results for the price of connectivity have some algorithmic consequences for the connected transversal problems. We want to understand if a study for the price of independence could have similar consequences. Until now there is no clear indication for a positive development in this direction: the difference between the minimum size of a transversal in a graph and the minimum size of an independent transversal in the graph can become unbounded quickly.

6.2 Our Results

In this section, as we broaden the study of the price of independence by investigating vertex cover, feedback vertex set and odd cycle transversal. We will concentrate on classes of graphs defined by a single forbidden induced subgraph H, just as was done for the price of connectivity [7,59]. That is, for a graph H, we ask what, for a given type of transversal, is the price of independence in the class of H-free graphs? The ultimate aim in each case is to find a dichotomy that allows us to say, given H, whether or not the size of a minimum size independent transversal can be bounded in terms of the size of a minimum transversal.

The Price of Independence for Vertex Cover. A graph has an independent vertex cover if and only if it is bipartite. For a bipartite graph G, let $\mathrm{vc}(G)$ denote the size of a minimum vertex cover, and let $\mathrm{ivc}(G)$ denote the size of a minimum independent vertex cover. Let $\mathcal{X}$ be a class of bipartite graphs. Then $\mathcal{X}$ is ivc–*bounded* if there exists a function $f : \mathbb{N} \to \mathbb{N}$ such that $\mathrm{ivc}(G) \leq f(\mathrm{vc}(G))$ for every $G \in \mathcal{X}$, and $\mathcal{X}$ is ivc–*unbounded* if no such function exists, that is, if there is a k such that for every $s \geq 0$ there is a graph G in $\mathcal{X}$ with $\mathrm{vc}(G) \leq k$, but $\mathrm{ivc}(G) \geq s$. Moreover, $\mathcal{X}$ is ivc-*identical* if $\mathrm{ivc}(G) = \mathrm{vc}(G)$ for every $G \in \mathcal{X}$.

In our first two results, proven in Section 6.3, we determine for every graph H, whether or not the class of H-free bipartite graphs is ivc–bounded or ivc–identical, respectively.

Theorem 25. *Let H be a graph. The class of H-free bipartite graphs is* ivc*–bounded if and only if H is an induced subgraph of $K_{1,r} + rP_1$ or $K_{1,r}^{+}$ for some $r \geq 1$.*

Theorem 26. *Let H be a graph. The class of H-free bipartite graphs is* ivc*–identical if and only if H is an induced subgraph of $K_{1,3}^{+}$ or $2P_1 + P_3$.*

The Price of Independence for Feedback Vertex Set. A graph has an independent feedback vertex set if and only if its vertex set can be partitioned into an independent set and a set of vertices that induces a forest; graphs that have such a partition are said to be *near-bipartite*. For a near-bipartite graph G, let $\mathrm{fvs}(G)$ denote the size of a minimum feedback vertex set, and let $\mathrm{ifvs}(G)$ denote the size of a minimum independent feedback vertex set. Given a class $\mathcal{X}$ of near-bipartite graphs, we say that $\mathcal{X}$ is ifvs-*bounded* if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that $\mathrm{ifvs}(G) \leq f(\mathrm{fvs}(G))$ for every $G \in \mathcal{X}$ and ifvs-*unbounded* otherwise. Moreover, a class $\mathcal{X}$ of near-bipartite graphs is ifvs-*identical* if $\mathrm{ifvs}(G) = \mathrm{fvs}(G)$ for every $G \in \mathcal{X}$.

In our next two results, proven in Section 6.4, we almost completely determine for every graph H, whether or not the class of H-free near-bipartite graphs is ifvs-bounded or ifvs-identical, respectively; the only open case left is determining whether the class of $K_{1,3}$-free near-bipartite graphs is ifvs-identical.

Theorem 27. *Let H be a graph. The class of H-free near-bipartite graphs is* ifvs-*bounded if and only if H is isomorphic to $P_1 + P_2$, a star or an edgeless graph.*

Theorem 28. *Let H be a graph different from $K_{1,3}$. The class of H-free near-bipartite graphs is* ifvs-*identical if and only if H is a (not necessarily induced) subgraph of P_3.*

The Price of Independence for Odd Cycle Transversal. A graph has an independent odd cycle transversal S if and only if it has a 3-colouring, since, by definition, we are requesting that S is an independent set of G such that $G - S$ has a 2-colouring. For a 3-colourable graph G, let oct(G) denote the size of a minimum odd cycle transversal, and let ioct(G) denote the size of a minimum independent odd cycle transversal. Given a class $\mathcal{X}$ of 3-colourable graphs, we say that $\mathcal{X}$ is ioct-*bounded* if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that $\text{ioct}(G) \leq f(\text{oct}(G))$ for every $G \in \mathcal{X}$ and ioct-*unbounded* otherwise. Moreover, a class $\mathcal{X}$ of 3-colourable graphs is ioct-*identical* if $\text{ioct}(G) = \text{oct}(G)$ for every graph $G \in \mathcal{X}$.

In our final two results, proven in Section 6.5, we address the question of whether or not, for a graph H, the class of H-free 3-colourable graphs is ioct-bounded or ioct-identical, respectively. Here, we do not have complete dichotomies. For the former question, we prove that the number of non-equivalent open cases left is three, namely the cases when $H \in \{K_{1,4}, K_{1,3}^+, K_{1,4}^+\}$. Note that for the latter question there are also three missing cases.

Theorem 29. *Let H be a graph. The class of H-free 3-colourable graphs is* ioct-*bounded:*

- *if H is an induced subgraph of P_4 or $K_{1,3} + sP_1$ for some $s \geq 0$ and*
- *only if H is an induced subgraph of $K_{1,4}^+$ or $K_{1,4} + sP_1$ for some $s \geq 0$.*

Theorem 30. *Let H be a graph such that $H \notin \{K_{1,3}, K_{1,3}^+, 2P_1 + P_3\}$. The class of H-free 3-colourable graphs is* ioct-*identical if and only if H is a (not necessarily induced) subgraph of P_4 that is not isomorphic to $2P_2$.*

6.3 Vertex Cover

In this section we prove Theorems 25 and 26 as part of a more general theorem. We start with a useful lemma.

Lemma 27. *Let $r, s \geq 1$. If G is a $(K_{1,r} + sP_1)$-free bipartite graph with bipartition (X, Y) such that $|X|, |Y| \geq rs + r - 1$, then either:*

- *every vertex of G has degree less than r or*
- *fewer than s vertices of X have more than $s - 1$ non-neighbours in Y and fewer than s vertices of Y have more than $s - 1$ non-neighbours in X.*

Proof. Let G be a $(K_{1,r} + sP_1)$-free bipartite graph with bipartition (X, Y) such that $|X|, |Y| \geq rs + r - 1$. No vertex in X can have both r neighbours and s non-neighbours in Y, otherwise G would contain an induced $K_{1,r} + sP_1$. Therefore every vertex in X has degree either at most $r - 1$ or at least $|Y| - (s - 1) \geq rs + r - s$. By symmetry, we may assume that there is a vertex $x \in X$ of degree at least r. Suppose, for contradiction, that there is a set $X' \subseteq X$ of s vertices, each of which has more than $s-1$ non-neighbours in Y. Then every vertex of X' has degree at most $r-1$. Since $\deg(x) \geq rs+r-s = s(r-1)+r$, there must be a set $Y' \subseteq N(x)$ of r neighbours of x that have no neighbours in X'. Then $G[\{x\} \cup Y' \cup X']$ is a $K_{1,r} + sP_1$, a contradiction. It follows that fewer than s vertices in X have more than $s - 1$ non-neighbours in Y. Since $|X| \geq r + (s - 1)$, there is a set $X'' \subsetneq X$ of r vertices, each of which has at most $s - 1$ non-neighbours in Y. Since $|Y| > r(s - 1)$, there must be a vertex $y \in Y$ that is complete to X'', and therefore has $\deg(y) \geq r$. Repeating the above argument, it follows that fewer than s vertices of Y have more than $s - 1$ non-neighbours in X. This completes the proof. □

We recall that a graph has an independent vertex cover if and only if it is bipartite, and we prove two more lemmas.

Lemma 28. *Let $r, s \geq 1$. If G is a $(K_{1,r} + sP_1)$-free bipartite graph, then* $\text{ivc}(G) \leq r \cdot \text{vc}(G) + rs$.

Proof. Let G be a $(K_{1,r} + sP_1)$-free bipartite graph. Fix a bipartition (X, Y) of G. Let S be a minimum vertex cover of G, so $|S| = \text{vc}(G)$. We may assume that $\text{vc}(G) \geq 2$, otherwise $\text{ivc}(G) = \text{vc}(G)$, in which case we are done. We may also assume that $|X|, |Y| > \text{vc}(G)r + rs > rs + r - 1$, otherwise X or Y is an independent vertex cover of the required size, and we are done. If every vertex of G has degree at most $r - 1$, then $S' = (S \cap Y) \cup (N(S \cap X))$ is an independent vertex cover in G of size at most $\text{vc}(G)(r-1)$, and we are done. By Lemma 27, we may therefore assume that fewer than s vertices of X have more than $s-1$ non-neighbours in Y. We will show that this leads to a contradiction. Since $|X|, |Y| \geq \text{vc}(G) + s$, there must be a set S' of $\text{vc}(G)+1$ vertices in X that each have at least $\text{vc}(G) + 1$ neighbours in Y. If a vertex $x \in V(G)$ has degree at least $\text{vc}(G) + 1$, then $|N(x)| > |S|$, so $x \in S$. Therefore every vertex of S' must be in S, contradicting the fact that $|S'| = \text{vc}(G) + 1 > \text{vc}(G) = |S|$. □

Lemma 29. *Let $r \geq 2$. If G is a $K_{1,r}^+$-free bipartite graph, then* $\mathrm{ivc}(G) \leq (r-1)(\mathrm{vc}(G))^2$.

Proof. Clearly it is sufficient to prove the lemma for connected graphs G. Let G be a connected $K_{1,r}^+$-free bipartite graph. Fix a bipartition (X, Y) of G. Let S be a minimum vertex cover of G, so $|S| = \mathrm{vc}(G)$. We may assume that $\mathrm{vc}(G) \geq 2$, otherwise $\mathrm{ivc}(G) = \mathrm{vc}(G)$ and we are done. We may also assume that $|X|, |Y| > (\mathrm{vc}(G))^2(r-1)$, otherwise X or Y is an independent vertex cover of the required size.

If there are two vertices $x, y \in X$ with $\mathrm{dist}(x, y) = 2$ and $\deg(x) \geq \deg(y) + (r-1)$, then x, y, a common neighbour of x and y, and $r-1$ vertices from $N(x) \setminus N(y)$ would induce a $K_{1,r}^+$ in G, a contradiction. Therefore, if $x, y \in X$ with $\mathrm{dist}(x, y) = 2$, then $|\deg(x) - \deg(y)| \leq r-2$. By the triangle inequality and induction, it follows that if $x, y \in X$, then $|\deg(x) - \deg(y)| \leq (\frac{r-2}{2}) \mathrm{dist}(x, y)$. Observe that $\mathrm{vc}(P_{2\,\mathrm{vc}(G)+2}) = \mathrm{vc}(G) + 1$, so G must be $P_{2\,\mathrm{vc}(G)+2}$-free. Since G is connected, it follows that if $x, y \in V(G)$, then $\mathrm{dist}(x, y) < 2\,\mathrm{vc}(G) + 1$. We conclude that if $x, y \in X$, then $|\deg(x) - \deg(y)| \leq \mathrm{vc}(G)(r-2)$. Note that if a vertex $x \in V(G)$ has degree at least $\mathrm{vc}(G) + 1$, then $|N(x)| > |S|$ and so $x \in S$.

Since $|X| > (\mathrm{vc}(G))^2(r-1) > \mathrm{vc}(G) = |S|$, there must be a vertex $y \in X \setminus S$. Since $y \in X \setminus S$, it follows that $\deg(y) \leq \mathrm{vc}(G)$. It follows that $\deg(x) \leq \deg(y) + \mathrm{vc}(G)(r-2) \leq \mathrm{vc}(G)(r-1)$ for all $x \in X$. We conclude that $S' = (S \cap Y) \cup (N(S \cap X))$ is an independent vertex cover in G of size at most $(\mathrm{vc}(G))^2(r-1)$. This completes the proof. □

A graph is an *almost complete bipartite graph* if it can be obtained from a complete bipartite graph by removing a (possibly empty) set of edges that form a matching. We need the following lemma due to Alekseev.

Lemma 30 ([2]). *Every connected $K_{1,3}^+$-free bipartite graph is either a path, a cycle or an almost complete bipartite graph.*

We also need the following lemma.

Lemma 31. *Let G be an almost complete bipartite graph. Then* $\mathrm{ivc}(G) = \mathrm{vc}(G)$.

Proof. Notice that $\mathrm{ivc}(G) = \mathrm{vc}(G)$ holds if and only if the equality holds for every connected component of G. Therefore, without loss of generality, we may assume that G is connected. Let X, Y be the parts of the bipartition of G, and let S be a minimum vertex cover of G. We may assume without loss of generality that $|X| \leq |Y|$. If $\mathrm{vc}(G) \leq 1$, then $\mathrm{ivc}(G) = \mathrm{vc}(G)$. Therefore we may assume that $|X| \geq \mathrm{vc}(G) \geq 2$. If S is independent or $|S| = |X|$, then again $\mathrm{ivc}(G) = \mathrm{vc}(G)$.

Now we assume that S is not independent and $|X| > |S|$. This implies that there exist two adjacent vertices $x \in X \cap S$ and $y \in Y \cap S$, and another vertex $y' \in Y \setminus S$. Since G

is a connected almost complete bipartite graph, the vertex y' is adjacent to all vertices of X but at most one. Moreover, since $y' \notin S$, the neighbourhood of y' is contained in S. Therefore $|X| > |S| \geq |\{y\} \cup N(y')| \geq 1 + (|X| - 1) = |X|$, a contradiction. □

Our next theorem is the main result of this section and immediately implies Theorems 25 and 26. If an upper bound given in this theorem is tight, that is, if there exists an H-free bipartite graph G for which equality holds, we indicate this by a $*$ in the corresponding row (whereas the other upper bounds are not known to be tight).

Theorem 31. *Let H be a graph. Then the following two statements hold:*

(i) the class of H-free bipartite graphs is ivc*–bounded if and only if H is an induced subgraph of $K_{1,r} + rP_1$ or $K^+_{1,r}$ for some $r \geq 1$; and*
(ii) the class of H-free bipartite graphs is ivc*–identical if and only if H is an induced subgraph of $K^+_{1,3}$ or $2P_1 + P_3$.*

In particular, the following statements hold for every H-free bipartite graph G:

(1)$*$ $\text{ivc}(G) = \text{vc}(G)$ *if* $H \subseteq_i K^+_{1,3}$ *or* $H \subseteq_i 2P_1 + P_3$
(2)$*$ $\text{ivc}(G) \leq \text{vc}(G) + 1$ *if* $H = K_{1,3} + P_1$
(3) $\text{ivc}(G) \leq \text{vc}(G) + s - 3$ *if* $H = sP_1$ *for* $s \geq 5$
(4) $\text{ivc}(G) \leq \text{vc}(G) + s - 2$ *if* $H = sP_1 + P_2$ *for* $s \geq 3$
(5)$*$ $\text{ivc}(G) \leq \text{vc}(G) + s - 2$ *if* $H = sP_1 + P_3$ *for* $s \geq 3$
(6) $\text{ivc}(G) \leq \text{vc}(G) + 3s + 2$ *if* $H = K_{1,3} + sP_1$ *for* $s \geq 2$
(7)$*$ $\text{ivc}(G) \leq (r - 1)\,\text{vc}(G) - 1$ *if* $H = K_{1,r}$ *for* $r \geq 4$
(8) $\text{ivc}(G) \leq r \cdot \text{vc}(G) + rs$ *if* $H = K_{1,r} + sP_1$ *for* $r \geq 4, s \geq 1$
(9) $\text{ivc}(G) \leq (r - 1)\,\text{vc}(G)^2$ *if* $H = K^+_{1,r}$ *for* $r \geq 4$

Proof. We start by proving (i).

(i): "⇐". First suppose that H is an induced subgraph of $K_{1,r} + rP_1$ or $K^+_{1,r}$ for some r, then Lemma 28 or 29, respectively, implies that the class of H-free bipartite graphs is ivc–bounded.

(i): "⇒". Now suppose that the class of H-free bipartite graphs is ivc–bounded, that is, there is a function $f : \mathbb{N} \to \mathbb{N}$ such that $\text{ivc}(G) \leq f(\text{vc}(G))$ for all H-free bipartite graphs G. We will show that H is an induced subgraph of $K_{1,r} + rP_1$ or $K^+_{1,r}$ for some r.

For $r \geq 1$, $s \geq 2$, let D^r_s denote the graph formed from $2K_{1,s}$ and P_{2r} by identifying the two end-vertices of the P_{2r} with the central vertices of the respective $K_{1,s}$'s (see also

Fig. 21; note that $D^1_s = D_{s,s}$). It is easy to verify that $\mathrm{vc}(D^r_s) = r + 1$ and $\mathrm{ivc}(D^r_s) = r + s$. Note that, for every $r \geq 1$,

$$\mathrm{ivc}(D^r_{f(r+1)}) = r + f(r + 1) = r + f(\mathrm{vc}(D^r_{f(r+1)})) > f(\mathrm{vc}(D^r_{f(r+1)})).$$

Hence, for every $r \geq 1$, $D^r_{f(r+1)}$ cannot be H-free. Note that for $r \geq 1$ and $s, t \geq 2$, if $s \leq t$ then D^r_s is an induced subgraph of D^r_t. Therefore, for each $r \geq 1$, there must be an s such that D^r_s is not H-free. In other words, for each $r \geq 1$, H must be an induced subgraph of D^r_s for some s.

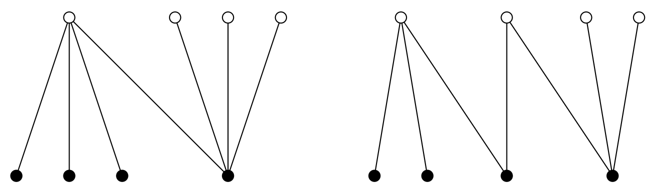


Fig. 21: The graphs $D^1_3 = D_{3,3}$ and D^2_2. The black vertices form a minimum independent vertex cover.

In particular, the above means that we may assume that H is an induced subgraph of D^1_t for some $t \geq 1$. If H contains at most one of the central vertices of the stars that form the D^1_t, then H is an induced subgraph of $K_{1,t} + tP_1$ and we are done, so we may assume H contains both central vertices. If one of these central vertices has at most one neighbour that is not a central vertex, then H is an induced subgraph of $K^+_{1,t+1}$, and we are done. We may therefore assume that H contains an induced D^1_2. However, for every $s \geq 2$, D^2_s is D^1_2-free and therefore H-free, a contradiction. This completes the proof of (i).

We now prove (ii). Let H be a graph.

(ii): "⇐". First suppose that H is an induced subgraph of $K^+_{1,3}$ or of $2P_1 + P_3$.

Case 1: $H = K^+_{1,3}$.
Let G be a $K^+_{1,3}$-free bipartite graph. We may assume without loss of generality that G is connected. By Lemma 30, G is either a path, a cycle or an almost complete bipartite graph. For the first two cases it is readily seen that $\mathrm{ivc}(G) = \mathrm{vc}(G)$. For the third case we apply Lemma 31.

Case 2: $H = 2P_1 + P_3$.

Let G be a $(2P_1 + P_3)$-free bipartite graph with bipartite classes A and B, and let S be a minimum vertex cover of G. Suppose S is not an independent set. Then S contains two adjacent vertices x and y, say $x \in A$ and $y \in B$. Let I_x and I_y be the set of neighbours of x and y, respectively, in $V(G) \setminus S$. As S has minimum size, I_x and I_y are both nonempty. Moreover, as G is bipartite, $I_x \cap I_y = \emptyset$. As the vertices of $G - S$ form an independent set, no two vertices in $I_x \cup I_y$ are adjacent. Then $|I_x| \leq 1$ or $|I_y| \leq 1$, say $|I_x| \leq 1$, as otherwise x, two vertices of I_x and two vertices of I_y form an induced $2P_1 + P_3$ in G, a contradiction.

Let $I_x = \{u\}$. If $|I_y| \geq 2$, we replace S by $S' = (S \setminus \{x\}) \cup \{u\}$ to obtain another minimum vertex cover of G. Moreover, u has no neighbours in S'. In order to see this, let z be a neighbour of u in S', and let v_1, v_2 be two vertices in I_y. As $V(G) \setminus S$ is an independent set, u is non-adjacent to v_1 and v_2. As v_1, v_2, x, z all belong to A, they are also mutually non-adjacent. Hence, the set $\{v_1, v_2, x, u, z\}$ induces a $2P_1 + P_3$ in G, a contradiction. We conclude that replacing x by u yields a minimum vertex cover S' such that $G[S']$ contains at least one fewer edge than $G[S]$.

Let now S^* be a minimum vertex cover such that $G[S^*]$ has as few edges as possible. If S^* is independent, then we have proven that $\text{ivc}(G) = \text{vc}(G)$. Suppose S^* is not an independent set. Then S^* contains two adjacent vertices x^* and y^*, say $x^* \in A$ and $y^* \in B$. By the choice of S^* and the above discussion, we conclude that each of x^* and y^* has exactly one (private) neighbour in $V(G) \setminus S^*$. Since G is $(2P_1 + P_3)$-free, this means that $G - S^*$ has at most three vertices. The latter implies that at least one of $|A \setminus S^*|$ and $|B \setminus S^*|$, say $|A \setminus S^*|$, has at most one vertex. But now, since $|B \cap S^*| \geq 1$, it follows that $\text{ivc}(G) \geq \text{vc}(G) = |S^*| = |A \cap S^*| + |B \cap S^*| \geq |A \cap S^*| + |A \setminus S^*| = |A| \geq \text{ivc}(G)$, and hence $\text{ivc}(G) = \text{vc}(G)$.

(ii): "⇒". Now suppose that H is not an induced subgraph of $K_{1,3}^+$ or of $2P_1 + P_3$. By (i), we need only consider the case when H is an induced subgraph of $K_{1,r} + rP_1$ or $K_{1,r}^+$ for some $r \geq 1$. Hence, H contains an induced subgraph from the set $\{K_{1,4}, K_{1,3} + P_1, 3P_1 + P_2, 5P_1\}$. Let G be the double star $D_{2,2}$ with two leaves for each central vertex, that is, G is the tree on vertices x, y, u_1, u_2, v_1, v_2 and edges xy, u_1x, u_2x, v_1y, v_2y. We note that G is bipartite and $(K_{1,4}, K_{1,3} + P_1, 3P_1 + P_2, 5P_1)$-free and thus H-free, while $\text{vc}(G) = 2$ and $\text{ivc}(G) = 3$. This completes the proof of (ii).

We now consider Statements (1)–(9). Statement (1) follows directly from (ii), whereas Lemmas 28 and 29 imply statements (8) and (9), respectively. We prove Statements (2)–(7) separately.

(2). Let G be a $(K_{1,3} + P_1)$-free bipartite graph with partition classes A and B. If G has maximum degree 2, then G is the disjoint union of paths and even cycles, implying that $\mathrm{ivc}(G) = \mathrm{vc}(G)$. Hence, we may assume that G contains a vertex u of degree at least 3, say $u \in A$. Note that G must be connected, as otherwise u, three neighbours of u and a vertex from another connected component of G induce a $K_{1,3} + P_1$ in G. By the $(K_{1,3} + P_1)$-freeness of G, we also find that u is adjacent to every vertex of B.

First suppose that $|B| \geq 5$. Consider an arbitrary vertex $u' \in A \setminus \{u\}$. We find that u' is adjacent to all but at most two vertices of B, as otherwise u, u' and three non-neighbours of u' in B induce a $K_{1,3} + P_1$ in G, a contradiction. As $|B| \geq 5$, this means that u' has at least three neighbours in B. Again by $(K_{1,3} + P_1)$-freeness, we find that u' is also adjacent to all vertices of B. As u' is an arbitrary vertex, we conclude that G is a complete bipartite graph, which implies that $\mathrm{ivc}(G) = \mathrm{vc}(G)$.

Now suppose that $|B| \leq 4$. As B is an independent vertex cover of G, we find that $\mathrm{ivc}(G) \leq 4$. If $\mathrm{vc}(G) = 3$, then $\mathrm{ivc}(G) \leq \mathrm{vc}(G) + 1$ (so Statement (2) holds). If $\mathrm{vc}(G) \leq 1$, then $\mathrm{ivc}(G) = \mathrm{vc}(G)$. Hence, we may assume that $\mathrm{vc}(G) = 2$. Let $S = \{x, y\}$ be a minimum vertex cover. If S is independent, then $\mathrm{ivc}(G) = \mathrm{vc}(G) = 2$, so we may assume that x and y are adjacent. As G is connected, bipartite, and $V(G) \setminus S$ is an independent set, we find that G is a double star. As G is $(K_{1,3} + P_1)$-free and contains a vertex of degree at least 3, and moreover S is a minimum vertex cover of G, we find that $G = D_{1,2}$ or $G = D_{2,2}$. Then $\mathrm{ivc}(G) = \mathrm{vc}(G)$ holds in the former case and $\mathrm{ivc}(G) = \mathrm{vc}(G) + 1$ holds in the latter case. Hence we have proven the bound of (2) and also, as demonstrated by the graph $D_{2,2}$, that this bound is tight.

(3). For some $s \geq 5$, let G be an sP_1-free bipartite graph with partition classes A and B. If $\mathrm{vc}(G) \leq 1$, then $\mathrm{ivc}(G) = \mathrm{vc}(G)$ and thus $\mathrm{ivc}(G) \leq \mathrm{vc}(G) + s - 3$. Suppose that $\mathrm{vc}(G) \geq 2$. As G is sP_1-free, $|A| \leq s - 1$ holds. As A is an independent vertex cover, this means that $\mathrm{ivc}(G) \leq s - 1 = 2 + s - 3 \leq \mathrm{vc}(G) + s - 3$.

(4) and (5). Note that the bound for (5) immediately implies (4), so it is sufficient to prove Statement (5). For some $s \geq 3$, let G be a $(sP_1 + P_3)$-free bipartite graph with partition classes A and B. Let S be a minimum vertex cover of G. First suppose that each vertex of S has at most one neighbour in $V(G) \setminus S$. As S has minimum size, this means that each vertex of S has exactly one neighbour in $V(G) \setminus S$. We replace every $u \in S \cap A$ with its unique neighbour in $V(G) \setminus S$, and note that his neighbour belongs to B. This results in a vertex cover S^* of the same size as S, but which is a subset of B. This implies that S^* is independent. Thus in this case it follows that $\mathrm{ivc}(G) = \mathrm{vc}(G)$.

Now suppose that S contains a vertex u, say $u \in A$, with at least two neighbours in $V(G) \setminus S$. As G is $(sP_1 + P_3)$-free and $V(G) \setminus S$ is independent, this means that at most $s - 1$ vertices of $G - S$ belong to A. First suppose that $S \subseteq A$. Then, as A is an independent set, we find that S is independent and thus $\text{ivc}(G) = \text{vc}(G)$. Now suppose that $S \setminus A \neq \emptyset$, so $|A \cap S| \leq |S| - 1$. As A is an independent vertex cover of G, we find that $\text{ivc}(G) \leq |A| = |A \cap S| + |A \cap V(G - S)| \leq |S| - 1 + s - 1 = \text{vc}(G) + s - 2$.

The graph $D_{s-1,s-1}$, which is $(sP_1 + P_3)$-free, demonstrates the above bound is tight: indeed $\text{vc}(D_{s-1,s-1}) = 2$, whereas $\text{ivc}(D_{s-1,s-1}) = s - 1 + 1 = s = \text{vc}(D_{s-1,s-1}) + s - 2$.

(6). For $s \geq 2$, let G be a $(K_{1,3} + sP_1)$-free bipartite graph with partition classes A and B. If A or B has fewer than $\max\{3s + 2, \text{vc}(G) + s\}$ vertices, then we can take the smallest partition class as an independent vertex cover to obtain the desired bound. We may therefore assume that both A and B have size at least $\max\{3s + 2, \text{vc}(G) + s\}$.

If every vertex in G has degree at most 2, then G is $K_{1,3}$-free and by (1) we find that $\text{ivc}(G) = \text{vc}(G)$. By Lemma 27, we may therefore assume that fewer than s vertices of A have more than $s-1$ non-neighbours in B. We will show that this leads to a contradiction.

Let S be a minimum vertex cover of G. Since A and B each have at least $\text{vc}(G) + s$ vertices, there must be a set S' of $\text{vc}(G) + 1$ vertices in A that has at least $\text{vc}(G) + 1$ neighbours in B. If a vertex $x \in V(G)$ has degree at least $\text{vc}(G) + 1$, then $|N(x)| > |S|$, so $x \in S$. Therefore every vertex of S' must be in S, contradicting the fact that $|S'| = \text{vc}(G) + 1 > \text{vc}(G) = |S|$.

(7). For some $r \geq 4$, let G be a $K_{1,r}$-free bipartite graph with partition classes A and B. Let S be a minimum vertex cover of G. If S is independent, then $\text{ivc}(G) = \text{vc}(G)$. Suppose that S is not independent. Let $A^* \subseteq A$ be the set of neighbours of the vertices in $S \cap B$. Note that $|(S \cap A) \cap A^*| \geq 1$, as S is not independent. Also note that $(S \cap A) \cup A^*$ is an independent vertex cover of G. Hence $\text{ivc}(G) \leq |(S \cap A) \cup A^*| = |S \cap A| + |A^*| - |(S \cap A) \cap A^*| \leq |S \cap A| + (r-1)|S \cap B| - 1$. Similarly, $\text{ivc}(G) \leq |S \cap B| + (r-1)|S \cap A| - 1$. Therefore $\text{ivc}(G) \leq \frac{1}{2}(|S \cap A| + (r-1)|S \cap B| - 1 + |S \cap B| + (r-1)|S \cap A| - 1) = \frac{1}{2}(r|S \cap A| + r|S \cap B| - 2) = \frac{1}{2}(r|S|) - 1 = \frac{r}{2}|S| - 1$. To see that this is tight, note that $D_{r-2,r-2}$ is a $K_{1,r}$-free bipartite graph with $\text{vc}(D_{r-2,r-2}) = 2$ and $\text{ivc}(D_{r-2,r-2}) = r - 1 = \frac{r}{2}\,\text{vc}(D_{r-2,r-2}) - 1$. □

6.4 Feedback Vertex Set

In this section we prove Theorems 27 an 28 as part of a more general theorem. Recall that a graph has an independent feedback vertex set if and only if it is near-bipartite. We first show the following lemma.

Lemma 32. *If G is a $(P_1 + P_2)$-free near-bipartite graph, then* $\mathrm{ifvs}(G) = \mathrm{fvs}(G)$.

Proof. Let G be a $(P_1 + P_2)$-free near-bipartite graph. Note that $\overline{G}$ is a P_3-free graph, so $\overline{G}$ is a disjoint union of cliques. It follows that G is a complete multi-partite graph, say with a partition of its vertex sets into k non-empty independent sets $V_1, \ldots, V_k$. We may assume that $k \geq 2$, otherwise G is an edgeless graph, in which case $\mathrm{ifvs}(G) = \mathrm{fvs}(G) = 0$ and we are done. Since G is near-bipartite, it contains an independent set I such that $G - I$ is a forest. Note that $I \subseteq V_i$ for some $i \in \{1, \ldots, k\}$. Since near-bipartite graphs are 3-colourable, it follows that $k \leq 3$. Furthermore, if $k = 3$, then $|V_j| = 1$ for some $j \in \{1, 2, 3\} \setminus \{i\}$, otherwise $G - I$ would contain an induced C_4, a contradiction. In other words G is either a complete bipartite graph or the graph formed from a complete bipartite graph by adding a dominating vertex.

First suppose that $k = 2$, so G is a complete bipartite graph. Without loss of generality assume that $|V_1| \geq |V_2| \geq 1$. Let S be a feedback vertex set of G. If there are two vertices in $V_1 \setminus S$ and two vertices in $V_2 \setminus S$, then these vertices would induce a C_4 in $G - S$, a contradiction. Therefore S must contain all but at most one vertex of V_1 or all but at most one vertex of V_2, so $\mathrm{fvs}(G) \geq \min\{|V_1| - 1, |V_2| - 1\} = |V_2| - 1$. Let I be a set consisting of $|V_2| - 1$ vertices of V_2. Then I is independent and $G - I$ is a star, so I is an independent feedback vertex set. It follows that $\mathrm{ifvs}(G) \leq |V_2| - 1$. Since $\mathrm{fvs}(G) \leq \mathrm{ifvs}(G)$, we conclude that $\mathrm{ifvs}(G) = \mathrm{fvs}(G)$ in this case.

Now suppose that $k = 3$, so G is obtained from a complete bipartite graph by adding a dominating vertex. Without loss of generality assume that $|V_1| \geq |V_2| \geq |V_3| = 1$. Let S be a feedback vertex set of G. By the same argument as in the $k = 2$ case, S must contain all but at most one vertex of V_1 or all but at most one vertex of V_2. If there is a vertex in $V_i \setminus S$ for all $i \in \{1, 2, 3\}$, then these three vertices would induce a C_3 in $G - S$, a contradiction. Therefore S must contain every vertex in V_i for some $i \in \{1, 2, 3\}$. Since $|V_1| \geq |V_2| \geq |V_3| = 1$, it follows that $|S| \geq \min\{|V_2| - 1 + |V_3|, |V_2|\} = |V_2|$. Therefore $\mathrm{fvs}(G) \geq |V_2|$. Now V_2 is an independent set and $G - V_2$ is a star, so V_2 is an independent feedback vertex set. It follows that $\mathrm{ifvs}(G) \leq |V_2|$. Since $\mathrm{fvs}(G) \leq \mathrm{ifvs}(G)$, we conclude that $\mathrm{ifvs}(G) = \mathrm{fvs}(G)$. □

Lemma 33. *If $r \geq 1$ and G is a $K_{1,r}$-free near-bipartite graph, then* $\mathrm{ifvs}(G) \leq (2r^2 - 5r + 3)\,\mathrm{fvs}(G)$.

Proof. Fix integers $k \geq 0$ and $r \geq 1$. Suppose G is a $K_{1,r}$-free near-bipartite graph with a feedback vertex set S such that $|S| = k$. Since G is near-bipartite, $V(G)$ can be partitioned into an independent set V_1 and a set $V(G) \setminus V_1$ that induces a forest in G. Since forests are bipartite, we can partition $V(G) \setminus V_1$ into two independent sets V_2 and V_3.

Suppose $x \in V_i$ for some $i \in \{1, 2, 3\}$. Then x has no neighbours in V_i since V_i is an independent set. For $j \in \{1, 2, 3\} \setminus \{i\}$, the vertex x can have at most $r - 1$ neighbours in V_j, otherwise G would contain an induced $K_{1,r}$. It follows that $\deg(x) \leq 2(r - 1)$ for all $x \in V(G)$.

Let $S' = S$. Let $F' = V(G) \setminus S'$, so $G[F']$ is a forest. To prove the lemma, we will iteratively modify S' until we obtain an independent feedback vertex set S' of G with $|S'| \leq (2r^2 - 5r + 3)|S|$. Every vertex $u \in S'$ has at most $2r - 2$ neighbours in F'. Consider two neighbours v, w of u in F'. As F' is a forest, there is at most one induced path in F' from v to w, so there is at most one induced cycle in $G[F' \cup \{u\}]$ that contains all of u, v and w. Therefore $G[F' \cup \{u\}]$ contains at most $\binom{2r-2}{2} = \frac{1}{2}(2r-2)(2r-2-1) = 2r^2 - 5r + 3$ induced cycles. Note that every cycle in G contains at least one vertex of V_1. Therefore, if $s \in S' \cap (V_2 \cup V_3)$, then we can find a set X of at most $2r^2 - 5r + 3$ vertices in $V_1 \setminus S'$ such that if we replace s in S' by the vertices of X, then we again obtain a feedback vertex set. Repeating this process iteratively, for each vertex we remove from $S' \cap (V_2 \cup V_3)$, we add at most $2r^2 - 5r + 3$ vertices to $S' \cap V_1$. We stop the procedure once $S' \cap (V_2 \cup V_3)$ becomes empty, at which point we have produced a feedback vertex set S' with $|S'| \leq (2r^2 - 5r + 3)|S|$. Furthermore, at this point $S' \subseteq V_1$, so S' is independent. It follows that $\text{ifvs}(G) \leq (2r^2 - 5r + 3)\,\text{fvs}(G)$. □

Note that all near-bipartite graphs are 3-colourable (use one colour for the independent set and the two other colours for the forest). We prove the following lemma.

Lemma 34. *Let $k \geq 3$. The class of C_k-free near-bipartite graphs is* ifvs*-unbounded and* ioct*-unbounded.*

Proof. For $r, s \geq 2$, let S_s^r denote the graph constructed as follows (see also Fig. 22). Start with the graph that is the disjoint union of $2s$ copies of P_{2r}, and label these copies $U^1, \ldots, U^s, V^1, \ldots, V^s$. Add a vertex u adjacent to both endpoints of every U^i and a vertex v adjacent to both endpoints of every V^i. Finally, add an edge between u and v.

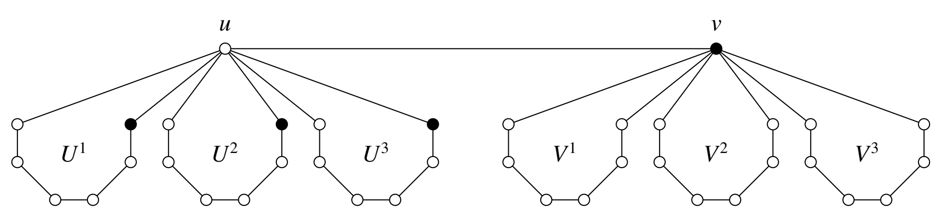


Fig. 22: The graph S_3^3.

Every induced cycle in S^r_s is isomorphic to C_{2r+1}, which is an odd cycle. Thus a set $S \subseteq V(S^r_s)$ is a feedback vertex set for S^r_s if and only if it is an odd cycle transversal for S^r_s. It follows that $\text{fvs}(S^r_s) = \text{oct}(S^r_s)$ and $\text{ifvs}(S^r_s) = \text{ioct}(S^r_s)$.

Now $\{u, v\}$ is a minimum feedback vertex set of S^r_s, so $\text{fvs}(S^r_s) = \text{oct}(S^r_s) = 2$. However, any independent feedback vertex set S contains at most one vertex of u and v; say it does not contain u. Then it must contain at least one vertex of each U^i. It follows that $\text{ifvs}(S^r_s) = \text{ioct}(S^r_s) \geq s + 1$. Since for every $s \geq 2$, $k \geq 3$, the graph S^k_s is C_k-free, this completes the proof. □

We are now ready to prove the main result of this section, which immediately implies Theorems 27 and 28. If an upper bound given in this theorem is tight, that is, if there exists an H-free near-bipartite graph G for which equality holds, we again indicate this by a $*$ in the corresponding row (whereas the other upper bounds are not known to be tight).

Theorem 32. *Let H be a graph. Then the following two statements hold:*

(i) the class of H-free near-bipartite graphs is ifvs*-bounded if and only if H is isomorphic to $P_1 + P_2$, a star or an edgeless graph.*
(ii) for $H \neq K_{1,3}$, the class of H-free near-bipartite graphs is ifvs*-identical if and only if H is a (not necessarily induced) subgraph of P_3.*

In particular, the following statements hold for every H-free near-bipartite graph G:

(1)$*$ $\text{ifvs}(G) = \text{fvs}(G)$ *if* $H \subseteq P_3$
(2)$*$ $\text{ifvs}(G) \leq \text{fvs}(G) + 1$ *if* $H = 4P_1$
(3) $\text{ifvs}(G) \leq \text{fvs}(G) + s - 3$ *if* $H = sP_1$ *for* $s \geq 5$
(4) $\text{ifvs}(G) \leq (2r^2 - 5r + 3)\,\text{fvs}(G)$ *if* $H = K_{1,r}$ *for* $r \geq 3$.

Proof. We start by proving (i).

(i): "⇐". First suppose that H is isomorphic to $P_1 + P_2$, a star or an edgeless graph. If $H = P_1 + P_2$, then the class of H-free near-bipartite graphs is ifvs-bounded by Lemma 32. If H is isomorphic to a star or an edgeless graph, then H is an induced subgraph of $K_{1,r}$ for some $r \geq 1$. In this case the class of H-free near-bipartite graphs is ifvs-bounded by Lemma 33.

(i): "⇒". Now suppose that the class of H-free near-bipartite graphs is ifvs-bounded. By Lemma 34, H must be a forest. We will show that H is isomorphic to $P_1 + P_2$, a star or an edgeless graph.

We start by showing that H must be $(P_1 + P_3, 2P_1 + P_2, 2P_2)$-free. Let vertices x_1, x_2, x_3, x_4, in that order, form a path on four vertices. For $s \geq 3$, let T_s be the graph obtained from this path by adding an independent set I on s vertices (see also Fig. 23) that is complete to the path and note that T_s is near-bipartite. Then $\{x_1, x_2, x_3\}$ is a minimum feedback vertex set in T_s. However, if S is an independent feedback vertex set, then S contains at most two vertices in $\{x_1, x_2, x_3, x_4\}$. Therefore S must contain at least $s - 1$ vertices of I, otherwise $T_s - S$ would contain an induced C_3 or C_4. Therefore $\text{fvs}(T_s) = 3$ and $\text{ifvs}(T_s) \geq s - 1$. Note that T_s is $(P_1 + P_3, 2P_1 + P_2, 2P_2)$-free (this is easy to see by casting to the complement and observing that $\overline{T_s}$ is the disjoint union of a P_4 and a complete graph). Therefore H cannot contain $P_1 + P_3$, $2P_1 + P_2$ or $2P_2$ as an induced subgraph, otherwise T_s would be H-free, a contradiction.

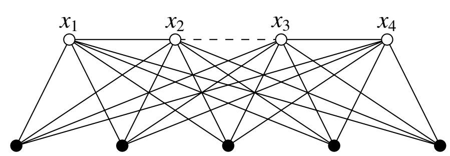


Fig. 23: The graphs T_5 and T'_5. The edge x_2x_3 is present in T_5, but not in T'_5.

Next, we show that H must be P_4-free. For $s \geq 3$ let T'_s be the graph obtained from T_s by removing the edge x_2x_3 (see also Fig. 23). Then $\{x_1, x_2, x_3\}$ is a minimum feedback vertex set in T'_s, so $\text{fvs}(T'_s) = 3$. By the same argument as for T_s, we find that $\text{ifvs}(T'_s) \geq s - 1$. Now the complement $\overline{T'_s}$ is the disjoint union of a C_4 and a complete graph, so T'_s is P_4-free. Therefore H cannot contain P_4 as an induced subgraph.

We may now assume that H is a $(P_1 + P_3, 2P_1 + P_2, 2P_2, P_4)$-free forest. If H is connected, then it is a P_4-free tree, so it is a star, in which case we are done. We may therefore assume that H is disconnected. We may also assume that H contains at least one edge, otherwise we are done. Since H is $(2P_1 + P_2)$-free, it cannot have more than two connected components. Since H is $2P_2$-free, one of its two connected components must be isomorphic to P_1. Since H is a $(P_1 + P_3)$-free forest, its other connected component must be isomorphic to P_2. Hence H is isomorphic to $P_1 + P_2$. This completes the proof of (i).

We now prove (ii). Let H be a graph not isomorphic to $K_{1,3}$.

(ii): "⇐". First suppose that H is a subgraph of P_3. If $H \subseteq_i P_1+P_2$, then $\text{ifvs}(G) = \text{fvs}(G)$ for every H-free near-bipartite graph G by Lemma 32. If $H \subseteq_i P_3$, then every H-free

near-bipartite graph G is a disjoint union of complete graphs on at most three vertices, and hence $\text{ifvs}(G) = \text{fvs}(G)$ holds. Finally, suppose that $H \subseteq_i 3P_1$. Let G be a $3P_1$-free near-bipartite graph. As G is $3P_1$-free, every minimum independent feedback vertex set of G has size at most 2. Hence, every minimum feedback vertex set of G also has size at most 2. Moreover, if it has size less than 2, then it is an independent feedback vertex set. We conclude that $\text{ifvs}(G) = \text{fvs}(G)$.

(ii): "⇒". Now suppose that H is not a subgraph of P_3. Recall that we assume that $H \neq K_{1,3}$. By (i), we may then assume that $H = K_{1,r}$ for some $r \geq 4$ or $H = sP_1$ for some $s \geq 4$. Consider the graph G in Fig. 24. It is straightforward to check that G is $4P_1$-free and near-bipartite; $\{u, v\}$ is a minimum feedback vertex set (indeed $G - \{u, v\}$ is P_5) while $\text{ifvs}(G) = 3$; for instance, $\{v, v_2, v_3\}$ is a minimum independent feedback vertex set of G. This completes the proof of (ii).

We now consider Statements (1)–(4). Statement (1) follows directly from Statement (ii), whereas Lemma 33 implies Statement (4). We prove Statements (2) and (3) below.

(2) and (3). First note that, as shown in the proof of Statement (ii), the graph G in Fig. 24 is $4P_1$-free, with $\text{fvs}(G) = 2$ and $\text{ifvs}(G) = 3$, so the bound in Statement (2) is tight. It remains to prove that $\text{ifvs}(G) \leq \text{fvs}(G) + s - 3$ if $H = sP_1$ with $s \geq 4$ (this proves the bounds in Statements (2) ($s = 4$) and (3) ($s \geq 5$)). Let G be an sP_1-free near-bipartite graph. If $\text{fvs}(G) \leq 1$, then $\text{ifvs}(G) = \text{fvs}(G)$. Hence, we may assume that $\text{fvs}(G) \geq 2$. As G is near-bipartite, $V(G)$ can be partitioned into three independent sets V_1, V_2, V_3, such that $V_2 \cup V_3$ induce a forest. Hence, V_1 is an independent feedback vertex set. As G is sP_1-free, V_1 has size at most $s - 1$. This means that $\text{ifvs}(G) \leq s - 1 = 2 + s - 3 \leq \text{fvs}(G) + s - 3$. This completes the proof of Statements (2) and (3). □

6.5 Odd Cycle Transversal

In this section we prove Theorems 29 and 30 as part of a more general theorem. Recall that a graph has an independent odd cycle transversal if and only if it is 3-colourable. Before proving the main result of this section, we first provide a sequence of auxiliary statements.

Lemma 35. *If G is a P_4-free 3-colourable graph, then* $\text{ioct}(G) = \text{oct}(G)$.

Proof. Let G be a P_4-free 3-colourable graph. It suffices to prove the lemma for each connected component, so we may assume that G is connected. Note that G cannot contain

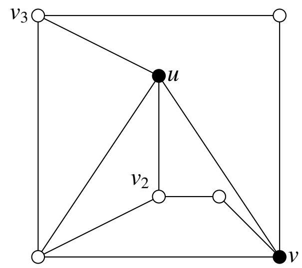


Fig. 24: An example of a $4P_1$-free near-bipartite graph G with $\text{ifvs}(G) = \text{fvs}(G) + 1$, which shows that the bound in Theorem 32(2) is tight.

any induced odd cycles on more than three vertices, as it is P_4-free. Let (V_1, V_2, V_3) be a partition of $V(G)$ into independent sets. We may assume that G is not bipartite, otherwise $\text{ioct}(G) = \text{oct}(G) = 0$, in which case we are done. As G is connected, P_4-free and contains more than one vertex, its complement $\overline{G}$ must be disconnected. Therefore we can partition the vertex set of G into two parts X_1 and X_2 such that X_1 is complete to X_2. No independent set V_i can have vertices in both X_1 and X_2, so without loss of generality we may assume that $X_1 = V_1$ and $X_2 = V_2 \cup V_3$. Since $G[X_2]$ is a P_4-free bipartite graph, it is readily seen that it is a disjoint union of complete bipartite graphs.

Note that $G - X_1$ is a bipartite graph, so X_1 is an odd cycle transversal of G. Furthermore, X_1 is independent. Now let S be a minimum vertex cover of $G[X_2]$. Observe that $G - S$ is bipartite, so S is an odd cycle transversal of G. Since $G[X_2]$ is the disjoint union of complete bipartite graphs, for every connected component C of $G[X_2]$, S must contain one part of the bipartition of C, or the other; by minimality of S, it only contains vertices from one of the parts. It follows that S is independent.

We now claim that every minimum odd cycle transversal S of G contains either X_1 or a minimum vertex cover of $G[X_2]$, both of which we have shown are independent odd cycle transversals; by the minimality of S, this will imply that S is equal to one of them. Indeed, suppose for contradiction that S is a minimum odd cycle transversal such that there is a vertex $x \in X_1 \setminus S$ and two adjacent vertices $y, z \in X_2 \setminus S$. Then $G[\{x, y, z\}]$ is a C_3 in $G - S$. This contradiction completes the proof. □

Lemma 36. *If G is a $K_{1,3}$-free 3-colourable graph, then* $\text{ioct}(G) \leq 3\,\text{oct}(G)$.

Proof. Fix an integer $k \geq 0$. Let G be a $K_{1,3}$-free 3-colourable graph with an odd cycle transversal S such that $|S| = k$. Fix a partition of $V(G)$ into three independent sets V_1, V_2, V_3. Without loss of generality assume that $|S \cap V_1| \geq |S \cap V_2|, |S \cap V_3|$, so

$|S \cap (V_2 \cup V_3)| \leq \frac{2k}{3}$. Let $S' = S$ and note that $G - S'$ is bipartite by definition of odd cycle transversal. To prove the lemma, we will iteratively modify S' until we obtain an independent odd cycle transversal S' of G with $|S'| \leq 3k$.

Suppose $x \in V_i$ for some $i \in \{1, 2, 3\}$. Then x has no neighbours in V_i since V_i is an independent set. For $j \in \{1, 2, 3\} \setminus \{i\}$, the vertex x can have at most two neighbours in V_j, otherwise G would contain an induced $K_{1,3}$. It follows that $\deg(x) \leq 4$ for all $x \in V(G)$.

As $G - S'$ is a bipartite $K_{1,3}$-free graph, it is a disjoint union of paths and even cycles. Every vertex $u \in S'$ has at most four neighbours in $V(G) \setminus S'$. An induced odd cycle in $G - (S' \setminus \{u\})$ consists of the vertex u and an induced path P in $G - S'$ between two neighbours v, w of u such that $P \cap N(u)$ does not contain any vertices apart from v and w. If u has q neighbours in some connected component C of $G - S'$, then there can be at most q such paths P that lie in this connected component. It follows that there are at most four induced odd cycles in $G - (S' \setminus \{u\})$. Note that every induced odd cycle in G contains at least one vertex in each V_i. Therefore, if $s \in S' \cap (V_2 \cup V_3)$, then we can find a set X of at most four vertices in $V_1 \setminus S'$ such that if we replace s in S' by the vertices of X, then we again obtain an odd cycle transversal. Repeating this process iteratively, for each vertex we remove from $S' \cap (V_2 \cup V_3)$, we add at most four vertices to $S' \cap V_1$, so $|S'|$ increases by at most 3. We stop the procedure once $S' \cap (V_2 \cup V_3)$ becomes empty, at which point we have produced an odd cycle transversal S' with $|S'| \leq |S| + 3|S \cap (V_2 \cup V_3)| \leq k + 3 \times \frac{2k}{3} = 3k$. Furthermore, at this point $S' \subseteq V_1$, so S' is independent. It follows that $\mathrm{ioct}(G) \leq 3\,\mathrm{oct}(G)$. □

Lemma 37. *Let* $r, s \geq 1$. *Suppose there is a function* $f : \mathbb{N} \to \mathbb{N}$ *such that* $\mathrm{ioct}(G) \leq f(\mathrm{oct}(G))$ *for every* $K_{1,r}$*-free* 3*-colourable graph* G. *Then* $\mathrm{ioct}(G) \leq \max\{\mathrm{oct}(G)r + r^2 + 3rs - 2r, f(\mathrm{oct}(G))\}$ *for every* $(K_{1,r} + sP_1)$*-free* 3*-colourable graph* G.

Proof. Fix $r, s \geq 1$ and $k \geq 0$. Let G be a $(K_{1,r} + sP_1)$-free 3-colourable graph with a minimum odd-cycle transversal T on k vertices. Fix a partition of $V(G)$ into three independent sets V_1, V_2, V_3. We may assume that $\mathrm{oct}(G) \geq 2$, otherwise $\mathrm{ioct}(G) = \mathrm{oct}(G)$ and we are done. If $|V_i| \leq \max\{\mathrm{oct}(G)r + r^2 + 3rs - 2r, f(\mathrm{oct}(G))\}$ for some $i \in \{1, 2, 3\}$, then deleting V_i from G yields a bipartite graph, so $\mathrm{ioct}(G) \leq \max\{\mathrm{oct}(G)r + r^2 + 3rs - 2r, f(\mathrm{oct}(G))\}$ and we are done. We may therefore assume that $|V_i| > \max\{\mathrm{oct}(G)r + r^2 + 3rs - 2r, f(\mathrm{oct}(G))\}$ for all $i \in \{1, 2, 3\}$. If G is $K_{1,r}$-free, then $\mathrm{ioct}(G) \leq f(\mathrm{oct}(G))$, so suppose that G contains an induced $K_{1,r}$, say with vertex set X. Note that $|X| = r + 1$, and each V_i can contain at most r vertices of X, since every V_i is an independent set.

For every $i \in \{1, 2, 3\}$, there cannot be a set of s vertices in $V_i \setminus X$ that are anti-complete to X, otherwise G would contain an induced $K_{1,r} + sP_1$, a contradiction. For every $i \in \{1, 2, 3\}$, since $|V_i| > \mathrm{oct}(G)r + r^2 + 3rs - 2r \geq r^2 + 3rs$, it follows that

$|V_i \setminus X| \geq |V_i| - r > (s-1)+(r+1)(r-1) = (s-1)+|X|(r-1)$. Hence for every $i \in \{1,2,3\}$, there must be a vertex $x \in X$ that has at least r neighbours in V_i. Applying this for each i in turn, we find that at least two of the graphs in $\{G[V_1 \cup V_2], G[V_1 \cup V_3], G[V_2 \cup V_3]\}$ contain a vertex of degree at least r; without loss of generality assume that this is the case for $G[V_1 \cup V_2]$ and $G[V_1 \cup V_3]$. Let V_2' and V_3' denote the set of vertices in V_2 and V_3, respectively, that have more than $s-1$ non-neighbours in V_1. By Lemma 27, $|V_2'|, |V_3'| \leq s-1$.

Suppose a vertex $x \in V_2 \setminus V_2'$ is adjacent to a vertex $y \in V_3 \setminus V_3'$. By definition of V_2' and V_3', the vertices x and y each have at most $s-1$ non-neighbours in V_1. Since $|V_1| - 2(s-1) \geq \text{oct}(G)+1$, it follows that $|N(x) \cap N(y) \cap V_1| \geq \text{oct}(G)+1$ so $N(x) \cap N(y) \cap V_1 \not\subseteq T$. We conclude that at least one of x or y must be in T. In other words, $T \cap ((V_2 \setminus V_2') \cup (V_3 \setminus V_3'))$ is a vertex cover of $G[(V_2 \setminus V_2') \cup (V_3 \setminus V_3')]$, of size at most $\text{oct}(G)$. Therefore $(T \cap ((V_2 \setminus V_2') \cup (V_3 \setminus V_3'))) \cup V_2' \cup V_3'$ is a vertex cover of $G[V_2 \cup V_3]$ of size at most $\text{oct}(G)+2(s-1)$. By Lemma 28, there is an independent vertex cover T' of $G[V_2 \cup V_3]$ of size at most $(\text{oct}(G)+2(s-1))r + rs = \text{oct}(G)r + 3rs - 2r$. Note that by definition of vertex cover, $(V_2 \cup V_3) \setminus T'$ is an independent set, and so $G - T'$ is bipartite. Therefore T' is an independent odd cycle transversal for G of size at most $\text{oct}(G)r + 3rs - 2r$. This completes the proof. □

The following result follows immediately from combining Lemmas 36 and 37.

Corollary 2. *For* $s \geq 1$, $\text{ioct}(G) \leq 3\,\text{oct}(G) + 9s + 3$ *for every* $(sP_1 + K_{1,3})$*-free* 3*-colourable graph* G.

Lemma 38. *The class of* $(P_1 + P_4, 2P_2)$*-free* 3*-colourable graphs is* ioct*-unbounded.*

Proof. Let $s \geq 2$. We construct the graph Q_s as follows (see also Fig. 25). First, let A, B and C be disjoint independent sets of s vertices. Choose vertices $a \in A$, $b \in B$ and $c \in C$. Add edges so that a is complete to $B \cup C$, b is complete to $A \cup C$ and c is complete to $A \cup B$. Let Q_s be the resulting graph and note that it is 3-colourable with colour classes A, B and C.

Note that $\{a, b\}$ is a minimum odd cycle transversal of Q_s, so $\text{oct}(Q_s) = 2$.

Let S be a minimum independent odd cycle transversal. Then S contains at most one vertex in $\{a, b, c\}$, say S contains neither b nor c. If a vertex $x \in A$ is not in S, then $Q_s[\{x, b, c\}]$ is a C_3 in $Q_s - S$, a contradiction. Hence every vertex of A is in S, and so $\text{ioct}(Q_s) \geq s$.

It remains to show that Q_s is $(P_1 + P_4, 2P_2)$-free. Consider a vertex $x \in A$. Then $Q_s - N[x]$ is an edgeless graph if $x = a$ and $Q_s - N[x]$ is the disjoint union of a star and an edgeless graph otherwise. It follows that $Q_s - N[x]$ is P_4-free. By symmetry,

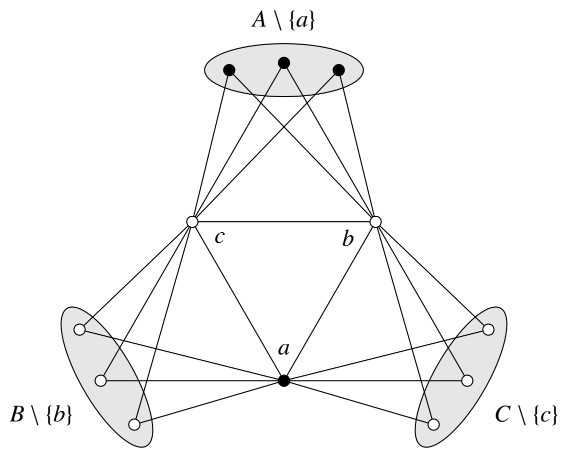


Fig. 25: The graph Q_4.

we conclude that Q_s is $(P_1 + P_4)$-free. Now consider a vertex $y \in N(a) \cap B$. Then $Q_s - N[\{a, y\}]$ is empty if $y = b$ and $Q_s - N[\{a, y\}]$ is an edgeless graph otherwise. It follows that $Q_s - N[\{a, y\}]$ is P_2-free. By symmetry, we conclude that Q_s is $2P_2$-free. This completes the proof.

Lemma 39. *Let H be a graph with more than one vertex of degree at least* 3*. Then the class of H-free* 3*-colourable graphs is* ioct*-unbounded.*

Proof. Let $s \geq 1$. We construct the graph Z_s as follows (see also Fig. 26). Start with the disjoint union of s copies of P_4 and label these copies $U^1, \ldots, U^s$. Add an edge ab and make a and b adjacent to both endpoints of every U^i. Let Z_s be the resulting graph and note that Z_s is 3-colourable (colour a and b with Colours 1 and 2, respectively, colour the endpoints of the U^is with Colour 3 and colour the remaining vertices of the U^is with Colours 1 and 2).

Note that $Z_s - \{a, b\}$ is bipartite, so $\{a, b\}$ is a minimum odd cycle transversal and $\text{oct}(Z_s) = 2$. However, any independent odd cycle transversal S contains at most one vertex of a and b; say it does not contain a. For every $i \in \{1, \ldots, s\}$, the graph $Z_s[U^i \cup \{a\}]$ is a C_5. Therefore S must contain at least one vertex from each U^i. It follows that $\text{ioct}(Z_s) \geq s$.

Let H be a graph with more than one vertex of degree at least 3. By Lemma 34, we may assume that H is a forest. It remains to show that Z_s is H-free. Suppose, for contradiction, that Z_s contains H as an induced subgraph and let x and y be two vertices that have degree at least 3 in H. Since H is a forest, x and y must each have three pairwise

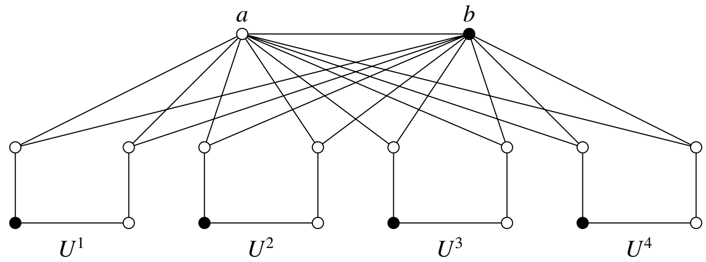


Fig. 26: The graph Z_4.

non-adjacent neighbours in Z_s. The endpoints of each U^i have exactly three neighbours, but two of them (a and b) are adjacent. Without loss of generality we may therefore assume that $x = a$ and $y = b$. Since x has degree at least 3 in H, the vertex x must have a neighbour $z \neq y$ in H and so z must be the endpoint of a U^i. Therefore x, y and z are pairwise adjacent, so $H[\{x, y, z\}]$ is a C_3, contradicting the fact that H is a forest. It follows that Z_s is H-free. This completes the proof. □

Lemma 40. *The class of $K_{1,5}$-free 3-colourable graphs is* ioct-*unbounded.*

Proof. Let $s \geq 1$. We construct the graph Y_s as follows (see also Fig. 27).

1. Start with the disjoint union of four copies of P_{3s} and label the vertices of these paths $a_1, \ldots, a_{3s}$, $b_1, \ldots, b_{3s}$, $c_1, \ldots, c_{3s}$ and $d_1, \ldots, d_{3s}$ in order, respectively.
2. For each $i \in \{1, \ldots, 3s\}$ add the edges a_ib_i and c_id_i.
3. For each $i \in \{1, \ldots, 3s - 1\}$ add the edges a_ic_{i+1} and d_ib_{i+1}.
4. Finally, add an edge xy and make x adjacent to a_1 and d_1 and y adjacent to a_1, b_1, c_1 and d_1.

Let Y_s be the resulting graph.

First note that Y_s is $K_{1,5}$-free. The vertices y, a_1 and d_1 all have degree 5, but their neighbourhood is not independent, so they cannot be the central vertex of an induced $K_{1,5}$. All the other vertices have degree at most 4, so they cannot be the central vertex of an induced $K_{1,5}$ either. Therefore no vertex in Y_s is the central vertex of an induced $K_{1,5}$, so Y_s is $K_{1,5}$-free.

The graph $Y_s - \{x, y\}$ is bipartite with bipartition classes:

1. $\{a_i, c_i \mid 1 \leq i \leq 3s, i \equiv 1 \bmod 2\} \cup \{b_i, d_i \mid 1 \leq i \leq 3s, i \equiv 0 \bmod 2\}$ and
2. $\{a_i, c_i \mid 1 \leq i \leq 3s, i \equiv 0 \bmod 2\} \cup \{b_i, d_i \mid 1 \leq i \leq 3s, i \equiv 1 \bmod 2\}$.

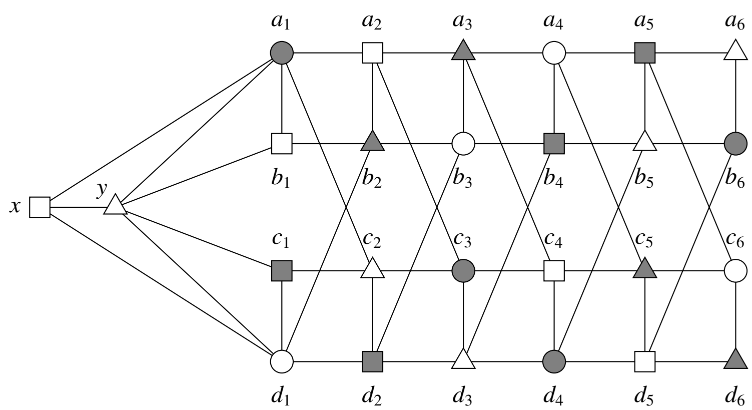


Fig. 27: The graph Y_2. Different shapes show the unique 3-colouring of Y_2. Different colours show the 2-colouring of $Y_2 - \{x, y\}$.

It follows that $\mathrm{oct}(Y_s) = 2$.
Furthermore, Y_s is 3-colourable with colour classes:

1. $\{x\} \cup \{a_i, d_i \mid 1 \leq i \leq 3s, i \equiv 2 \bmod 3\} \cup \{b_i, c_i \mid 1 \leq i \leq 3s, i \equiv 1 \bmod 3\}$,
2. $\{y\} \cup \{a_i, d_i \mid 1 \leq i \leq 3s, i \equiv 0 \bmod 3\} \cup \{b_i, c_i \mid 1 \leq i \leq 3s, i \equiv 2 \bmod 3\}$ and
3. $\{a_i, d_i \mid 1 \leq i \leq 3s, i \equiv 1 \bmod 3\} \cup \{b_i, c_i \mid 1 \leq i \leq 3s, i \equiv 0 \bmod 3\}$.

In fact, we will show that this 3-colouring is unique (up to permuting the colours). To see this, suppose that $c : V(Y_s) \to \{1, 2, 3\}$ is a 3-colouring of Y_s. Since x and y are adjacent we may assume without loss of generality that $c(x) = 1$ and $c(y) = 2$. Since a_1 and d_1 are adjacent to both x and y, it follows that $c(a_1) = c(d_1) = 3$. Since b_1 is adjacent to y and a_1, it follows that $c(b_1) = 1$. By symmetry $c(c_1) = 1$.

We prove by induction on i that for every $i \in \{1, \dots, 3s\}$, $c(a_i) = c(d_i) \equiv i + 2 \bmod 3$ and $c(b_i) = c(c_i) \equiv i \bmod 3$. We have shown that this is true for $i = 1$. Suppose that the claim holds for $i - 1$ for some $i \in \{2, \dots, 3s\}$. Then $c(a_{i-1}) = c(d_{i-1}) \equiv (i - 1) + 2 \bmod 3$ and $c(b_{i-1}) = c(c_{i-1}) \equiv i - 1 \bmod 3$. Since b_i is adjacent to b_{i-1} and d_{i-1}, it follows that $c(b_i) \equiv i \bmod 3$. Since a_i is adjacent to b_i and a_{i-1}, it follows that $c(a_i) \equiv i+2 \bmod 3$. By symmetry $c(c_i) \equiv i \bmod 3$ and $c(d_i) \equiv i+2 \bmod 3$. Therefore the claim holds for i. By induction, this completes the proof of the claim and therefore shows that the 3-colouring of Y_s is indeed unique.

Furthermore, note that the colour classes in this colouring have sizes $4s + 1$, $4s + 1$ and $4s$, respectively. A set S is an independent odd cycle transversal of a graph if and only if it is a colour class in some 3-colouring of this graph. It follows that $\text{ioct}(Y_s) = 4s$. This completes the proof. □

Before we can prove our main theorem of this section, we need one more lemma, due to Olariu.

Lemma 41 ([86]). *Every connected component of a $\overline{P_1 + P_3}$-free graph is either C_3-free or complete multi-partite.*

We are now ready to prove the main result of this section, which immediately implies Theorems 29 and 30. If an upper bound given in this theorem is tight, that is, if there exists an H-free 3-colourable graph G for which equality holds, we again indicate this by a $*$ in the corresponding row (whereas the other upper bounds are not known to be tight).

Theorem 33. *Let H be a graph. Then the following two statements hold:*

(i) the class of H-free 3-colourable graphs is ioct*-bounded*
  - *if H is an induced subgraph of P_4 or $K_{1,3} + sP_1$ for some $s \geq 0$ and*
  - *only if H is an induced subgraph of $K_{1,4}^+$ or $K_{1,4} + sP_1$ for some $s \geq 0$.*

(ii) For $H \notin \{K_{1,3}, K_{1,3}^+, 2P_1 + P_3\}$, the class of H-free 3-colourable graphs is ioct*-identical if and only if H is a (not necessarily induced) subgraph of P_4 that is not isomorphic to $2P_2$.*

In particular, the following statements hold for every H-free bipartite graph G:

(1)$*$ $\text{ioct}(G) = \text{oct}(G)$ *if* $H \subseteq P_4$ *but* $H \neq 2P_2$
(2) $\text{ioct}(G) \leq \text{oct}(G) + s - 3$ *if* $H = sP_1$ *for* $s \geq 5$
(3) $\text{ioct}(G) \leq \text{oct}(G) + 3s - 1$ *if* $H = sP_1 + P_2$ *for* $s \geq 3$
(4) $\text{ioct}(G) \leq 2\,\text{oct}(G) + 6s$ *if* $H = sP_1 + P_3$ *for* $s \geq 2$
(5) $\text{ioct}(G) \leq 3\,\text{oct}(G)$ *if* $H = K_{1,3}$
(6) $\text{ioct}(G) \leq 3\,\text{oct}(G) + 9s + 3$ *if* $H = K_{1,3} + sP_1$ *for* $s \geq 1$.

Proof. We start by proving (i).

(i): "⇐". First suppose that H is an induced subgraph of P_4 or $K_{1,3} + sP_1$ for some $s \geq 0$. Then the class of H-free 3-colourable graphs is ioct-bounded by Lemma 35 or Corollary 2, respectively.

(i): "⇒". Now suppose that the class of H-free 3-colourable graphs is ioct-bounded. We will prove that H must be an induced subgraph of $K_{1,4}^+$ or $K_{1,4} + sP_1$ for some $s \geq 0$. By Lemma 34, H must be a forest. By Lemma 40, H must be $K_{1,5}$-free. Since H is a $K_{1,5}$-free forest, it has maximum degree at most 4. By Lemma 38, H must be $(P_1 + P_4, 2P_2)$-free.

First suppose that H is P_4-free, so every connected component of H is a P_4-free tree. Hence every connected component of H is a star. In fact, as H has maximum degree at most 4, every connected component of H is an induced subgraph of $K_{1,4}$. As H is $2P_2$-free, at most one connected component of H contains an edge. Therefore H is an induced subgraph of $K_{1,4} + sP_1$ for some $s \geq 0$ and we are done.

Now suppose that H contains an induced P_4, say on vertices x_1, x_2, x_3, x_4 in that order and let $X = \{x_1, x_2, x_3, x_4\}$. Since H is a forest, every vertex $v \in V(H) \setminus X$ has at most one neighbour in X. A vertex $v \in V(H) \setminus X$ cannot be adjacent to x_1 or x_4, since H is $2P_2$-free. By Lemma 39, the vertices x_2 and x_3 cannot both have neighbours outside X; without loss of generality assume that x_3 has no neighbours in $V(H) \setminus X$. Since H is $(P_1 + P_4)$-free, every vertex $v \in V(H) \setminus X$ must have at least one neighbour in X, so it must be adjacent to x_2. As H has maximum degree at most 4, it follows that H is an induced subgraph of $K_{1,4}^+$. This completes the proof of (i).

We now prove (ii). Let H be a graph that is not isomorphic to a graph in $\{K_{1,3}, K_{1,3}^+, 2P_1+P_3\}$.

(ii): "⇐". First suppose that H is a subgraph of P_4 that is not isomorphic to $2P_2$. If H is an induced subgraph of P_4, then the claim follows from Lemma 35. It is sufficient to prove that $\text{ioct}(G) = \text{oct}(G)$ if G is a 3-colourable H-free graph in three remaining cases, namely when $H = 4P_1$, $H = P_1 + P_3$ and $H = 2P_1 + P_2$.

Case 1: $H = 4P_1$**.**
Let G be a $4P_1$-free 3-colourable graph and let X_1, X_2, X_3 be the colour classes of some 3-colouring of G. Note that $|X_1|, |X_2|, |X_3| \leq 3$ since G is $4P_1$-free and X_1, X_2, X_3 are independent sets. If $\text{oct}(G) \leq 1$ then $\text{ioct}(G) = \text{oct}(G)$, so we need only consider the case when $\text{oct}(G) \geq 2$. Since G is $4P_1$-free, every independent odd cycle transversal has at most three vertices, so $\text{ioct}(G) \leq 3$.

Suppose, for contradiction, that $\text{oct}(G) \neq \text{ioct}(G)$. Since $\text{oct}(G) \leq \text{ioct}(G)$, it follows that $\text{oct}(G) = 2$ and $\text{ioct}(G) = 3$. If $|X_i| < 3$ for some $i \in \{1, 2, 3\}$ then X_i is an independent odd cycle transversal on fewer than three vertices, a contradiction. It follows that $|X_1| = |X_2| = |X_3| = 3$ and so G has exactly nine vertices. Let S be a minimum odd cycle transversal of G, in which case $|S| = 2$. Then $G - S$ is a bipartite graph on seven vertices. Therefore one of the parts of $G - S$ contains at least four vertices, and so $G - S$ (and therefore G) contains an induced $4P_1$. This contradiction implies that $\text{ioct}(G) = \text{oct}(G)$.

Case 2: $H = P_1 + P_3$**.**
Let G be a $(P_1 + P_3)$-free 3-colourable graph and let X_1, X_2, X_3 be the colour classes of some 3-colouring of G. By Lemma 41, every connected component of a $\overline{P_1 + P_3}$-free graph is either C_3-free or complete multi-partite. Let $D_1, \ldots, D_r$ be the connected components of $\overline{G}$. Then $V(G)$ can be partitioned into sets $A_1, \ldots, A_r$, with $A_i = V(D_i)$ for $i \in \{1, \ldots, r\}$, such that

(a) for all $i \in \{1, \ldots, r\}$, the graph $G[A_i]$ is either $3P_1$-free or a disjoint union of complete graphs, and
(b) for all $i, j \in \{1, \ldots, r\}$ with $i \neq j$, the set A_i is complete to the set A_j.

As G is 3-colourable and hence contains no K_4, Property (b) implies that $r \leq 3$. First suppose that $r = 3$. Then, as G is 3-colourable, each A_i must be an independent set. Hence, G is a complete 3-partite graph with partition classes A_1, A_2, A_3. It follows that $\text{ioct}(G) = \text{oct}(G) = \min\{|A_1|, |A_2|, |A_3|\}$.

Now suppose that $r = 2$. As G is 3-colourable and A_1 is complete to A_2, one of the sets A_1 or A_2, say A_1, must be an independent set, and the other set, A_2, must induce a bipartite graph. First assume that $G[A_2]$ is a disjoint union of complete graphs. As $G[A_2]$ is bipartite, this means that every connected component of $G[A_2]$ has at most two vertices (see Fig. 28 for an example). Pick a vertex of each edge in $G[A_2]$ and let A_2' be the set of

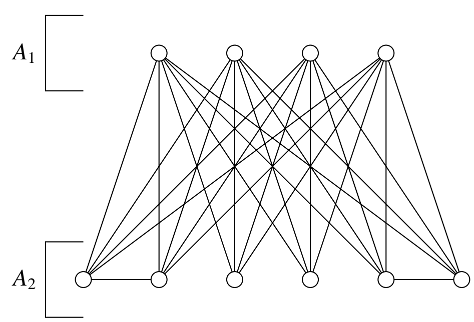


Fig. 28: An example of a $(P_1 + P_3)$-free 3-colourable graph G in the case when $r = 2$ and $G[A_2]$ is the disjoint union of one or more complete graphs on at most two vertices.

selected vertices. Then $\text{ioct}(G) = \text{oct}(G) = \min\{|A_1|, |A_2'|\}$. By Property (a), it remains to consider the case when $G[A_2]$ is bipartite and $3P_1$-free. Then $\text{ivc}(G[A_2]) \leq 2$ and so $\text{ioct}(G) \leq 2$ and therefore $\text{ioct}(G) = \text{oct}(G)$.

Finally, suppose that $r = 1$. If $G = G[A_1]$ is the disjoint union of complete graphs, then each complete graph must have at most three vertices (as G is 3-colourable). This implies that $\text{ioct}(G) = \text{oct}(G)$. If $G = G[A_1]$ is $3P_1$-free, then $\text{ioct}(G) \leq 2$ and therefore $\text{ioct}(G) = \text{oct}(G)$. We conclude that $\text{ioct}(G) = \text{oct}(G)$.

Case 3: $H = 2P_1 + P_2$**.**
Let G be a $(2P_1+P_2)$-free 3-colourable graph. As G is 3-colourable, we can partition $V(G)$ into three independent sets A, B, C. If $\text{oct}(G) \leq 1$, then $\text{ioct}(G) = \text{oct}(G)$. Hence, we may assume that $\text{oct}(G) \geq 2$. For contradiction, we assume that $\text{ioct}(G) \geq \text{oct}(G) + 1$. As $\text{oct}(G) \geq 2$, it follows that G is not bipartite. Hence, A, B, C are non-empty and moreover, there exists an edge between each pair of these sets. We claim that every subgraph of G induced by two vertices in one set in $\{A, B, C\}$ and two vertices in another set in $\{A, B, C\}$ has at least one edge. This can be seen as follows. For contradiction, suppose that there exist two vertices a_1, a_2 of A and two vertices b_1, b_2 of B, such that $\{a_1, a_2, b_1, b_2\}$ is an independent set. As $G[A \cup B]$ contains an edge, there exist adjacent vertices $x \in A$ and $y \in B$. As $\{a_1, a_2, b_1, b_2\}$ is an independent set, it follows that $x \notin \{a_1, a_2\}$ or $y \notin \{b_1, b_2\}$. Assume without loss of generality that $x \notin \{a_1, a_2\}$. Then y must be adjacent to least one of a_1, a_2, as otherwise $\{a_1, a_2, x, y\}$ would induce $2P_1 + P_2$. Assume without loss of generality that y is adjacent to a_1. Then $y \notin \{b_1, b_2\}$, as $\{a_1, a_2, b_1, b_2\}$ is an independent set. However, now $\{b_1, b_2, a_1, y\}$ induces $2P_1 + P_2$, a contradiction. Hence, the claim holds.

Now let S be a minimum odd cycle transversal of G. Let $A' = A \setminus S$, $B' = B \setminus S$ and $C' = C \setminus S$. First suppose that each of A', B', C' contains at least three vertices. As S is an odd cycle transversal, $G - S = G[A' \cup B' \cup C']$ is bipartite. Hence, $A' \cup B' \cup C'$ can be partitioned into two independent sets X and Y. As each of A', B', C' has at least three vertices, one of X, Y, say X, contains two vertices of at least two sets of A', B', C'. By the above claim, $G[X]$ contains an edge, a contradiction. Hence, we may assume without loss of generality that $|A'| \leq 2$, so $|S \cap A| \geq |A| - 2$. Since A is an independent odd cycle transversal, it follows that $|A| \geq \text{ioct}(G)$. Hence, we obtain

$$|S \cap A| \geq |A| - 2 \geq \text{ioct}(G) - 2 \geq \text{oct}(G) - 1 = |S| - 1.$$

As S is not an independent set, this implies that $|S \cap A| = |A| - 2 = |S| - 1$, and thus S contains exactly one vertex from $B \cup C$, say, $S \cap B = \{b\}$ (and thus $S \cap C = \emptyset$). As $|S \cap A| = |A| - 2$, it follows that $|A'| = |A \setminus S| = 2$. Let $A' = \{a', a''\}$. Since $\text{ioct}(G) > \text{oct}(G) \geq 2$, and B and C are odd cycle transversals, it follows that $|B|, |C| \geq 3$.

Suppose that $|B| \geq 4$. As $\text{ioct}(G) > \text{oct}(G)$, the independent set $(A \cap S) \cup \{a''\}$ is not an odd cycle transversal. Consequently, $G - ((A \cap S) \cup \{a''\}) = G[\{a'\} \cup B \cup C]$ is not bipartite. As $G[B \cup C]$ is bipartite, this means that $G - ((A \cap S) \cup \{a''\})$ has an odd cycle containing a'. This implies that a' has a neighbour in both B and C. As G is $(2P_1 + P_2)$-free and $|B| \geq 4$, this means that a' has at least three neighbours in B, and thus at at least two neighbours b_1, b_2 in $B \setminus \{b\}$. As $|C| \geq 3$, we find for the same reason that a' has at least two neighbours c_1, c_2 in C. By our previous claim, there is at least one edge with one end-vertex in $\{b_1, b_2\}$, say b_1, and the other one in $\{c_1, c_2\}$, say c_1. However, now $\{a', b_1, c_1\}$ induces a C_3 in $G - ((A \cap S) \cup \{b\})$, contradicting the fact that $S = (A \cap S) \cup \{b\}$ is an odd cycle transversal. We conclude that $|B| = 3$, say $B = \{b, b', b''\}$.

As $3 = |B| \geq \text{ioct}(G) > \text{oct}(G) = |S| \geq 2$, we find that $|S| = 2$. Hence $|S \cap A| = 1$ and $|A| = |S| + 2 = 3$, say $S = \{a, b\}$ and $A = \{a, a', a''\}$. In particular, both a' and a'' are adjacent to at least one vertex of B and to at least one vertex of C, as otherwise $\{a, a''\}$ or $\{a, a'\}$, respectively, is an independent odd cycle transversal of G of size 2.

By our claim, there exists at least one edge between a vertex of $\{a', a''\}$, say a', and a vertex of $\{b', b''\}$, say b'. Since $\{b, b''\}$ is not an odd cycle transversal and $G[A \cup C]$ is bipartite, b' belongs to an odd cycle in $G - \{b, b''\} = G[A \cup C \cup \{b'\}]$. This implies that b' has a neighbour in C. This, together with the fact that G is $(2P_1 + P_2)$-free, implies that b' is adjacent to all but at most one vertex in C. Recall that a' also has a neighbour in C. By the same argument, this means that a' is adjacent to all but at most one vertex in C. Since $|C| \geq 3$, we find that a' and b' have a common neighbour $c \in C$. Then, as a' and b' are adjacent, $\{a', b', c\}$ induces a C_3 in $G - \{a, b\}$, contradicting the fact that $S = \{a, b\}$ is an odd cycle transversal of G. We conclude that $\text{ioct}(G) = \text{oct}(G)$.

(ii): "⇒". Now suppose that $H = 2P_2$ or H is not a subgraph of P_4. By (i) we may assume that H is an induced subgraph of $K_{1,4}^+$ or $K_{1,4} + sP_1$ for some $s \geq 0$, which in particular implies that $H \neq 2P_2$. Recall that $H \notin \{K_{1,3}, K_{1,3}^+, 2P_1 + P_3\}$. This means that H contains an induced subgraph from the set $\{K_{1,4}, K_{1,3} + P_1, 5P_1, 3P_1 + P_2\}$.

First consider the graph G from Fig. 29 and note G is $(K_{1,4}, K_{1,3} + P_1, 5P_1)$-free and 3-colourable. Moreover, $\{u, v\}$ is a minimum odd cycle transversal, so $\text{oct}(G) = 2$, while $\text{ioct}(G) = 3$ (for instance, $\{u, u_1, u_2\}$ is a minimum independent odd cycle transversal of G). Now consider the graph G from Fig. 30. It is readily seen that G is $(3P_1 + P_2)$-free and 3-colourable. Moreover, $\text{oct}(G) = 2$, as $\{u, v\}$ is a minimum odd cycle transversal, while $\text{ioct}(G) = 3$ (for instance, $\{u, u_1, u_2\}$ is a minimum independent odd cycle transversal of G). This completes the proof of (ii).

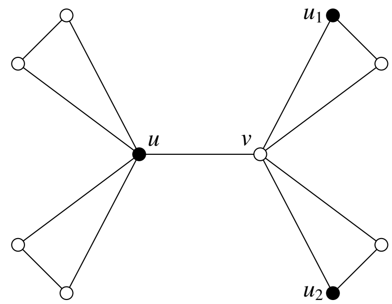


Fig. 29: A $(K_{1,4}, K_{1,} + P_1, 5P_1)$-free 3-colourable graph G with $\text{ioct}(G) = \text{oct}(G) + 1$.

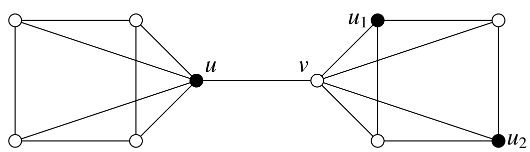


Fig. 30: A $(3P_1 + P_2)$-free 3-colourable graph G with $\text{ioct}(G) = \text{oct}(G) + 1$.

We now consider Statements (1)–(6). Statement (1) immediately follows from Statement (ii), whereas Lemma 36 and Corollary 2 imply Statements (5) and (6), respectively. It remains to prove Statements (2)–(4).

(2). Let $H = sP_1$ for some $s \geq 5$. Let G be an sP_1-free 3-colourable graph. If $\text{oct}(G) \leq 1$, then $\text{ioct}(G) = \text{oct}(G)$. Hence, we may assume that $\text{oct}(G) \geq 2$. As G is 3-colourable, $V(G)$ can be partitioned into three independent sets V_1, V_2, V_3. Hence, V_1 is an independent odd cycle transversal. As G is sP_1-free, V_1 has size at most $s - 1$. This means that $\text{ioct}(G) \leq s - 1 = 2 + s - 3 \leq \text{oct}(G) + s - 3$.

(3) and (4). Statements (3) and (4) follow from Lemma 37 after observing that $\text{ioct}(G) = \text{oct}(G)$ holds for every $K_{1,r}$-free 3-colourable graph G with $r \in \{1, 2\}$ (this also follows from (1)). This completes the proof. □

6.6 Conclusions

To develop an insight into the price of independence for classical concepts, we have investigated whether or not the size of a minimum independent vertex cover, feedback vertex set or odd cycle transversal is bounded in terms of the minimum size of the not-necessarily-independent variant of each of these transversals for H-free graphs (that have such independent transversals). While we note that the bounds we give in some of our results are tight, in this section we were mainly concerned with obtaining dichotomy results on whether there is a bound, rather than trying to find exact bounds. We will now discuss some open problems resulting from our work.

We fully classified for which graphs H the class of H-free bipartite graphs is ivc–bounded and for which graphs H the class of H-free near-bipartite graphs is ifvs-bounded. By Lemma 37, for $r, s \geq 1$ the class of $K_{1,r}$-free 3-colourable graphs is ioct-bounded if and only if the class of $(K_{1,r} + sP_1)$-free 3-colourable graphs is ioct-bounded. Therefore, Theorem 29 (and similarly, Theorem 33 (i)) leaves three open cases with respect to ioct-boundedness, as follows:

Open Problem 14 *Determine whether the class of H-free* 3*-colourable graphs is* ioct-*bounded when H is:*

1. $K_{1,4}$ *(or equivalently $K_{1,4} + sP_1$ for any $s \geq 1$),*
2. $K_{1,3}^{+}$ *or*
3. $K_{1,4}^{+}$.

We fully classified for which graphs H the class of H-free bipartite graphs is ivc–identical. However, we have a few remaining cases for the notions of being ifvs-identical (one open case) and being ioct-identical (three open cases):

Open Problem 15 *Does there exist a $K_{1,3}$-free near-bipartite graph G with* $\text{ifvs}(G) > \text{fvs}(G)$*?*

Open Problem 16 *For $H \in \{K_{1,3}, K_{1,3}^{+}, 2P_1+P_3\}$, does there exist an H-free* 3*-colourable graph G with* $\text{ioct}(G) > \text{oct}(G)$*?*

In particular, we note that the $H = K_{1,3}^{+}$ case is the only one open for both Open Problem 14 and Open Problem 16. We also note that, in contrast to the class of $(2P_1 + P_3)$-free

3-colourable graphs (see, for example, [19]), the classes of $K_{1,3}$-free near-bipartite graphs and $K_{1,3}$-free 3-colourable graphs are NP-complete to recognize. This follows from the results that the problems of deciding near-bipartiteness [12] and deciding 3-colourability [61] are NP-complete for line graphs, which form a subclass of $K_{1,3}$-free graphs.

As results for the price of connectivity implied algorithmic consequences for connected transversal problems [28,64], it is natural to ask whether our results for the price of independence have similar consequences. The problems Independent Vertex Cover, Independent Feedback Vertex Set and Independent Odd Cycle Transversal ask to determine the minimum size of the corresponding independent transversal. The first problem is readily seen to be polynomial-time solvable. The other two problems are NP-hard for H-free graphs whenever H is not a linear forest [12], just like their classical counterparts Feedback Vertex Set [79,83] and Odd Cycle Transversal [39] (see also [66,69]). The complexity of these four problems restricted to H-free graphs is still poorly understood when H is a linear forest. Our results suggest that it is unlikely that we can obtain polynomial algorithms for the independent variants based on results for the original variants, as the difference between ifvs(G) and fvs(G) and between ioct(G) and oct(G) can become unbounded quickly.

References

1. T. Abrishami, M. Chudnovsky, M. Pilipczuk, P. Rzążewski, and P. Seymour. Induced subgraphs of bounded treewidth and the container method. *Proc. SODA 2021*, pages 1948–1964, 2021.
2. V. E. Alekseev. Polynomial algorithm for finding the largest independent sets in graphs without forks. *Discrete Applied Mathematics*, 135(1-3):3–16, 2004.
3. G. Bacsó and Z. Tuza. Dominating cliques in P_5-free graphs. *Periodica Mathematica Hungarica*, 21(4):303–308, 1990.
4. V. Balachandran, P. Nagavamsi, and C. P. Rangan. Clique transversal and clique independence on comparability graphs. *Information Processing Letters*, 58(4):181–184, 1996.
5. E. Balas and C. S. Yu. On graphs with polynomially solvable maximum-weight clique problem. *Networks*, 19(2):247–253, 1989.
6. L. W. Beineke. Characterizations of derived graphs. *Journal of Combinatorial theory*, 9(2):129–135, 1970.
7. R. Belmonte, P. van 't Hof, M. Kamiński, and D. Paulusma. The price of connectivity for feedback vertex set. *Discrete Applied Mathematics*, 217(2):132–143, 2017.
8. B. Bergougnoux, C. Papadopoulos, and J. A. Telle. Node multiway cut and subset feedback vertex set on graphs of bounded mim-width. *Proc. WG 2020, LNCS*, 12301:388–400, 2020.
9. H. L. Bodlaender, N. Brettell, M. Johnson, G. Paesani, D. Paulusma, and E. J. van Leeuwen. Steiner trees for hereditary graph classes: a treewidth perspective. *Theoretical Computer Science*, 867:30–39, 2021.
10. H. L. Bodlaender and R. H. Möhring. The pathwidth and treewidth of cographs. *SIAM Journal on Discrete Mathematics*, 6(2):181–188, 1993.
11. M. Bonamy, K. K. Dabrowski, C. Feghali, M. Johnson, and D. Paulusma. Recognizing graphs close to bipartite graphs. *Proc. MFCS 2017, LIPIcs*, 83:70:1–14, 2017.
12. M. Bonamy, K. K. Dabrowski, C. Feghali, M. Johnson, and D. Paulusma. Independent feedback vertex set for P_5-free graphs. *Algorithmica*, 81(4):1342–1369, 2019.
13. A. Brandstädt and D. Kratsch. On the restriction of some NP-complete graph problems to permutation graphs. *Proc. FCT 1985, LNCS*, 199:53–62, 1985.
14. A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: A Survey*, volume 3 of *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, 1999.
15. A. Brandstädt and R. Mosca. Maximum weight independent set for $l$claw-free graphs in polynomial time. *Discrete Applied Mathematics*, 237:57–64, 2018.
16. A. Brandstädt and H. Müller. The NP-completeness of Steiner Tree and Dominating Set for chordal bipartite graphs. *Theoretical Computer Science*, 53:257–265, 1987.
17. N. Brettell, M. Johnson, G. Paesani, and D. Paulusma. Computing subset transversals in H-free graphs. *Theoretical Computer Science*, accepted for publication, 2021.
18. N. Brettell, M. Johnson, and D. Paulusma. Computing weighted subset transversals in Hs-free graphs. *CoRR*, abs/2007.14514, 2020.

19. H. Broersma, P. A. Golovach, D. Paulusma, and J. Song. Updating the complexity status of coloring graphs without a fixed induced linear forest. *Theoretical Computer Science*, 414(1):9–19, 2012.
20. E. Camby. Price of independence for the dominating set problem. Manuscript, 2017.
21. E. Camby. Price of connectivity for the vertex cover problem and the dominating set problem: Conjectures and investigation of critical graphs. *Graphs and Combinatorics*, 35(1):103–118, 2019.
22. E. Camby, J. Cardinal, S. Fiorini, and O. Schaudt. The price of connectivity for vertex cover. *Discrete Mathematics & Theoretical Computer Science*, 16(1):207–224, 2014.
23. E. Camby and F. Plein. A note on an induced subgraph characterization of domination perfect graphs. *Discrete Applied Mathematics*, 217(3):711–717, 2017.
24. E. Camby and O. Schaudt. The price of connectivity for dominating set: upper bounds and complexity. *Discrete Applied Mathematics*, 177:53–59, 2014.
25. E. Camby and O. Schaudt. A new characterization of P_k-free graphs. *Algorithmica*, 75(1):205–217, 2016.
26. J. Cardinal and E. Levy. Connected vertex covers in dense graphs. *Theoretical Computer Science*, 411(26-28):2581–2590, 2010.
27. B. V. Cherkassky, A. V. Goldberg, and T. Radzik. Shortest paths algorithms: theory and experimental evaluation. *Mathematical Programming*, 73:129–174, 1996.
28. N. Chiarelli, T. R. Hartinger, M. Johnson, M. Milanič, and D. Paulusma. Minimum connected transversals in graphs: new hardness results and tractable cases using the price of connectivity. *Theoretical Computer Science*, 705:75–83, 2018.
29. R. Chitnis, F. V. Fomin, D. Lokshtanov, P. Misra, M. S. Ramanujan, and S. Saurabh. Faster exact algorithms for some terminal set problems. *Journal of Computer and System Sciences*, 88:195–207, 2017.
30. M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. *Annals of mathematics*, 164(1):51–229, 2006.
31. D. G. Corneil, H. Lerchs, and L. S. Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3(3):163–174, 1981.
32. D. G. Corneil, Y. Perl, and L. K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14(4):926–934, 1985.
33. M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
34. M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. Subset feedback vertex set is fixed-parameter tractable. *SIAM Journal on Discrete Mathematics*, 27(1):290–309, 2013.
35. K. K. Dabrowski, C. Feghali, M. Johnson, G. Paesani, D. Paulusma, and P. Rzążewski. On cycle transversals and their connected variants in the absence of a small linear forest. *Algorithmica*, 82(10):2841–2866, 2020.
36. K. K. Dabrowski, M. Johnson, G. Paesani, D. Paulusma, and V. Zamaraev. On the price of independence for vertex cover, feedback vertex set and odd cycle transversal. *Proc. MFCS 2018, LIPIcs*, 117:63:1–63:15, 2018.

37. K. K. Dabrowski, M. Johnson, G. Paesani, D. Paulusma, and V. Zamaraev. Independent transversals versus transversals. *Proc. EuroComb 2019, Acta Mathematica Universitatis Comenianae*, 88:585–591, 2019.
38. D. Du and X. Hu. *Steiner Tree Problems in Computer Communication Networks*. World Scientific, 2008.
39. T. Emden-Weinert, S. Hougardy, and B. Kreuter. Uniquely colourable graphs and the hardness of colouring graphs of large girth. *Combinatorics, Probability and Computing*, 7(4):375–386, 1998.
40. P. Erdős and G. Szekeres. A combinatorial problem in peometry. *Compositio Mathematica*, 2:463–470, 1935.
41. B. Escoffier, L. Gourvès, and J. Monnot. Complexity and approximation results for the connected vertex cover problem in graphs and hypergraphs. *Journal of Discrete Algorithms*, 8(1):36–49, 2010.
42. A. Farrugia, P. Mihók, R. B. Richter, and G. Semanisin. Factorizations and characterizations of induced-hereditary and compositive properties. *Journal of Graph Theory*, 49(1):11–27, 2005.
43. H. Fernau and D. F. Manlove. Vertex and edge covers with clustering properties: complexity and algorithms. *Journal of Discrete Algorithms*, 7(2):149–167, 2009.
44. S. Földes and P. L. Hammer. Split graphs having dilworth number two. *Canadian Journal of Mathematics*, 29(3):666–672, 1977.
45. F. V. Fomin, P. Heggernes, D. Kratsch, C. Papadopoulos, and Y. Villanger. Enumerating minimal subset feedback vertex sets. *Algorithmica*, 69(1):216–231, 2014.
46. F. V. Fomin and Y. Villanger. Finding induced subgraphs via minimal triangulations. *Proc. STACS 2010, LIPIcs*, 5:383–394, 2010.
47. G. Fricke, S. T. Hedetniemi, and D. P. Jacobs. Independence and irredundance in k-regular graphs. *Ars Combinatoria*, 49, 1998.
48. M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.
49. M. R. Garey, D. S. Johnson, and L. J. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.
50. P. A. Golovach, P. Heggernes, D. Kratsch, and R. Saei. Subset feedback vertex sets in chordal graphs. *Journal of Discrete Algorithms*, 26:7–15, 2014.
51. P. A. Golovach, M. Johnson, D. Paulusma, and J. Song. A survey on the computational complexity of colouring graphs with forbidden subgraphs. *Journal of Graph Theory*, 84(4):331–363, 2017.
52. A. Grigoriev and R. Sitters. Connected feedback vertex set in planar graphs. *Proc. WG 2009, LNCS*, 5911:143–153, 2010.
53. C. Groenland, K. Okrasa, P. Rzążewski, A. Scott, P. Seymour, and S. Spirkl. H-colouring P_t-free graphs in subexponential time. *Discrete Applied Mathematics*, 267:184–189, 2019.
54. M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

55. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.
56. A. Grzesik, T. Klimošová, M. Pilipczuk, and M. Pilipczuk. Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. *Proc. SODA 2019*, pages 1257–1271, 2019.
57. V. Guruswami and C. P. Rangan. Algorithmic aspects of clique-transversal and clique-independent sets. *Discrete Applied Mathematics*, 100(3):183–202, 2000.
58. F. Harary and I. C. Ross. A procedure for clique detection using the group matrix. *Sociometry*, 20(3):205–215, 1957.
59. T. R. Hartinger, M. Johnson, M. Milanič, and D. Paulusma. The price of connectivity for cycle transversals. *European Journal of Combinatorics*, 58:203–224, 2016.
60. E. C. Hols and S. Kratsch. A randomized polynomial kernel for subset feedback vertex set. *Theory of Computing Systems*, 62(1):63–92, 2018.
61. I. Holyer. The NP-Completeness of edge-coloring. *SIAM Journal on Computing*, 10(4):718–720, 1981.
62. J. E. Hopcroft and R. E. Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Commununications of the ACM*, 16(6):372–378, 1973.
63. Y. Iwata, M. Wahlström, and Y. Yoshida. Half-integrality, LP-branching, and FPT algorithms. *SIAM Journal on Computing*, 45(4):1377–1411, 2016.
64. M. Johnson, G. Paesani, and D. Paulusma. Connected vertex cover for $(sP_1 + P_5)$-free graphs. *Algorithmica*, 82(1):20–40, 2020.
65. N. Kakimura, K. Kawarabayashi, and Y. Kobayashi. Erdős-Pósa property and its algorithmic applications: parity constraints, subset feedback set, and subset packing. *Proc. SODA 2012*, pages 1726–1736, 2012.
66. M. Kamiński and V. V. Lozin. Coloring edges and vertices of graphs without short or long cycles. *Contributions to Discrete Mathematics*, 2(1), 2007.
67. R. M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.
68. K. Kawarabayashi and Y. Kobayashi. Fixed-parameter tractability for the subset feedback set problem and the S-cycle packing problem. *Journal of Combinatorial Theory, Series B*, 102(4):1020–1034, 2012.
69. D. Král', J. Kratochvíl, Zs. Tuza, and G. J. Woeginger. Complexity of coloring graphs without forbidden induced subgraphs. *Proc. WG 2001, LNCS*, 2204:254–262, 2001.
70. S. Kratsch and M. Wahlström. Representative sets and irrelevant vertices: new tools for kernelization. *Journal of the ACM*, 67(3):16:1–16:50, 2020.
71. Y. Li, Z. Yang, and W. Wang. Complexity and algorithms for the connected vertex cover problem in 4-regular graphs. *Applied Mathematics and Computation*, 301:107–114, 2017.
72. D. Lokshantov, M. Vatshelle, and Y. Villanger. Independent set in P_5-free graphs in polynomial time. *Proc. SODA 2014*, pages 570–581, 2014.
73. D. Lokshtanov, P. Misra, M. S. Ramanujan, and S. Saurabh. Hitting selected (odd) cycles. *SIAM Journal on Discrete Mathematics*, 31(3):1581–1615, 2017.

74. V. V. Lozin and M. Milanič. A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. *Journal of Discrete Algorithms*, 6(4):595–604, 2008.
75. V. V. Lozin and R. Mosca. Independent sets in extensions of $2P_2$-free graphs. *Discrete Applied Mathematics*, 146(1):74–80, 2005.
76. V. V. Lozin and R. Mosca. Maximum regular induced subgraphs in $2P_3$-free graphs. *Theoretical Computer Science*, 460:26–33, 2012.
77. R. D. Luce and A. D. Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(1):95–116, 1949.
78. G. J. Minty. On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory, Series B*, 28(3):284–304, 1980.
79. N. Misra, G. Philip, V. Raman, and S. Saurabh. On parameterized independent feedback vertex set. *Theoretical Computer Science*, 461:65–75, 2012.
80. P. Misra, V. Raman, M. S. Ramanujan, and S. Saurabh. Parameterized algorithms for even cycle transversal. *Proc. WG 2012*, 7551:172–183, 2012.
81. E. F. Moore. The shortest path through a maze. *Proc. ISTS 1959*, pages 285—-292, 1959.
82. R. Mosca. Stable sets for $(P_6, K_{2,3})$-free graphs. *Discussiones Mathematicae Graph Theory*, 32:387–401, 2012.
83. A. Munaro. Boundary classes for graph problems involving non-local properties. *Theoretical Computer Science*, 692:46–71, 2017.
84. A. Munaro. On line graphs of subcubic triangle-free graphs. *Discrete Mathematics*, 340(6):1210–1226, 2017.
85. K. Okrasa and P. Rzążewski. Subexponential algorithms for variants of the homomorphism problem in string graphs. *Journal of Computer and System Sciences*, 109:126–144, 2020.
86. S. Olariu. Paw-free graphs. *Information Processing Letters*, 28(1):53–54, 1988.
87. C. Papadopoulos and S. Tzimas. Polynomial-time algorithms for the subset feedback vertex set problem on interval graphs and permutation graphs. *Discrete Applied Mathematics*, 258:204–221, 2019.
88. C. Papadopoulos and S. Tzimas. Subset feedback vertex set on graphs of bounded independent set size. *Theoretical Computer Science*, 814:177–188, 2020.
89. S. Poljak. A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 15:307–309, 1974.
90. P. L. K. Priyadarsini and T. Hemalatha. Connected vertex cover in 2-connected planar graph with maximum degree 4 is NP-complete. *International Journal of Mathematical, Physical and Engineering Sciences*, 2(1):51–54, 2008.
91. H. J. Prömel and A. Steger. *The Steiner Tree Problem: A Tour through Graphs, Algorithms, and Complexity*. Springer Science & Business Media, 2012.
92. V. Raman, S. Saurabh, and S. Sikdar. Improved exact exponential algorithms for vertex bipartization and other problems. *Proc. ICTCS 2005, LNCS*, 3701:375–389, 2005.
93. N. Robertson and P. D. Seymour. Graph minors. I. Excluding a forest. *Journal of Combinatorial Theory. Series B.*, 35(1):39–61, 1983.

94. N. Robertson and P. D. Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory. Series B.*, 92(2):325–357, 2004.
95. N. Sbihi. Algorithme de recherche d'un stable de cardinalité maximum dans un graphe sans étoile. *Discrete Mathematics*, 29(1):53–76, 1980.
96. S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505–517, 1977.
97. S. Ueno, Y. Kajitani, and S. Gotoh. On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. *Discrete Mathematics*, 72(1–3):355–360, 1988.
98. K. Wagner. Über eine eigenschaft der ebenen komplexe. *Mathematische Annalen*, 114(1):570–590, 1937.
99. T. Watanabe, S. Kajita, and K. Onaga. Vertex covers and connected vertex covers in 3-connected graphs. *Proc. IEEE 1991*, 2:1017–1020, 1991.
100. K. White, M. Farber, and W. R. Pulleyblank. Steiner trees, connected domination and strongly chordal graphs. *Networks*, 15(1):109–124, 1985.
101. M. Xiao and H. Nagamochi. Confining sets and avoiding bottleneck cases: a simple maximum independent set algorithm in degree-3 graphs. *Theoretical Computer Science*, 469:92–104, 2013.
102. M. Xiao and H. Nagamochi. Exact algorithms for maximum independent set. *Information and Computation*, 255(1):126–146, 2017.